



# Slot Accounting System Protocol

## Version 6.03

August 28, 2017

**PURPOSE:** The purpose of the SAS protocol version 6.03 is to facilitate communications between gaming machines and gaming systems.

**BENEFITS:** SAS 6.03 is intended to benefit electronic gaming device manufacturers, system manufacturers, operators, and regulators by defining the system/game communication protocol. The goal of this specification is to improve interoperability between equipment provided by various gaming equipment manufacturers.

## NOTICE OF COPYRIGHT

The information contained within this document is the confidential and proprietary property of International Game Technology. Unauthorized reproductions or distributions, complete or partial, are strictly prohibited. Any person who desires a copy of this document should contact IGT.

## Disclaimer of Warranty

THIS SPECIFICATION IS PROVIDED "AS IS," AND IGT MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE WITH RESPECT TO ANY SUCH MATERIALS, AND IGT SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IGT DOES NOT WARRANT THAT THE SPECIFICATION WILL MEET USER'S REQUIREMENTS, THAT THE OPERATION OF THE SPECIFICATION WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT DEFECTS IN THE SPECIFICATION WILL BE CORRECTED. FURTHERMORE, IGT DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING USE OR THE RESULTS OF THE USE OF THE SPECIFICATION IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY OR OTHERWISE. LICENSOR DISCLAIMS ANY WARRANTY RELATING TO INFRINGEMENT OF PATENTS OWNED BY OTHERS. USER ASSUMES RESPONSIBILITY FOR AVOIDING INFRINGEMENT OF PATENTS OWNED BY OTHERS APPLICABLE TO THE SPECIFICATION, DERIVATIVE SOFTWARE AND/OR LICENSED PRODUCTS.

## Limited Liability

IN NO EVENT WILL IGT BE LIABLE TO ANY PERSON OR ENTITY FOR LOSS OF DATA, COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR ANY SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES, UNDER ANY CAUSE OF ACTION AND WHETHER OR NOT SUCH PARTY OR ITS AGENTS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THIS LIMITATION WILL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

# Slot Accounting System Protocol Version 6.03

## Table of Contents

Table of Contents .....	i
Revision History .....	vii
Overview.....	xiii
<b>Section 1 GAMING MACHINE INTERFACE .....</b>	<b>1-1</b>
<b>1.1 Physical Interface .....</b>	<b>1-1</b>
1.1.1 Daisy Chain .....	1-1
1.1.2 Smart Interface Boards.....	1-2
<b>1.2 Logical Interface .....</b>	<b>1-2</b>
1.2.1 Wakeup Mode .....	1-2
<b>Section 2 COMMUNICATIONS.....</b>	<b>2-1</b>
<b>2.1 Gaming Machine Addressing.....</b>	<b>2-1</b>
<b>2.2 Host Polling .....</b>	<b>2-1</b>
2.2.1 General Polls .....	2-1
2.2.2 Long Polls .....	2-2
2.2.3 Transmitted Data Formats.....	2-4
<b>2.3 Timing Requirements.....</b>	<b>2-4</b>
2.3.1 Gaming Machine Response Time .....	2-4
2.3.2 Inter-Byte Delay Time .....	2-4
2.3.3 Polling Rate .....	2-5
<b>Section 3 HOST ACKNOWLEDGMENT .....</b>	<b>3-1</b>
<b>3.1 Implied Acknowledgment .....</b>	<b>3-1</b>
<b>3.2 Implied Negative Acknowledgment.....</b>	<b>3-1</b>
<b>3.3 Synchronization.....</b>	<b>3-1</b>
<b>Section 4 ERROR CONDITIONS .....</b>	<b>4-1</b>
<b>4.1 Gaming Machine Busy Response .....</b>	<b>4-1</b>
<b>4.2 Loop Break Indication .....</b>	<b>4-1</b>
<b>4.3 Link Down Detection .....</b>	<b>4-1</b>
<b>4.4 Unsupported Long Polls .....</b>	<b>4-1</b>
<b>4.5 Collisions .....</b>	<b>4-1</b>
<b>Section 5 CYCLICAL REDUNDANCY CHECK.....</b>	<b>5-1</b>
<b>5.1 Convention.....</b>	<b>5-1</b>
<b>5.2 Host and Gaming Machine CRC Generation .....</b>	<b>5-1</b>
<b>Section 6 ROM SIGNATURE.....</b>	<b>6-1</b>
<b>6.1 Verification .....</b>	<b>6-1</b>

<b>6.2</b>	<b>LP 21: ROM Signature Verification .....</b>	<b>6-1</b>
<b>Section 7</b>	<b>LONG POLL RESPONSE SPECIFICATIONS.....</b>	<b>7-1</b>
<b>7.1</b>	<b>Single Meter Accounting Long Polls .....</b>	<b>7-1</b>
<b>7.2</b>	<b>Multiple Meter Accounting Long Polls.....</b>	<b>7-1</b>
<b>7.3</b>	<b>LP 2F: Send Selected Meters for Game N.....</b>	<b>7-3</b>
<b>7.4</b>	<b>Enable/Disable Long Polls.....</b>	<b>7-4</b>
7.4.1	LP 01, 02: Shutdown (Lock Out Play) / Startup (Enable Play) .....	7-5
7.4.2	LP 0A, 0B: Enter/Exit Maintenance Mode.....	7-5
<b>7.5</b>	<b>LP 08: Configure Bill Denominations .....</b>	<b>7-5</b>
<b>7.6</b>	<b>Multi-Game Long Polls .....</b>	<b>7-7</b>
7.6.1	LP 09: Enable/Disable Game N.....	7-7
7.6.2	LP 2D: Send Total Hand Paid Cancelled Credits .....	7-7
7.6.3	LP 51: Send Number of Games Implemented .....	7-8
7.6.4	LP 52: Send Game N Meters .....	7-9
7.6.5	LP 53: Send Game N Configuration .....	7-9
7.6.6	LP 55: Send Selected Game Number.....	7-11
7.6.7	LP 56: Send Enabled Game Numbers.....	7-11
<b>7.7</b>	<b>LP 18: Send Games Played Since Last Power Up and Slot Door Closure Long Poll .....</b>	<b>7-11</b>
<b>7.8</b>	<b>LP 1B: Send Handpay Information.....</b>	<b>7-13</b>
7.8.1	Handpay Queue .....	7-14
7.8.2	Legacy Handpay Reporting.....	7-15
<b>7.9</b>	<b>LP 94: Remote Handpay Reset .....</b>	<b>7-15</b>
<b>7.10</b>	<b>LP 1F: Send Gaming Machine ID and Information.....</b>	<b>7-16</b>
<b>7.11</b>	<b>LP 48: Send Last Accepted Bill Information .....</b>	<b>7-17</b>
<b>7.12</b>	<b>LP 8E: Send Card Information .....</b>	<b>7-18</b>
<b>7.13</b>	<b>LP 8F: Send Physical Reel Stop Information.....</b>	<b>7-19</b>
<b>7.14</b>	<b>LP A0: Send Enabled Features.....</b>	<b>7-20</b>
<b>7.15</b>	<b>LP 54: Send SAS Version ID and Gaming Machine Serial Number .....</b>	<b>7-21</b>
<b>7.16</b>	<b>LP A4: Send Cash Out Limit.....</b>	<b>7-22</b>
<b>7.17</b>	<b>LP 7F: Receive Date and Time .....</b>	<b>7-22</b>
<b>7.18</b>	<b>LP 7E: Send Current Date and Time.....</b>	<b>7-23</b>
<b>7.19</b>	<b>LP 4F: Send Current Hopper Status .....</b>	<b>7-23</b>
<b>7.20</b>	<b>LP AA: Enable/Disable Game Auto Rebet.....</b>	<b>7-25</b>
<b>7.21</b>	<b>LP 6F, AF: Send Extended Meters for Game N .....</b>	<b>7-25</b>
<b>7.22</b>	<b>LP B3: Send Token Denomination .....</b>	<b>7-27</b>
<b>7.23</b>	<b>LP B5: Send Extended Game N Information .....</b>	<b>7-27</b>
<b>7.24</b>	<b>Weighted Average Theoretical Payback Percentage.....</b>	<b>7-29</b>
7.24.1	Calculated By Gaming Machine .....	7-29
7.24.2	LP B4: Send Wager Category Information.....	7-30
<b>Section 8</b>	<b>ADVANCED FUNDS TRANSFER.....</b>	<b>8-1</b>
<b>8.1</b>	<b>LP 73: AFT Register Gaming Machine .....</b>	<b>8-2</b>

<b>8.2</b>	<b>LP 74: AFT Game Lock and Status Request .....</b>	<b>8-5</b>
<b>8.3</b>	<b>LP 72: AFT Transfer Funds .....</b>	<b>8-8</b>
<b>8.4</b>	<b>Accepting Transfers .....</b>	<b>8-18</b>
<b>8.5</b>	<b>Bonus Awards .....</b>	<b>8-19</b>
<b>8.6</b>	<b>Transaction History.....</b>	<b>8-19</b>
<b>8.7</b>	<b>Host Cashout Enable .....</b>	<b>8-20</b>
<b>8.8</b>	<b>Cash Out Button Pressed .....</b>	<b>8-21</b>
<b>8.9</b>	<b>Lock After Transfer .....</b>	<b>8-22</b>
<b>8.10</b>	<b>AFT Meters .....</b>	<b>8-22</b>
<b>8.11</b>	<b>Transaction Receipts .....</b>	<b>8-22</b>
8.11.1	LP 75: Set AFT Receipt Data.....	8-23
8.11.2	Transaction Receipt Layout .....	8-24
8.11.3	Sample Transaction Receipts.....	8-26
<b>8.12</b>	<b>LP 76: Set Custom AFT Ticket Data .....</b>	<b>8-28</b>
<b>Section 9</b>	<b>RESERVED .....</b>	<b>9-1</b>
<b>Section 10</b>	<b>PROGRESSIVES .....</b>	<b>10-1</b>
<b>10.1</b>	<b>LP 80, 86: Progressive Broadcasts .....</b>	<b>10-1</b>
10.1.1	Group.....	10-2
10.1.2	Level.....	10-2
10.1.3	Amount.....	10-2
<b>10.2</b>	<b>Timing.....</b>	<b>10-2</b>
<b>10.3</b>	<b>Contributions .....</b>	<b>10-2</b>
<b>10.4</b>	<b>LP 84: Send Progressive Win Amount.....</b>	<b>10-2</b>
<b>10.5</b>	<b>SAS-Controlled Progressives .....</b>	<b>10-3</b>
10.5.1	LP 85: Send SAS Progressive Win Amount.....	10-3
10.5.2	LP 87: Send Multiple SAS Progressive Win Amounts.....	10-4
<b>10.6</b>	<b>Resetting Progressive Levels .....</b>	<b>10-5</b>
<b>10.7</b>	<b>LP 83: Send Cumulative Progressive Wins .....</b>	<b>10-5</b>
<b>Section 11</b>	<b>TOURNAMENT .....</b>	<b>11-1</b>
<b>11.1</b>	<b>LP 8C: Enter/Exit Tournament Mode .....</b>	<b>11-1</b>
<b>11.2</b>	<b>Entering Tournament Mode.....</b>	<b>11-1</b>
<b>11.3</b>	<b>Tournament Accounting .....</b>	<b>11-1</b>
<b>Section 12</b>	<b>REAL TIME EVENT REPORTING .....</b>	<b>12-1</b>
<b>12.1</b>	<b>LP 0E: Enable/Disable Real Time Event Reporting.....</b>	<b>12-1</b>
<b>12.2</b>	<b>Polling Method .....</b>	<b>12-1</b>
<b>12.3</b>	<b>Priority.....</b>	<b>12-1</b>
<b>12.4</b>	<b>Host/Gaming Machine Acknowledgment .....</b>	<b>12-1</b>
<b>12.5</b>	<b>Event Response Format .....</b>	<b>12-1</b>
12.5.1	Exc 4F: Bill Accepted .....	12-2

12.5.2	Exc 7C: Legacy Bonus Pay Was Awarded .....	12-2
12.5.3	Exc 7E: Game Start .....	12-3
12.5.4	Exc 7F: Game End .....	12-4
12.5.5	Exc 88: Reel N Has Stopped.....	12-5
12.5.6	Exc 8A: Game Recall Entry Displayed .....	12-5
12.5.7	Exc 8B: Card Held/Not Held .....	12-5
12.5.8	Exc 8C: Game Selected .....	12-6
12.5.9	Exc 9E: Session Start.....	12-6
12.5.10	Exc 9F: Session End .....	12-7
<b>12.6</b>	<b>No Activity Exceptions .....</b>	<b>12-7</b>
<b>12.7</b>	<b>ROM Signature Response.....</b>	<b>12-8</b>
<b>Section 13</b>	<b>BONUSING .....</b>	<b>13-1</b>
<b>13.1</b>	<b>Enabling/Disabling Bonusing .....</b>	<b>13-1</b>
<b>13.2</b>	<b>Reporting Active Players .....</b>	<b>13-1</b>
<b>13.3</b>	<b>LP 8A: Initiate Legacy Bonus Pay.....</b>	<b>13-2</b>
13.3.1	During Game Play.....	13-2
13.3.2	During Idle.....	13-2
13.3.3	During a Handpay.....	13-2
13.3.4	During Player Screens .....	13-2
13.3.5	During a Malfunction, Door Open, or Maintenance .....	13-3
<b>13.4</b>	<b>LP 8B: Multiplied Jackpot Mode (obsolete) .....</b>	<b>13-3</b>
13.4.1	Multiplied Jackpots and Multi-Line Gaming Machines .....	13-3
13.4.2	Multiplied Jackpots and Bonus Awards .....	13-4
13.4.3	Multiplied Jackpots and Progressive Wins.....	13-4
<b>13.5</b>	<b>LP 90: Send Legacy Bonus Win Amount .....</b>	<b>13-4</b>
<b>13.6</b>	<b>LP 9A: Send Legacy Bonus Meters.....</b>	<b>13-5</b>
<b>13.7</b>	<b>LP 2E: Game Delay .....</b>	<b>13-5</b>
<b>Section 14</b>	<b>JACKPOT HANDPAY RESET METHODS.....</b>	<b>14-1</b>
<b>14.1</b>	<b>Attendant Authorization .....</b>	<b>14-1</b>
<b>14.2</b>	<b>LP 8A: Enable Jackpot Handpay Reset Method.....</b>	<b>14-1</b>
<b>Section 15</b>	<b>VALIDATION AND TICKET REDEMPTION .....</b>	<b>15-1</b>
<b>15.1</b>	<b>Improved Ticket Expiration Support .....</b>	<b>15-2</b>
<b>15.2</b>	<b>LP 7B: Extended Validation Status .....</b>	<b>15-3</b>
<b>15.3</b>	<b>LP 7C: Set Extended Ticket Data .....</b>	<b>15-6</b>
<b>15.4</b>	<b>LP 7D: Set Ticket Data (legacy).....</b>	<b>15-8</b>
<b>15.5</b>	<b>LP 3D: Send Cash Out Ticket Information .....</b>	<b>15-10</b>
<b>15.6</b>	<b>LP 4C: Set Secure Enhanced Validation ID .....</b>	<b>15-10</b>
<b>15.7</b>	<b>LP 57: Send Pending Cashout Information .....</b>	<b>15-11</b>
<b>15.8</b>	<b>LP 58: Receive Validation Number .....</b>	<b>15-12</b>
<b>15.9</b>	<b>System Validation Examples.....</b>	<b>15-13</b>
15.9.1	Example 1, Host validates cashout ticket.....	15-13
15.9.2	Example 2, Host refuses to validate cashout ticket .....	15-14

<b>15.10</b>	<b>LP 4D: Send Enhanced Validation Information.....</b>	<b>15-14</b>
<b>15.11</b>	<b>LP 70: Send Ticket Validation Data .....</b>	<b>15-17</b>
<b>15.12</b>	<b>LP 71: Redeem Ticket .....</b>	<b>15-18</b>
<b>15.13</b>	<b>LP 50: Send Validation Meters .....</b>	<b>15-23</b>
<b>15.14</b>	<b>Secure Enhanced Validation Algorithm .....</b>	<b>15-24</b>
<b>Section 16</b>	<b><i>MULTI-DENOM EXTENSIONS.....</i></b>	<b>16-1</b>
<b>16.1</b>	<b>LP B0: Multi-Denom Preamble.....</b>	<b>16-1</b>
<b>16.2</b>	<b>Multi-Denom Preamble Examples.....</b>	<b>16-4</b>
16.2.1	Example 1 .....	16-4
16.2.2	Example 2 .....	16-5
<b>16.3</b>	<b>LP B1: Send Current Player Denomination.....</b>	<b>16-6</b>
<b>16.4</b>	<b>LP B2: Send Enabled Player Denominations.....</b>	<b>16-6</b>
<b>Section 17</b>	<b><i>COMPONENT AUTHENTICATION PROTOCOL .....</i></b>	<b>17-1</b>
<b>17.1</b>	<b>LP 6E: Send Authentication Info .....</b>	<b>17-1</b>
17.1.1	Interrogate Number of Installed Components .....	17-6
17.1.2	Interrogate Status.....	17-6
17.1.3	Authenticate Component .....	17-6
<b>Section 18</b>	<b><i>NEW IN SAS 6.03 .....</i></b>	<b>18-1</b>
<b>18.1</b>	<b>Meter Change Notification Support .....</b>	<b>18-1</b>
18.1.1	LP B6: Meter Collect Status .....	18-1
18.1.2	LP B7: Set Machine Numbers .....	18-3
<b>18.2</b>	<b>Session Play Support .....</b>	<b>18-5</b>
18.2.1	Session Play Jackpot Reporting.....	18-5
18.2.2	Exc 9E, 9F: Session Play Start/End .....	18-5
18.2.3	Targeted Restricted Promotional Tickets .....	18-5
<b>18.3</b>	<b>Tip Money Meter Support .....</b>	<b>18-6</b>
<b>18.4</b>	<b>Keyed Credits Support.....</b>	<b>18-6</b>
	Regular Gaming Machine Accounting Meters affected by Keyed Credit Transfers.....	18-7
<b>18.5</b>	<b>Foreign Currency Redemption Support .....</b>	<b>18-7</b>
18.5.1	Ticket Redemption Process Enhancements.....	18-7
18.5.2	LP 59: Send Enabled Currency Codes .....	18-9
18.5.3	LP 5A: Send Supported Bills .....	18-10
18.5.4	LP 5B: Send Bill Meters .....	18-11
18.5.5	LP 5C: Foreign Bill Reporting Mode .....	18-12
<b>18.6</b>	<b>Enhanced Progressive Data Reporting.....</b>	<b>18-13</b>
18.6.1	LP 5E: Send configured progressive controllers .....	18-13
18.6.2	LP 5F: Send progressive broadcast values .....	18-14
18.6.3	LP 77: Send progressive accounting data .....	18-16
<b>18.7</b>	<b>Non-SAS Progressive Hit Reporting .....</b>	<b>18-17</b>
18.7.1	Reporting Non-SAS Progressive Hits.....	18-18
18.7.2	LP 5D: Send Non-SAS Progressive Win Data .....	18-18
<b>18.8</b>	<b>Gaming Machine Enforced Max Progressive Payback .....</b>	<b>18-19</b>
18.8.1	LP 7A: Extended Progressive Broadcast.....	18-19
<b>18.9</b>	<b>Exc 5A: Jackpot Handpay Keyed Off to Machine Pay.....</b>	<b>18-20</b>

<b>Appendix A</b>	<b>GENERAL POLL EXCEPTION CODES.....</b>	<b>A-1</b>
<b>Appendix B</b>	<b>LONG POLL COMMANDS.....</b>	<b>B-1</b>
<b>Appendix C</b>	<b>DATA TABLES.....</b>	<b>C-1</b>
<b>Appendix D</b>	<b>FIGURES .....</b>	<b>D-1</b>
<b>Appendix E</b>	<b>LONG POLL 09 GUIDELINES.....</b>	<b>E-1</b>
	<b>Available Long Poll 09 Responses .....</b>	<b>E-1</b>
	<b>Enable/Disable Game 0000 .....</b>	<b>E-1</b>
	<b>Multi-Denom Games .....</b>	<b>E-1</b>
	<b>Interaction Between Host and Player/Operator.....</b>	<b>E-1</b>
	<b>Enable/Disable Tournament Games.....</b>	<b>E-1</b>
	<b>Example Responses.....</b>	<b>E-2</b>
<b>Glossary .....</b>		<b>1</b>



**Revision History**

Version	Revision
6.03	<p>SAS 6.03 consolidates all previously published errata and addenda, as well as adding new clarifications and functionality.</p> <p>Clarified inter-byte delay enforcement in Section 2.3.2.</p> <p>Clarified Shutdown/Startup commands in Section 7.4.1.</p> <p>Clarified long poll 08 behavior in Section 7.5 (Errata 12/6/2009).</p> <p>Clarified usage of Additional ID field in long polls 1F and 53.</p> <p>Clarified “center or first payline” in Section 7.13 (Errata 12/6/2009).</p> <p>Extended long poll B5 to report the number of number of player denoms a game can be configured for (Addenda 5/5/2008).</p> <p>Clarified Max Bet for disabled denominations and games in Section 7.23.</p> <p>Deprecated AFT Lock After Transfer feature (Errata 12/6/2009).</p> <p>Clarified multiple progressive wins in long poll 84, Send Progressive Win Amount (Addenda 5/5/2008).</p> <p>Clarified “SAS-controlled progressives” in Section 10.5.</p> <p>Clarified RTE NACK in Section 12.4.</p> <p>Clarified handling of reels with more than 255 stops in Section 12.5.5.</p> <p>Clarified Game Recall Entry Displayed in Section 12.5.6.</p> <p>Added authentication method SHA-256 to Table 17.1c (New in SAS 6.03)</p> <p>Updated memory-related exception names to better map to modern technology.</p> <p>Added exception 2F, Bill acceptor version changed (Addenda 5/5/2008).</p> <p>Clarified exception 3C, configuration option changed.</p> <p>Reserved long poll code EB.</p> <p>Added Appendix E, Long Poll 09 Implementation Guidelines.</p> <p>Added Section 18, New in SAS 6.03, with the following new functionality added since SAS 6.02 was published:</p> <ul style="list-style-type: none"> <li>• Meter Change Notification Support (Addenda 5/5/2008)</li> <li>• Session Play Support (Addenda 5/5/2008)</li> <li>• Tip Money Meter Support (Addenda 5/5/2008)</li> <li>• Keyed Credits Support (Addenda 5/5/2008)</li> <li>• Foreign Currency Redemption Support (Addenda 6/16/2011)</li> <li>• Enhanced Progressive Data Reporting (New in SAS 6.03)</li> <li>• Non-SAS Progressive Hit Reporting (New in SAS 6.03)</li> <li>• Gaming Machine Enforced Max Progressive Payback (New in SAS 6.03)</li> <li>• Jackpot Handpay Keyed Off to Machine Pay Exception (New in SAS 6.03)</li> </ul>

**Revision History (cont.)**

Version	Revision
6.02	<p>Merged errata approved by GSA on 3/16/2005.</p> <p>Merged Wager Category Support addendum approved by GSA on 2/25/2005.</p> <p>Merged bill hopper meters addendum approved by GSA on 10/19/2004.</p> <p>Corrected typographical errors.</p> <p>Clarified game numbering rules (Sections 2.2.2.3 and 7.6.3).</p> <p>Clarified link down rules (Section 4.3).</p> <p>Relabeled "credits" to "accounting denom" where appropriate.</p> <p>Clarified global broadcast poll addressing options.</p> <p>Clarified meter size for long poll 2F.</p> <p>Clarified usage of meter 007F to report weighted average theoretical payback percentage (Section 7.24.1).</p> <p>Clarified distinction between regular cashable (not including debit or promotional) and total cashable.</p> <p>Clarified intended use of AFT registration (Section 8.1).</p> <p>Eliminated 800 ms delay after cashout button pressed (Section 8.8).</p> <p>Added "Lock After Transfer" feature to AFT (Section 8).</p> <p>Clarified meaning of cumulative meters in AFT Transfer Complete (Section 8.3).</p> <p>Clarified SAS vs. non-SAS progressives (Section 10).</p> <p>Recommend not implementing exception 1F (Section 13.2).</p> <p>Recommend not implementing multiplied jackpots feature (long poll 8B).</p> <p>Clarified behavior when host redeems a restricted ticket with a past expiration date (Section 15.1).</p> <p>Clarified types of validated handpays (Table C-7).</p> <p>Corrected typo in Step 5 of Enhanced Validation algorithm (Section 15.15).</p> <p>Clarified operation of "RTE Only" exceptions.</p> <p>Added exception 2E, cashbox near full detected (Appendix A)</p> <p>Added Game ID codes to Table C-1.</p> <p>Added fractional denoms to Table C-4 and removed obscure data.</p> <p>Clarified foreign currency rules for Tables C-4 and C-6.</p> <p>Added "gaming machine" ticket count meters (meters 0035 through 0039).</p> <p>Added validation types and transfer types to Table C-7 for protocol-specific meters (from FAQ).</p> <p>Clarified content of AFT count meters (Table C-7).</p>

**Revision History (cont.)**

Version	Revision
6.01	<p>Corrected miscellaneous typographical errors.</p> <p>Clarified required vs. optional behavior.</p> <p>Changed Jurisdictional Cancelled Credits to Total Cancelled Credits for clarity.</p> <p>Clarified response options for invalid polls (Section 2.2.2).</p> <p>Clarified implied NACK due to loss of sync (Section 3.3).</p> <p>Clarified collision detection (Section 4.5).</p> <p>Explained Maintenance Mode (Section 7.4.2).</p> <p>Clarified handpay reporting (Section 7.8.1, 7.8.2).</p> <p>Clarified game lock process (Section 8.2).</p> <p>Clarified exception 56 behavior (Section 10.4.1).</p> <p>Consistently refer to “secure” enhanced validation, when separate from system validation, for clarity (Section 15). There is no non-secure enhanced validation.</p> <p>Clarified defaults for validation control/status bits (Table 15.2c).</p> <p>Clarified operation of long polls 3D and 4D to non-validating host (Section 15).</p> <p>Clarified operation of exceptions 3D and 3E (Section 15.10).</p> <p>Moved long poll B3 (Send Token Denom) and B5 (Send Extended Game N Info) from Sections 16.5 and 16.6 to Sections 7.22 and 7.23.</p> <p>Clarified requirements for authenticating components during game play.</p> <p>Indicate 40 ms polling rate support in long poll A0 response.</p> <p>Improved Link Down detection (Section 4.3).</p> <p>Added guidelines for unique AFT transaction IDs (Section 8).</p> <p>Changed “ATM” to “Debit” in several places.</p> <p>Corrected requirements for line 19 of AFT receipts.</p> <p>Added several bit definitions in AFT long polls 72 and 74.</p> <p>Added Transfer Code FE interrogation to long poll 72.</p> <p>Deleted Section 8.9, reporting player initiated cash outs (exception 26 and lp 66).</p> <p>Added authentication status codes to Table 17.1d.</p> <p>Added Section 8.12, set custom AFT ticket data (long poll 76).</p> <p>Added Send Multiple SAS Progressive Win Amounts poll (Section 10.4.3).</p> <p>Added Game Identification codes to Table C-1.</p> <p>Added \$0.40 denomination to Table C-4.</p> <p>Added meter 007F, weighted theoretical average, to Table C-7.</p> <p>Allow all type G polls to be sent to a specific address.</p>

**Revision History (cont.)**

Version	Revision
6.00	<p>Clarified exception queue (Section 2.2.1).</p> <p>Clarified “link down” condition (Section 4.3).</p> <p>Added long poll AF as alternate 6F meter poll, to allow consecutive meter polls.</p> <p>Clarified several issues and added features for Advanced Funds Transfers:</p> <ul style="list-style-type: none"> <li>Transaction ID must be printable ASCII.</li> <li>Eliminated requirement to reissue exceptions 6C and 6D (registration process).</li> <li>Specify only one lock condition with lock and status command.</li> <li>Clarified operation of funds transfer response transaction history buffer position.</li> <li>Clarified that ALL successful bonus transfers go in history, even if zero amount.</li> <li>Host may request cashout from gaming machine in funds transfer poll.</li> </ul> <p>Added transfer status types 93, 94, 95 and 9F to Table 8.3e.</p> <p>Clarified some issues in Real Time Event reporting:</p> <ul style="list-style-type: none"> <li>Maximum of nine reels in Reel N Has Stopped real time event.</li> <li>Only base hand reported in Card Held/Not Held real time event.</li> </ul> <p>Added ASCII game name to long poll B5.</p> <p>Added Section 17, Component Authentication Protocol.</p> <p>Removed legacy EFT poll 27, request current restricted promotional credits.</p> <p>Removed “TBD” polls 0C, 0D, 81, 82 and 93 since they were never defined.</p> <p>Added exceptions 98, 99, 9A and 9B (Table A-1).</p> <p>Added Game ID codes BI, CY, SD and SE to Table C-1.</p> <p>Added \$500,000 and \$1,000,000 bills to Table C-6.</p> <p>Added several meters, including bill meters, to Table C-7.</p>
5.10	<p>Added the following:</p> <ul style="list-style-type: none"> <li>Auto rebet command.</li> <li>Extended meter support.</li> <li>Advanced Funds Transfer Protocol.</li> <li>Extended validation support.</li> <li>Several more meters to Table C-7.</li> </ul> <p>Removed documentation of EFT.</p> <p>Reworked glossary.</p>
5.02	<p>Added EFT long poll 6B, Transfer Promotional Credits To Host.</p> <p>Added Section 16, multi-denom extensions.</p> <p>Corrected known typographical errors in 5.01.</p> <p>Extended EFT player cashout intercept time to 800ms.</p> <p>Renamed Hand Paid Credits meter to Hand Paid Cancelled Credits for clarity.</p> <p>Clarified multi-game support indication using long poll 51.</p> <p>Clarified max bet reporting.</p> <p>Changed the term “voucher” to “ticket” or “receipt” consistently.</p> <p>Clarified difference between cashout ticket and handpay receipt.</p> <p>Clarified proper metering of printed and redeemed tickets.</p> <p>Updated Game ID list (Table C-1).</p> <p>Added denominations (Table C-4).</p>
5.01	<p>Added the following:</p> <ul style="list-style-type: none"> <li>System Validation extensions to Enhanced Validation.</li> <li>Additional ticket meters for long poll 2F.</li> <li>New Hopper Status long poll 4F in Section 7.</li> <li>Updated Fig. 1 (4/26/2000).</li> </ul>

## Revision History (cont.)

Version	Revision
5.00	<p>Added the following:</p> <ul style="list-style-type: none"> <li>New selected meters command in Section 7.</li> <li>New date and time messages in Section 7.</li> <li>Remote handpay reset command in Section 7.</li> <li>Section 15 to describe the standard and enhanced validation support, and ticket redemption support.</li> <li>Clarified variable length messages, exception reporting.</li> </ul>
4.02	<p>Added the following:</p> <ul style="list-style-type: none"> <li>Section 14 to describe the jackpot handpay reset methods functionality.</li> <li>Additional progressive functionality in Section 10.</li> </ul> <p>Described the following:</p> <ul style="list-style-type: none"> <li>Game behavior upon accepting the game disable command.</li> <li>Bonus behavior upon recovering from a link down condition.</li> <li>ROM signature response during real time event reporting mode.</li> <li>Game behavior when a bonus is pending and the SAS link is lost.</li> <li>Game behavior when a bonus is received during maintenance, door open, handpays, and player screens.</li> <li>Use of the 'no activity' exceptions 00 and 1F.</li> </ul> <p>Clarified several glossary definitions.</p>
4.01	<p>Added long poll 55 (Send Selected Game Numbers) and 56 (Send Enabled Game Numbers).</p> <p>Added a game option configuration for Winner's Choice.</p> <p>Added game identification code CM for Coin Master UK.</p>
4.00	<p>Added additional long poll descriptions.</p> <p>Removed EFT long polls and renamed ECT to EFT.</p> <p>Added SAS progressives and SAS bonuses.</p> <p>Added functionality for real time event reporting.</p>
3.13	<p>Added the capability for the host to perform enhanced cashless transaction (ECT) to the gaming machine.</p>
3.06	<p>Added the capability for the host to perform electronic fund transfers (EFT) to the gaming machine. Game start and end exceptions have been added.</p>
2.83	<p>Added the generic bill accepted exception 4F and the corresponding long poll 48.</p> <p>Added long polls for entering and exiting maintenance mode, sending the cancelled credit meter, and sending 10 through 15. Added long polls to obtain the number of bills currently in the stacker and the total credit amount of all bills currently in the stacker.</p> <p>Added a country code and bill denomination table.</p> <p>The format for this document has been changed to better present the intended information.</p>
2.82	<p>Added schematics for fiber optic and PT95A-to-gamng machine electrical connections.</p> <p>Fixed various syntax and typographical errors.</p>
2.81	<p>Added the capability for host to enable/disable individual bill denominations.</p>
2.80	<p>Added extensions for ticket printer exceptions and ticket validation, multi-game long polls 51 and 52, and enhanced bill acceptor status reporting.</p>
2.40	<p>Added bill acceptor activity and reporting to version 2.00.</p>

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH

## Overview

This document specifies the logical and physical interface of a gaming machine to a slot accounting system host. The communication between the gaming machine and host occurs at 19.2Kbaud, using a wakeup format. Gaming machines can be interfaced to a host either by daisy chaining multiple gaming machines to a single data collection unit, or by connecting single machines to smart interface boards. To distinguish one gaming machine from another when using a daisy chained configuration, gaming machines must support an attendant configurable system address with a range of {0~127}. When a gaming machine is configured with an address of zero, it ignores all communications from the host.

The host requests data by sending general polls and long polls to the gaming machine. General polls are sent to the gaming machine to obtain event information. Gaming machines respond to general polls with a single byte exception code indicating that an event has occurred (e.g., door open, bill accepted, or handpay pending). When the host desires accounting information, such as the gaming machine's coin in meter, it issues a long poll requesting the specific data. When responding to a host long poll, the gaming machine message includes its address, host command, requested data, and a two-byte CRC.

To verify a gaming machine's Read Only Memory (ROM), the host issues a ROM signature request. Gaming machines are required to continue communications with the host while generating the signature. Once the gaming machine has completed generating the signature, it sends the signature to the host in response to the next general poll it receives. This response behavior is unique to the ROM signature request.

The host can provide progressive information to the gaming machines by performing a progressive broadcast. Coin in contributions for the progressives levels can be obtained by the host through the use of delta coin in amounts, coin inserted exceptions from the gaming machine, and/or the game start real time event.

For gaming machines that support tournament operation, the host can issue the enter/exit tournament mode command. This command includes the tournament time, number of starting credits, and an enable/disable tournament pulses field. For time only tournaments, this command is issued with a zero credit field. For credit only tournaments, this command is issued with a zero time field. To exit tournament mode, this command is issued with a zero for both time and credit fields.

In order to better obtain exceptions in a real time manner, the host can configure a gaming machine for real-time event reporting. Instead of responding with a single byte exception code, gaming machines respond with an exception message consisting of its address, event response identifier, exception code, any additional data, and a two-byte CRC. When in this mode, gaming machines can respond to long polls with event responses.

The host can act as a bonus controller for the gaming machines. Bonus awards and multiplied jackpots can be issued. Bonus awards instruct a gaming machine to award a single bonus amount. Multiplied jackpots configures a gaming machine to multiply certain wins before awarding them to the player.

To provide extra flexibility regarding resetting jackpot handpays, the host can instruct the gaming machine to reset a pending jackpot handpay amount to the gaming machine credit meter. This reduces the amount of jackpot attendant pays required for high denomination gaming machines while maintaining the handpay lock up functionality.

When a system requires a high level of cash out ticket and/or handpay security, support is available for an enhanced style of validation. Support is provided to allow tickets printed on one gaming machine to be redeemed on any gaming machine connected to the same slot monitoring system.

Support has been added for gaming machines that allow a player to select from more than one credit value for game play. Systems are able to track meters and play activity per denomination as well as per game.

Enhancements allow better support of extended meters. When the host requests meters in the extended format, the gaming machine can provide up to 18 digits per meter. Additional support has been added for restricted and nonrestricted promotional tickets.

Advanced Funds Transfer provides an improved, robust method for implementing electronic cashless systems.

The Component Authentication Protocol adds a mechanism to allow for remote verification that all executable programs and other fixed data stored in a gaming machine exactly matches the data that has been approved by the local jurisdiction.

To provide the greatest level of interoperability in the field between gaming machines and systems from different manufacturers, it is important to follow this specification as closely as possible, particularly when implementing features such as ticketing and funds transfer. However, it must be understood that it is not the intent of SAS to dictate gaming machine design, and some processes reported by SAS existed long before the protocol. Most importantly, whenever jurisdictional requirements are in direct conflict with the protocol, jurisdictional requirements always take precedence.



## SECTION 1 GAMING MACHINE INTERFACE

Section 1 details the physical and logical interface required for implementing SAS communications between a gaming machine and host.

### 1.1 Physical Interface

The gaming machine can be interfaced to the host by two methods. One method involves interfacing each gaming machine to a fiber tap board. The fiber tap boards can be daisy chained together to connect multiple gaming machines to a single host data collection unit.

The second interface method involves connecting each gaming machine to a smart interface board (SMIB). The SMIB polls the gaming machine to which it is connected and passes the information for that gaming machine to the host. Both of these interface methods are detailed below.

#### 1.1.1 Daisy Chain

Daisy chaining involves connecting multiple gaming machines to a single host via fiber tap boards. In an example configuration, the gaming machine provides a four-wire communication cable and a two-wire AC power cable to a fiber tap board (illustrated in Figure 1 of Appendix D). The communication cable is terminated with a Molex 70066 Series single-row connector (p/n 50-57-9404). Table 1.1.1a details the communication cable pin assignments.

Table 1.1.1a Pin Assignments for the 4-Wire Communication Cable		
Pin	Assignment	Description
1	Vdd	10 volts typical
2	Rxd	Serial data input to gaming machine
3	Txd	Serial data output from gaming machine
4	Gnd	Ground

A 3-wire power cable must provide UNSWITCHED 120V/220V AC power and may be terminated with an AMP connector (p/n 1-480701-0) or equivalent. Table 1.1.1b details the pin assignments for this connector.

Table 1.1.1b Pin Assignments for the 3-Wire Power Cable		
Pin	Assignment	Description
1	Hot	120V/220V AC
2	Gnd	Ground
3	Com	Common

### 1.1.2 Smart Interface Boards

The alternative to daisy-chaining multiple gaming machines to a single host is to install a SMIB in each gaming machine to continuously obtain and update information for a single gaming machine and to relay this information to the host as needed. Host manufacturers may develop their own SMIBs to communicate with gaming machines. For the IGT developed SMIB (i.e., PT95A player tracking device), a sample schematic showing the preferred and optional interface is illustrated in Figure 2 of Appendix D. When interfacing gaming machines to a non-IGT SMIB, contact the SMIB manufacturer for interface specifications.

## 1.2 Logical Interface

Communication between the host and gaming machines occurs through a serial data link operating at 19.2 KBaud in a "wakeup" mode. The 11-bit data packet consists of one start bit, eight data bits, a ninth 'wakeup' bit, and one stop bit.

### 1.2.1 Wakeup Mode

In wakeup mode, the host sets the 9th (wakeup) bit each time it sends the first byte of a message to the gaming machine. For all additional bytes in the message, this bit is cleared. Gaming machines use the wakeup bit to determine whether the received byte is the first byte of a new message or an additional byte of the current message.

Gaming machines clear the wakeup bit for all bytes when responding to the host, except when reporting a loop break condition (refer to 4.2 Loop Break Indication on page 4-1).

---

**Note:** *For UARTs/DUARTs that do not directly support wakeup mode, the parity bit can be used in place of the wakeup bit.*

---

## SECTION 2 COMMUNICATIONS

### 2.1 Gaming Machine Addressing

Gaming machines must support an attendant-configurable address with a range of 0 to 127. When configured with an address of 0, the gaming machine ignores **all** communications from the host. When a gaming machine suffers a critical memory error, it defaults its address to 0, thereby disabling communication until properly configured.

### 2.2 Host Polling

The two primary forms of polls that the host can use to interrogate the gaming machine are general and long. General polls are used to request event exceptions from a gaming machine. Long polls are used to request specific information from a gaming machine and to configure the gaming machine.

#### 2.2.1 General Polls

To request an event exception from a gaming machine, the host transmits a single-byte message consisting of the gaming machine's address ORed with 80 hex with the wakeup bit set. The addressed gaming machine can reply to a general poll by sending a single byte exception or a ROM signature verification long poll response. If no exceptions are pending, the gaming machine will respond with exception 00, no activity.

It is possible for a gaming machine to generate a series of exceptions at a rate that is faster than the polling cycle of the host. To accommodate this, gaming machines must maintain a first in/first out (FIFO) exception queue of at least 20 elements in non-volatile memory. In the event of exception queue overrun, the oldest exception is lost (subject to jurisdictional considerations). This ensures that the most recent exceptions are sent when requested by the host. If one or more exceptions have been lost from the queue, exception 70, buffer overflow, should be reported at the next opportunity. Exception 70 is not added to the queue. Once acknowledged, exception 70 is not reported again unless an exception is subsequently reported from the queue and then one or more exceptions have been lost.

Most exceptions indicate that an event has occurred on the gaming machine, such as a door opened or a tilt occurring. These exceptions are inserted in the exception queue in the order that the events are detected by the gaming machine. However, some exceptions are part of an interactive process with the host. Usually, interactive exceptions are intended to cause the host to send a particular long poll, and are reissued at some interval until the host polls for the particular data, or the condition requiring host interaction no longer exists. These exceptions are generally identified as priority exceptions and are not inserted in the exception queue. If the gaming machine has a priority exception pending and also an exception in the queue, the priority exception is always sent before the exception in the queue. When the protocol says to reissue an exception every 800 milliseconds, for example, this does **not** mean to insert another copy of the exception in the queue every 800 milliseconds. The correct procedure is to start a timer once the exception has been reported and acknowledged. If the timer expires and the condition requiring the exception to be reported still exists, the exception is then made pending again.

If multiple priority exceptions are pending at the same time, the gaming machine should generally report the exception first that relates to the most time sensitive task or most directly affects the player. The following list is a suggested guideline for a reasonable prioritization order, with the highest priority exceptions listed first.

57	System validation request
67	Ticket has been inserted
68	Ticket transfer complete
3F	Validation ID not configured
6A	AFT request for host cashout
6B	AFT request for host to cash out win
6F	Game locked
56	SAS progressive level hit
3D	A cash out ticket has been printed
3E	A handpay has been validated
69	AFT transfer complete
6C	AFT request to register
6D	AFT registration acknowledged
51	Handpay is pending
52	Handpay reset
8F	Authentication complete
70	Exception buffer overflow

Note that the host does not need to wait for the specified polling rate time period to respond to a priority exception with a long poll. Appendix A contains a complete list of currently defined exception codes.

## 2.2.2 Long Polls

Several types of long polls are available to communicate between the host and the gaming machines. Type R long polls are used to obtain basic gaming machine information. Type S long polls are used to send information to the gaming machine and to configure the gaming machine. Type M long polls are used to configure a specific game or obtain a specific game's information from a multi-game gaming machine. Type G long polls are sent by the host to multiple gaming machines simultaneously. These basic types of long polls are detailed below. For a complete list of long polls, refer to Appendix B.

### 2.2.2.1 Type R

This long poll type consists of the gaming machine address, with the wakeup bit set, followed by a single-byte command code. The gaming machine's response to type R long polls consists of its address, long poll command code, an optional length byte, requested data, and a two-byte message CRC.

### 2.2.2.2 Type S

The type S long poll consists of the gaming machine address, with the wakeup bit set, a single-byte command code, an optional length byte, optional data for the gaming machine, and a two-byte message CRC. When the gaming machine receives a type S long poll, it validates the message CRC and any message data. If the message is valid, the gaming machine acknowledges (ACKs) the host by one of two methods. Polls that do not request data from the gaming machine, such as game enable, are acknowledged by the gaming machine by transmitting its address. Polls that request data from the gaming machine are acknowledged by the gaming machine by transmitting its address, command code, requested data, and a two-byte message CRC. If the type S long poll is not received correctly by the gaming machine, the gaming machine will issue a negative acknowledgment (NACK) to the host by transmitting its address ORed with 80 hex, or ignore the message.

### 2.2.2.3 Type M

The type M long poll is a specialized form of the type S long poll detailed above. It consists of the gaming machine address, with the wakeup bit set, a single-byte command code, an optional length byte, a two-byte BCD game number, optional data for the gaming machine, and a two-byte message CRC. Upon receiving a type M long poll, the gaming machine validates the message CRC, any message data, and verifies that the received game number is within the valid range of available games on the gaming machine. If the message is valid, the gaming machine ACKs the host by one of two methods. Polls that do not request data from the gaming machine, such as enable/disable game n, are acknowledged by the gaming machine by transmitting its address. Polls that request data from the gaming machine are acknowledged by the gaming machine by transmitting its address, command code, two-byte BCD game number, requested data, and a two-byte message CRC. If the type M long poll is not received correctly, the gaming machine will issue a NACK to the host by transmitting its address ORed with 80 hex, or ignore the message. If the received game number is invalid or out of range, the gaming machine will ignore the message.

For multi-game gaming machines that allow only a subset of possible game types to be available to the player, game meters for all games implemented (as reported by long poll 51) must be available upon request by the host. Type M long polls containing a game number of zero indicate a request for the gaming machine data instead of a specific game's data.

---

**Note:** *Long poll 51 allows a host to determine the total number of games implemented on a multi-game gaming machine. Games must be assigned numbers from 0001 through the value returned by long poll 51, without gaps. The numbers assigned to games should not change dynamically. Any change in the relationship between paytables and game numbers must be accompanied by a change to the Paytable ID returned by long poll 1F.*

---

#### 2.2.2.4 Type G

To transmit data to all gaming machines simultaneously, the host can use the type G, or global broadcast, long poll. The type G long poll consists of a gaming machine address of 00 with the wakeup bit set, a single-byte command code, an optional length byte, data, and a two-byte message CRC.

Gaming machines do not ACK or NACK type G long polls. If the type G long poll is not received correctly by the gaming machine, it is ignored. Therefore, data transmitted via type G long polls should be transmitted periodically to ensure that all gaming machines receive it.

### 2.2.3 Transmitted Data Formats

Transmitted data, from both the host and from the gaming machine, can consist of any combination of packed binary coded decimal (BCD), ASCII, and binary formats. All data exchanged in the BCD and ASCII formats are sent most significant byte (MSB) first. All data exchanged in the binary format are sent least significant byte (LSB) first.

For variable length commands and/or responses, the length is a single binary byte that indicates the number of data bytes following the length byte. This length does not include the address, command, length or CRC bytes.

To allow for additional data to be added to variable length messages in future protocol revisions, the host and gaming machine must observe the following rules. A variable length command must always contain the number of bytes specified in the length byte, followed by a correct CRC, to be considered valid. When a gaming machine receives a valid variable length command with more data than is defined by the protocol, it should process the portion of the message it understands. Any extra bytes beyond the currently defined parameters must be ignored. Likewise, if a host receives a valid variable length response with more data than it expects, it will process the portion of the message it understands and ignore the extra bytes.

## 2.3 Timing Requirements

### 2.3.1 Gaming Machine Response Time

After a gaming machine has received an entire host message, it has 20 milliseconds (ms) in which to start transmitting its response. If the host has not begun receiving the gaming machine response after 20 ms, it may optionally time out the gaming machine and continue its polling cycle. Once a gaming machine has been timed out by the host, any message sent by that gaming machine is ignored.

### 2.3.2 Inter-Byte Delay Time

Inter-byte delay, the time between received bytes, cannot exceed 5 ms for both the host and gaming machine. If either host or gaming machine encounters an inter byte delay greater than 5 ms, the message may be considered invalid.

---

**Note:** *A host using fiber loop communications may enforce the inter-byte delay to prevent gaming machines from slowing communications to other gaming machines on the loop. For maximum compatibility, it is preferable that a gaming machine not enforce the delay on communications from the host.*

---

### 2.3.3 Polling Rate

The host may not issue general polls or long polls to any single gaming machine at a rate faster than once per 200 ms. The slowest allowable polling rate is 5000 ms (five seconds). The polling rate does not include the gaming machine response time or the inter-byte delay time for the host and gaming machine messages.

Note that some SAS features, such as RTE and ticketing, require the gaming machine to support a 40 ms polling rate. This will be indicated in the documentation of those features. Even if these features are not being used, it is recommended that a gaming machine support a 40 ms polling rate. Gaming machines capable of supporting a 40 ms polling rate should indicate this in the long poll A0 response.





## SECTION 3 HOST ACKNOWLEDGMENT

### 3.1 Implied Acknowledgment

An implied acknowledgment (ACK) concept is used to acknowledge data sent from the gaming machine to the host for both general and long polls. After the host performs a general or long poll, the gaming machine responds. If the host receives the gaming machine response correctly, it can perform an implied ACK to the gaming machine by any method detailed in Table 3.1. Once a gaming machine has received an implied ACK, it deletes the information from its transmit queue.

**Table 3.1**  
**Methods for Performing an Implied Acknowledgment**

Poll to ACK	Implied ACK
General	Issue a long poll to the same gaming machine
Long	Issue a general poll or a long poll with a different command byte to the same gaming machine
General or long	Issue a general or long poll to a different gaming machine address
General or long	Issue a global broadcast

**Note:** *While a poll to ANY different address serves as an ACK, hosts commonly use 80. This is a general poll to a gaming machine at address zero, and since no gaming machine can have address zero this allows a host to ACK a poll without actually polling a different gaming machine.*

### 3.2 Implied Negative Acknowledgment

If the host does not receive the gaming machine's response correctly, it repeats the general or long poll for the gaming machine. In this case, the host does not need to wait for the specified polling rate time period to re-issue the poll. This second consecutive poll is an implied negative acknowledgment (NACK) telling the gaming machine to re-send the requested information. If the host still does not receive the response correctly, a third and final poll is issued. To the gaming machine, a third consecutive poll is a final NACK. The gaming machine may respond to the poll but must not dispose of the volatile information. The host ignores any reply from the polled gaming machine and continues with its polling cycle.

### 3.3 Synchronization

Because the gaming machine must not delete volatile information after a third consecutive poll, the final NACK, the gaming machine needs to keep a counter of which poll state it is in. At startup, the gaming machine cannot just initialize the state counter to the first poll state because the host could be polling the gaming machine at that very moment. Therefore, after a warm or cold startup, or after any time when the gaming machine has not received any address byte for five seconds (see Section 4.2, loop break indication), the gaming machine needs to synchronize to the host polling cycle. Note that if the gaming machine loses synchronization while waiting for an implied ACK, this must be considered an implied NACK.

Synchronization to the polling cycle can be done in only one way. After startup or loop break detection, the gaming machine ignores any polls for itself and waits for another gaming machine to be polled. Once another gaming machine is polled, or a poll to address zero is seen, the state counter can be reset to the first polling cycle. The gaming machine can now respond the next time it is polled, knowing that it will be the first poll.

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH

## SECTION 4 ERROR CONDITIONS

### 4.1 Gaming Machine Busy Response

In the event that a gaming machine receives a long poll when it is processing a time-sensitive task (e.g., spinning reels or accepting a bill) it can respond to the host with a gaming machine busy response. This reply consists of the gaming machine address, followed by a 00 command code. Upon receiving a gaming machine busy response, the host aborts the long poll attempt and reinserts the long poll into its transmit queue for transmission at a later time.

### 4.2 Loop Break Indication

When a gaming machine does not receive any address byte (a byte with the wakeup bit set, regardless of the polling address) for five seconds, it “chirps” by transmitting its own address byte with the wakeup bit set every 200 ms. For gaming machines in a SMIB configuration, this indicates a failure in the gaming machine receive line. For gaming machines in a daisy chain configuration, this indicates that the communication loop is broken at a location just before the gaming machine that is chirping. Gaming machines located after the chirping machine will see the chirp as an address byte, and therefore will not chirp.

---

**Note:** *A gaming machine only chirps if it is not receiving any address bytes. A gaming machine must not chirp for any other link down condition.*

---

### 4.3 Link Down Detection

A gaming machine must consider the communications link to be down if it is not being actively polled by the host. At a minimum, the link must be considered down if the gaming machine has not received any address byte for five seconds (see Section 4.2, loop break indication), or has not received any implied acknowledgement (as defined in Section 3.1) from the host for 30 seconds.

### 4.4 Unsupported Long Polls

If a gaming machine receives a long poll it does not support, it must ignore the long poll and **not** NACK it. It is the responsibility of the host to determine which long polls are supported by the gaming machine.

### 4.5 Collisions

The gaming machine may only transmit data in response to a poll or when it is chirping (see Section 4.2, loop break indication). If the gaming machine is transmitting data or about to transmit data when it receives an address byte (a byte with the wakeup bit set), the gaming machine must abort its transmission immediately. To aid in duplicate address detection, the gaming machine must not abort its transmission simply because it receives a byte without the wakeup bit set.



## SECTION 5 CYCLICAL REDUNDANCY CHECK

### 5.1 Convention

The CRC follows the basic CCITT convention by starting with the most significant byte, least significant bit and applying the CRC polynomial  $x^{16}+x^{12}+x^5+1$ . Figure 5.1 details a fast CRC calculating routine from the public domain. The routine can be used to generate message CRCs as well as the variable-seed calculation needed for ROM signatures.

```
//Function:  CRC
//Purpose:   Calculate the 16-bit CRC of a string using
//           a byte-oriented tableless algorithm. The
//           routine inputs are the buffer pointer, the
//           buffer length, and the seed for the
//           calculation. The magic number 010201 octal
//           is derived from the CRC polynomial
//            $x^{16}+x^{12}+x^5+1$ .
//Passed in: unsigned char, int, unsigned short
//Passed out: unsigned short

unsigned short CRC(unsigned char *s, int len, unsigned
                  short crcval)
{
    register unsigned c,q;
    for (; len; len--)
    {
        c = *s++;
        q = (crcval ^ c) & 017;
        crcval = (crcval >> 4) ^ (q * 010201);
        q = (crcval ^ (c >> 4)) & 017;
        crcval = (crcval >> 4) ^ (q * 010201);
    }
    return (crcval);
}
```

**Figure 5.1**  
**Sample CRC Algorithm**

### 5.2 Host and Gaming Machine CRC Generation

The host calculates a CRC for all type S, type M and type G long polls. The CRC is calculated over the entire packet, including the address and command byte, with an initial seed value of zero. The gaming machine calculates the CRC in the same manner for all multi-byte long poll responses, except game busy.



## SECTION 6 ROM SIGNATURE

### 6.1 Verification

Any gaming machine may be required to perform a calculation to verify the contents of its game ROM(s) upon request. All of the gaming machine's program memory that influences game outcomes must be included in this calculation. The ROM signature calculation utilizes the 16-bit CRC algorithm, defined in Section 5, with the variable ROM verification seed.

The gaming machine receives a two-byte ROM verification seed to initiate a signature calculation using its ROM contents as data. The gaming machine reads its relevant ROM address space in a serial manner. For gaming machines with multiple byte wide ROMs, the signature of the first ROM is used as the seed for the second ROM, and so on. For gaming machines that utilize interleaved memory, the least significant byte of each word is used to calculate the signature of the lower ROM. The resultant signature is then used as the initial seed for calculating the signature over the most significant byte (i.e., upper ROM).

While performing this computation, the gaming machine must continue to respond to all communications. A gaming machine is expected to compute its signature as soon as possible after receiving the ROM verification seed. The ROM signature is returned to the host in response to the first general poll received after completing the signature calculation. This is a known exception to the rule for responding to a general poll, and the host takes care of this anomaly. If a second signature calculation request is received while a calculation is in progress or a ROM signature response is pending transmission, it supersedes the initial request.

### 6.2 LP 21: ROM Signature Verification

The ROM signature verification long poll is detailed in Table 6.2a. The gaming machine ACKs or NACKs the message, as detailed in Table 7.4b on page 7-5. Once the gaming machine has calculated the ROM signature, it sends the command detailed in Table 6.2b in response to the next general poll it receives.

**Table 6.2a**  
**ROM Signature Verification Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	21	ROM signature verification
Seed	2 binary	0000-FFFF	ROM verification seed
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 6.2b**  
**ROM Signature Verification Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	21	ROM signature verification
Signature	2 binary	0000-FFFF	ROM signature
CRC	2 binary	0000-FFFF	16-bit CRC



## SECTION 7 LONG POLL RESPONSE SPECIFICATIONS

### 7.1 Single Meter Accounting Long Polls

Many of the currently defined long polls request a single four-byte BCD meter from the gaming machine. Table 7.1a details the type R host command, and Table 7.1b details the gaming machine response. Some single meter polls are defined as multi-denom aware (see long poll preamble B0, Section 16.1) so these meters may also be retrieved for all games at a specific denomination. Please see Table 16.1d. For a complete list of single meter accounting long polls, refer to Appendix B.

**Table 7.1a**  
**Single Meter Accounting Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine to poll
Command	1 binary	?	Single meter accounting long poll

**Table 7.1b**  
**Single Meter Accounting Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	?	Single meter accounting long poll
Meter	4 BCD	00000000-99999999	four-byte BCD meter
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** *If a gaming machine does not support a meter, but knows the value must be zero, it should implement the meter poll and report a value of zero. For example, a gaming machine that is incapable of accepting \$100,000 bills can truthfully report a value of zero in response to long poll 44. However, a gaming machine capable of supporting a hopper, but not capable of tracking the number of coins in the hopper, must ignore long poll 2C, Send Current Hopper Level. A response of zero implies the hopper is empty, which may not be true.*

### 7.2 Multiple Meter Accounting Long Polls

Several long polls that allow the host to obtain multiple meters from the gaming machine by issuing a single long poll have been defined. The message format from the host is detailed in Table 7.1a. The response from the gaming machine varies, depending on which long poll the host sends. Each multiple meter accounting long poll response is detailed separately in Tables 7.2a through 7.2d.

**Table 7.2a**  
**Multiple Meter Long Poll 0F Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	0F	Send data defined in long polls 10~15
Total cancelled credits	4 BCD	XXXX	Total cancelled credits meter
Total coin in	4 BCD	XXXX	Total coin in meter
Total coin out	4 BCD	XXXX	Total coin out meter
Total drop	4 BCD	XXXX	Total drop meter
Total jackpot	4 BCD	XXXX	Total jackpot meter
Games played	4 BCD	XXXX	Games played meter
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 7.2b**  
**Multiple Meter Long Poll 19 Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	19	Send data defined in long polls 11~15
Total coin in	4 BCD	XXXX	Total coin in meter
Total coin out	4 BCD	XXXX	Total coin out meter
Total drop	4 BCD	XXXX	Total drop meter
Total jackpot	4 BCD	XXXX	Total jackpot meter
Games played	4 BCD	XXXX	Games played meter
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 7.2c**  
**Multiple Meter Long Poll 1C Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	1C	Send meters
Total coin in	4 BCD	XXXX	Total coin in meter
Total coin out	4 BCD	XXXX	Total coin out meter
Total drop	4 BCD	XXXX	Total drop meter
Total jackpot	4 BCD	XXXX	Total jackpot meter
Games played	4 BCD	XXXX	Games played meter
Games won	4 BCD	XXXX	Games won meter
Slot door opened	4 BCD	XXXX	Slot door opened meter
Power reset	4 BCD	XXXX	Power reset meter
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 7.2d**  
**Multiple Meter Long Poll 1E Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	1E	Send bill meters
\$1 bills accepted	4 BCD	XXXX	\$1 bills accepted meter
\$5 bills accepted	4 BCD	XXXX	\$5 bills accepted meter
\$10 bills accepted	4 BCD	XXXX	\$10 bills accepted meter
\$20 bills accepted	4 BCD	XXXX	\$20 bills accepted meter
\$50 bills accepted	4 BCD	XXXX	\$50 bills accepted meter
\$100 bills accepted	4 BCD	XXXX	\$100 bills accepted meter
CRC	2 binary	0000-FFFF	16-bit CRC

### 7.3 LP 2F: Send Selected Meters for Game N

Using the send selected meters command, the host can obtain up to ten meters by issuing a single long poll 2F. For ultimate flexibility, the host can select from the list of meters in Table C-7. All meters are reported using the number of BCD bytes listed as Min Size in Table C-7.

Long poll 2F is defined as a multi-denom-aware poll (see long poll preamble B0, Section 16.1), so some meters may also be retrieved for all games at a specific denomination. This variable length type M command is detailed in Table 7.3a. The variable length gaming machine response is detailed in Table 7.3b.

**Table 7.3a**  
**Send Selected Meters for Game N Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	2F	Send selected meters command
Length	1 binary	03-0C	Number of bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
Requested meter	1 binary	00-FF	Meter code for first requested meter (see Table C-7 in Appendix C for codes)
...	Variable	...	Additional meter codes (10 meters maximum per command)
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 7.3b**  
**Send Selected Meters for Game N Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	2F	Send selected meters command
Length	1 binary	02-3E	Number of bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
Meter code	1 binary	00-FF	Meter code for following meter (see Table C-7 in Appendix C for codes)
Meter value	x BCD	?	Meter value (use Min Size from Table C-7)
...	Variable	...	Additional meter code/meter value pairs
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** *To obtain terminal-wide meters, use game number 0000. It is possible that not all meters will be supported on all platforms, and that some meters that are supported on a terminal-wide basis may not be supported for individual games. If a gaming machine does not support a requested meter, the response will not contain a meter code/meter value pair for that meter. If none of the requested meters are supported, the length byte in the response will be 02, and no meter data will be returned. Be aware that some hosts may require a minimum set of supported meters.*

## 7.4 Enable/Disable Long Polls

Various aspects of the gaming machine can be enabled or disabled by the host. These include game play, sound, bill acceptor, and maintenance mode. The type S message format from the host includes an address, command, and message CRC, and it is detailed in Table 7.4a. When the gaming machine receives one of these long polls, it validates the message CRC and data, and if valid, ACKs the message. Otherwise the message is NACKed. The gaming machine response is detailed in Table 7.4b. For a complete list of enable/disable long polls, refer to Appendix B.

**Table 7.4a**  
**Host Enable/Disable Long Poll Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine to poll
Command	1 binary	01-07, 0A, 0B	Enable/disable command
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 7.4b**  
**Gaming Machine ACK/NACK Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F, 81-FF	Gaming machine address for ACK Gaming machine address ORed with 80 hex for NACK

#### 7.4.1 LP 01, 02: Shutdown (Lock Out Play) / Startup (Enable Play)

Long poll 01 is used to make a gaming machine unplayable (functionally equivalent to a tilt). Situations where a gaming machine may be disabled include preparing for casino maintenance, ROM signature mismatch, jurisdictional requirement, etc.

If a gaming machine is in the idle state when it receives the shutdown command, it should disable all user inputs except “cash out” and “change/attendant.” The gaming machine can either automatically cash out any accumulated credits or allow the user to cash them out.

If an active gaming machine receives the shutdown command, it must first complete the current game cycle, including any double up sequences. If there are any pending bonus awards, they are awarded upon completion of the game along with any base game win. If the win results in a handpay condition, the handpay condition is processed and reset normally. Once the gaming machine has completed processing the current game, it disables itself as detailed in the preceding paragraph.

Long poll 02, Startup, is used to remove the host’s shutdown control. This may or may not actually enable game play, depending on other conditions existing on the gaming machine. A host can only remove its own shutdown command. A startup command from one host must have no effect on a shutdown command from a different SAS host.

#### 7.4.2 LP 0A, 0B: Enter/Exit Maintenance Mode

Maintenance Mode is a feature used in some jurisdictions to allow the host to inform the gaming machine that an operator is properly logged into the system, to prevent a door open alarm on the gaming machine from sounding. Normally, long poll 0A is sent when an operator inserts a special card into the SMIB and enters a code. Long poll 0B is issued when the card is removed. Please consult your jurisdiction as to the need for this functionality. If you do not have a requirement to sound an alarm for unauthorized slot machine access, you probably do not need to implement this feature.

### 7.5 LP 08: Configure Bill Denominations

A special form of the enable/disable long poll, the configure bill denomination long poll allows the host to enable/disable the bill denominations independently of one another. This type S long poll from the host is detailed in Table 7.5. The gaming machine ACKs or NACKs this message, as detailed in Table 7.4b on page 7-5.

**Table 7.5**  
**Configure Bill Denominations Command**

Field	Bytes	Value	Description																																													
Address	1 binary	01-7F	Address of gaming machine to poll																																													
Command	1 binary	08	Configure bill denominations																																													
Bill Denominations	4 binary	????	Bill denominations sent LSB first (0 = disable, 1 = enable)																																													
			<table><tr><th>Bit</th><th>LSB</th><th>2nd Byte</th><th>3rd Byte</th><th>MSB</th></tr><tr><td>0</td><td>\$1</td><td>\$200</td><td>\$20000</td><td>TBD</td></tr><tr><td>1</td><td>\$2</td><td>\$250</td><td>\$25000</td><td>TBD</td></tr><tr><td>2</td><td>\$5</td><td>\$500</td><td>\$50000</td><td>TBD</td></tr><tr><td>3</td><td>\$10</td><td>\$1000</td><td>\$100000</td><td>TBD</td></tr><tr><td>4</td><td>\$20</td><td>\$2000</td><td>\$200000</td><td>TBD</td></tr><tr><td>5</td><td>\$25</td><td>\$2500</td><td>\$250000</td><td>TBD</td></tr><tr><td>6</td><td>\$50</td><td>\$5000</td><td>\$500000</td><td>TBD</td></tr><tr><td>7</td><td>\$100</td><td>\$10000</td><td>\$1000000</td><td>TBD</td></tr></table>	Bit	LSB	2nd Byte	3rd Byte	MSB	0	\$1	\$200	\$20000	TBD	1	\$2	\$250	\$25000	TBD	2	\$5	\$500	\$50000	TBD	3	\$10	\$1000	\$100000	TBD	4	\$20	\$2000	\$200000	TBD	5	\$25	\$2500	\$250000	TBD	6	\$50	\$5000	\$500000	TBD	7	\$100	\$10000	\$1000000	TBD
Bit	LSB	2nd Byte	3rd Byte	MSB																																												
0	\$1	\$200	\$20000	TBD																																												
1	\$2	\$250	\$25000	TBD																																												
2	\$5	\$500	\$50000	TBD																																												
3	\$10	\$1000	\$100000	TBD																																												
4	\$20	\$2000	\$200000	TBD																																												
5	\$25	\$2500	\$250000	TBD																																												
6	\$50	\$5000	\$500000	TBD																																												
7	\$100	\$10000	\$1000000	TBD																																												
Bill Acceptor Action Flag	1 binary	00-01	Action of bill acceptor after accepting a bill																																													
			<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>0 = Disable bill acceptor after each accepted bill 1 = Keep bill acceptor enabled after each accepted bill</td></tr><tr><td>1</td><td>TBD</td></tr><tr><td>2</td><td>TBD</td></tr><tr><td>3</td><td>TBD</td></tr><tr><td>4</td><td>TBD</td></tr><tr><td>5</td><td>TBD</td></tr><tr><td>6</td><td>TBD</td></tr><tr><td>7</td><td>TBD</td></tr></table>	Bit	Description	0	0 = Disable bill acceptor after each accepted bill 1 = Keep bill acceptor enabled after each accepted bill	1	TBD	2	TBD	3	TBD	4	TBD	5	TBD	6	TBD	7	TBD																											
Bit	Description																																															
0	0 = Disable bill acceptor after each accepted bill 1 = Keep bill acceptor enabled after each accepted bill																																															
1	TBD																																															
2	TBD																																															
3	TBD																																															
4	TBD																																															
5	TBD																																															
6	TBD																																															
7	TBD																																															
CRC	2 binary	0000-FFFF	16-bit CRC																																													

**Note:** *The gaming machine may be configured to ignore specific bills regardless of this message. Long poll 08 cannot enable bills that are otherwise disabled on the gaming machine.*

Long poll 08 (Configure Bill Denominations) provides a very different function than long polls 06 and 07 (Enable/Disable Bill Acceptor). Long polls 06 and 07 enable and disable the bill acceptor hardware. The bill acceptor must not accept any item into escrow when it is disabled. This fully disables both bill and voucher acceptance. When enabled, the bill acceptor may accept items into escrow. Its ability to accept vouchers and bills is then dependent upon the current configuration of the bill acceptor and any other factors controlling its behavior. Long polls 06 and 07 do not affect any configuration of the bill acceptor, including any configuration changes made using long poll 08.

Long poll 08 configures bill acceptance parameters, and allows individual bill denominations to be enabled and disabled. This determines the bill acceptor behavior when the bill acceptor hardware is enabled. If the bill acceptor is enabled but all bill denominations are disabled, then only vouchers can be accepted. If bill denominations are enabled while the bill acceptor is disabled, the new configuration will take effect the next time the bill acceptor is enabled.

If the bill acceptor is enabled and any bill denominations are enabled, vouchers plus the enabled denominations may be accepted by the gaming machine. All other bill denominations must be rejected. If the bill acceptor action flag is set to 01, bill acceptance does not change the configuration of the bill acceptor. If the bill acceptor action flag is set to 00, then after a bill is accepted the gaming machine will behave as though it had just received long poll 08 with all bill denominations disabled. Voucher acceptance is never affected by long poll 08, and acceptance of a voucher does not affect the configuration of the bill acceptor.

For compatibility with possible alternate implementations of long poll 08, it is recommended that a host send long poll 06 following every long poll 08. This will insure the bill acceptor hardware is enabled following the acceptance of a bill when the bill acceptor flag had previously been set to zero.

## 7.6 Multi-Game Long Polls

### 7.6.1 LP 09: Enable/Disable Game N

This type M long poll from the host, detailed in Table 7.6.1, specifies command code 09, the game number of the desired game, and the 1-byte binary flag indicating whether to enable or disable game n. Long poll 09 is defined as a multi-denom-aware poll (see long poll preamble B0, Section 16.1), so games may be enabled or disabled for a specific denomination. The gaming machine ACKs or NACKs this message, as detailed in Table 7.4b on page 7-5.

Table 7.6.1 Enable/Disable Game N Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	09	Enable/disable game n
Game number	2 BCD	0000-9999	Game number
Enable/Disable	1 binary	00-01	00 – Disable 01 – Enable
CRC	2 binary	0000-FFFF	16-bit CRC

Please see Appendix E for long poll 09 implementation guidelines.

### 7.6.2 LP 2D: Send Total Hand Paid Cancelled Credits

By issuing a type M long poll with a 2D command code, the host can request the total amount of hand paid cancelled credits for a specific game. These include all credits paid from the credit meter by an attendant handpay. They do not include any credits added to the jackpot meter. The command, detailed in Table 7.6.2a, specifies the game number of the desired game.

Table 7.6.2a Send Total Hand Paid Cancelled Credits Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	2D	Send total hand paid cancelled credits
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response, detailed in Table 7.6.2b, specifies the game number of the desired game along with a 4-byte BCD meter indicating the total number of hand paid cancelled credits.

**Table 7.6.2b**  
**Send Total Hand Paid Cancelled Credits Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	2D	Send total hand paid cancelled credits
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Hand paid credits	4 BCD	00000000-99999999	Total number of hand paid cancelled credits
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** *Send Total Hand Paid Cancelled Credits is defined as a multi-game poll. However, a gaming machine is not required to track cancelled credits for specific game numbers. If a gaming machine only tracks cancelled credits at the gaming machine level, it must ignore long poll 2D with a game number other than 0000.*

### 7.6.3 LP 51: Send Number of Games Implemented

The host may issue the type R long poll with a 51 command code to obtain the number of implemented games from a gaming machine. The gaming machine response to this long poll is detailed in Table 7.6.3 below:

**Table 7.6.3**  
**Send Number of Games Implemented Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	51	Send number of games implemented
Number of games	2 BCD	0000-9999	Total number of games implemented
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** *In response to long poll 51, gaming machines must send the total number of implemented games, not the number of games currently available to the player.*

The response to long poll 51 must indicate the total number of games implemented on the gaming machine. Games must be numbered from 0001 through the number in the long poll 51 response. The gaming machine must maintain an independent set of game play meters for each game.

If a gaming machine does not support multi-game extensions, it must respond with the number of games implemented equal to zero. It must also ignore all multi-game polls that specify a game number other than zero. However, for backwards compatibility, it must be understood that some gaming machines not supporting multi-game extensions will ignore long poll 51.



A gaming machine with only one game or payable may support multi-game extensions by responding to long poll 51 with the number of games implemented equal to one (0001). In this case, it would generally respond to multi-game polls that specify a game number of one with the same data as used to respond to game number zero. If more than one payable is available to the operator, and separate meters are maintained for each payable, a gaming machine must support multi-game extensions even if only one game can ever be available to the player at one time.

#### 7.6.4 LP 52: Send Game N Meters

By issuing a type M long poll with a 52 command code and specifying the desired game number, the host can request meters for a specific game in a multi-game gaming machine. The command, detailed in Table 7.6.4a, specifies the game number of the desired game.

Table 7.6.4a Send Game N Meters Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	52	Send game n meters
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

Table 7.6.4b details the gaming machine response.

Table 7.6.4b Send Game N Meters Response			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	52	Send game n meters
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Total coin in	4 BCD	XXXX	Total coin in meter for game n
Total coin out	4 BCD	XXXX	Total coin out meter for game n
Total jackpot	4 BCD	XXXX	Total jackpot meter for game n
Games played	4 BCD	XXXX	Games played meter for game n
CRC	2 binary	0000-FFFF	16-bit CRC

#### 7.6.5 LP 53: Send Game N Configuration

To obtain a specific game's information from a multi-game gaming machine, the host may issue a type M long poll with a 53 command code and the specific game number. The command is detailed in Table 7.6.5a.

**Table 7.6.5a**  
**Send Game N Configuration Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	53	Send game n configuration
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response is detailed in Table 7.6.5b.

**Table 7.6.5b**  
**Send Game N Configuration Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	53	Send game n configuration
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Game ID	2 ASCII	??	Game ID in ASCII (see Table C-1 in Appendix C)
Additional ID	3 ASCII	???	Additional ID in ASCII. If the gaming machine does not support an additional ID, this field should be padded with ASCII "0"s.
Denomination	1 binary	00-FF	Binary number representing the SAS accounting denom (see Table C-4 in Appendix C)
Max bet	1 binary	01-FF	Max bet for game n, in units of game credits, or FF if max bet greater than or equal to 255
Progressive group	1 binary	00-FF	Configured progressive group for game n. For EDT, stand alone, or non-progressive games, this field contains 0.
Game options	2 binary	0000-FFFF	Game options selected by the operator for game n. The bit configurations are dependent upon the type of gaming machine. (see Table C-2 in Appendix C)
Paytable ID	6 ASCII	??????	Paytable ID in ASCII for game n (see Table C-3 in Appendix C)
Base %	4 ASCII	??.??	Theoretical base pay back percentage for maximum bet in ASCII for game n. The decimal is implied and NOT transmitted.
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** The Additional ID field may be used to report additional Paytable ID characters, which allows up to nine total characters. If used for this purpose, unused characters should be padded with ASCII spaces.

*If the host issues the send game n configuration long poll with a 0000 game number, the information in the data fields must match the information returned in long poll response 1F.*

### 7.6.6 LP 55: Send Selected Game Number

The host may issue the type R long poll with a 55 command code to obtain the game number of the currently selected game on a multi-game gaming machine. If the gaming machine is in a game selection menu with no game currently selected when this long poll is received, it responds with game number zero (0000). The gaming machine response is detailed below in Table 7.6.6

**Table 7.6.6**  
**Send Selected Game Number Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	55	Send currently selected game number
Game number	2 BCD	0000-9999	Game number. A game number of 0000 indicates a gaming machine with no game currently selected.
CRC	2 binary	0000-FFFF	16-bit CRC

### 7.6.7 LP 56: Send Enabled Game Numbers

On multi-game gaming machines, only a subset of the games available to the operator for configuration may currently be available to the player. The host can issue a type R long poll with a 56 command code to obtain the game numbers of the game or games that are actually available to the player. For a multi-denom gaming machine, these will be the games enabled at the currently selected denomination. Long poll 56 is defined as a multi-denom-aware poll (see long poll preamble B0, Section 16.1), to allow the host to obtain the list of games enabled for any specific denomination. The variable length gaming machine response is detailed below in Table 7.6.7

**Table 7.6.7**  
**Send Enabled Game Numbers Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	56	Send enabled game numbers
Length	1 binary	01-FF	Number of bytes following, not including CRC
Number of games	1 binary	00-7F	Number of games currently enabled
Game number	Variable BCD	0001-9999 times number of games	2-byte BCD game number for each game currently enabled
CRC	2 binary	0000-FFFF	16-bit CRC

## 7.7 LP 18: Send Games Played Since Last Power Up and Slot Door Closure Long Poll

A variation of the multiple meter accounting long poll, the send games played since last power up and slot door closure long poll requires the gaming machine to respond with a pair of two-byte BCD meters. The host requests this data by sending a type R long poll with an 18 command code. The message format is identical to that detailed in Table 7.1a on page 7-1. The gaming machine response is detailed in Table 7.7.

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH

**Table 7.7**  
**Send Games Played Since Last Power Up and Slot Door Closure Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	18	Send games played since last power up and slot door closure
Games played since last power up	2 BCD	0000-9999	Games played since last power up meter
Games played since last slot door closure	2 BCD	0000-9999	Games played since last slot door closure meter
CRC	2 binary	0000-FFFF	16-bit CRC

## 7.8 LP 1B: Send Handpay Information

When the host receives exception 51 (i.e., handpay pending), it may request the handpay information by sending a type R long poll with a 1B command code. The gaming machine response to the send handpay information long poll is detailed in Table 7.8.

**Table 7.8**  
**Send Handpay Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	1B	Handpay information command
Progressive Group	1 binary	00-FF	Progressive group of the highest contributing progressive win for the handpay, if any 00 = Stand alone, non, or linked progressive 01-FF = Host controlled progressive
Level	1 binary	0-20, 40,80	Level of the highest contributing progressive win for the handpay, if any (01 = highest, 20 = lowest) 00 = Non progressive win amount 40 = Non-progressive top win amount (optional) 80 = Cancelled credits amount (in accounting units)
Amount	5 BCD	XXXXX	Total amount of the handpay. If any portion of the handpay is from a progressive win, the group and level are set according to the highest progressive contributor and the amount is in units of cents. If no portion of the handpay is from a progressive win, the amount is in SAS accounting denom units.
Partial pay	2 BCD	0000-9999	Any partial game win amount paid prior to the jackpot handpay in SAS accounting denom units.
Reset ID	1 binary	00-01	Available reset methods 00 – Only standard handpay reset is available 01 – Handpay reset to the credit meter is available
Session game win amount	5 BCD	XXXXX	Total win amount for a game within a session resulting in a jackpot lockup. If any portion of the total game win is from a progressive win, the amount is in units of cents. If no portion of the total game win is from a progressive win, the amount is in SAS accounting denom units.
Session game pay amount	5 BCD	XXXXX	Total positive amount added to the session balance for a game within a session resulting in a jackpot lockup. If any portion of the total game win is from a progressive win, the amount is in units of cents. If no portion of the total game win is from a progressive win, the amount is in SAS accounting denom units.
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** If the handpay amount does not include any progressive wins, the “amount” field in the 1B response indicates the amount of the handpay only, i.e., it does not include any partial pay amount. If the handpay amount includes one or more progressive wins, the “amount” field indicates the entire win amount, including the amount in the “partial pay” field.

The Handpay Amount only includes the amount that will actually be removed from the gaming machine if the handpay is reset normally. For session play jackpot wins, the actual win amounts for the game are reported in Session Game Win Amount and Session Game Pay Amount (see Section 18.2).

### 7.8.1 Handpay Queue

To prevent the loss of handpay information in the event that the SAS link is down and multiple handpays occur, the gaming machine must maintain an n-entry (minimum of 5) FIFO (first in/first out) handpay queue. When operating with the handpay queue, exceptions 51 and 52 are treated as priority exceptions. When a handpay occurs, the gaming machine stores all pertinent data required for the 1B long poll in the handpay queue and sends exception 51. If the handpay queue is already full, the oldest handpay record will be lost. When long poll 1B is received, the oldest unreported entry in the queue is sent to the host. When the gaming machine's response is acknowledged, this entry is marked as reported or removed from the queue.

If the 1B long poll is not received from the host within fifteen seconds after exception 51 has been sent and acknowledged, the gaming machine will re-issue exception 51 every fifteen seconds as long as the entry is in the queue.

Exception 52 is not sent until the corresponding 1B long poll has been received, responded to and acknowledged, and the associated handpay has been reset. If multiple handpays are queued, then after sending exception 52 and receiving an acknowledgement, the gaming machine will send another exception 51 and wait for another 1B long poll. This process repeats as necessary until all queued handpay entries have been reported. Long poll 1B returns all zeros if a handpay record has been reported and acknowledged, and a subsequent exception 51 has not been issued and acknowledged. If the final handpay has not been reset when the final handpay has been reported, exception 52 is not sent until the handpay is reset.

### 7.8.2 Legacy Handpay Reporting

The original handpay reporting behavior defined in SAS is to insert exception 51 in the exception queue when the gaming machine locks up in a handpay, and insert exception 52 in the queue when the handpay condition is reset. Any time the host sends the 1B long poll while the gaming machine is in a handpay lockup, the current handpay information is returned. Long poll 1B returns all zeros whenever the gaming machine is not currently in a handpay lockup.

Due to this original definition of the handpay exceptions, it must be understood that some systems may not know to poll for the 1B data in response to exception 51. In order to be compatible with such a system, a gaming machine must provide an operator configuration to enable legacy handpay reporting or otherwise disable the re-issuing of exception 51 every 15 seconds.

## 7.9 LP 94: Remote Handpay Reset

As an alternative to an attendant resetting a handpay condition, the host can remotely reset a handpay on a gaming machine by issuing a type S long poll with a 94 command code. The type S message format from the host includes an address, command, and message CRC, and it is detailed in Table 7.9a.

**Table 7.9a**  
**Reset Handpay Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine to poll
Command	1 binary	94	Reset handpay
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response, detailed in Table 7.9b, informs the host of its action using a reset code. If a gaming machine is not configured for remote handpay reset, it must ignore long poll 94.

**Table 7.9b**  
**Reset Handpay Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	94	Reset handpay
Reset code	1 binary	00-02	00 - Handpay was reset 01 - Unable to reset the handpay 02 - Not currently in a handpay condition
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** *If the W2-G Reset To Credit Meter function has been enabled using long poll A8, long poll 94 must reset the handpay to the credit meter instead of as an attendant pay.*

## 7.10 LP 1F: Send Gaming Machine ID and Information

To obtain specific information regarding the gaming machine, such as its max bet, denomination, payable information, progressive group, and game options, the host can issue a type R long poll with command code 1F to request gaming machine ID and information. The gaming machine response to this is detailed below in Table 7.10.



**Table 7.10**  
**Send Gaming Machine ID and Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	1F	Gaming machine information command
Game ID	2 ASCII	??	Game ID in ASCII. (see Table C-1 in Appendix C)
Additional ID	3 ASCII	???	Additional ID in ASCII. If the gaming machine does not support an additional ID, this field should be padded with ASCII "0"s.
Denomination	1 binary	00-FF	Binary number representing the SAS accounting denomination of this gaming machine (see Table C-4 in Appendix C)
Max bet	1 binary	01-FF	Largest configured max bet for the gaming machine, in units of game credits, or FF if largest configured max bet greater than or equal to 255
Progressive Group	1 binary	00-FF	Current configured progressive group for the gaming machine
Game options	2 binary	0000-FFFF	Game options selected by the operator The bit configurations are dependent upon the type of gaming machine. (see Table C-2 in Appendix C)
Paytable ID	6 ASCII	??????	Paytable ID in ASCII (see Table C-3 in Appendix C)
Base %	4 ASCII	??.??	Theoretical base pay back percentage for maximum bet in ASCII. The decimal is implied and NOT transmitted.
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** *The Additional ID field may be used to report additional Paytable ID characters, up to nine total characters. If used for this purpose, unused characters should be padded with ASCII spaces. For multi-game gaming machines, it is recommended that the Paytable ID and Additional ID fields be used to uniquely identify the set of paytables installed on the gaming machine.*

*For multi-game gaming machines in which the games available to the player are a subset of the total implemented games, the max bet field should contain the largest configured max bet for the games currently available to the player, and the base % field should contain an average of the theoretical percentage for the games currently available to the player.*

## 7.11 LP 48: Send Last Accepted Bill Information

When a gaming machine accepts a bill, it reports a corresponding bill accepted exception code (i.e., 47~4E, 50), or the general bill accepted exception 4F to the host (never both). In standard event reporting mode, exception 4F is only used if there is not a specific exception defined for the bill value. In RTE event reporting mode, it is preferred to always use exception 4F. Regardless of the exception reported, the bill information must be saved for the host to retrieve using long poll 48. The host, in response to the exception code, may poll the gaming machine for the bill information. It is up to the host to request

the bill information in a timely manner as the gaming machine only saves the most recently accepted bill information. If there has never been a bill accepted, all fields will be zero.

To request the last accepted bill information, the host issues a type R long poll with a 48 command code. The gaming machine response is detailed in Table 7.11.

**Table 7.11**  
**Send Last Accepted Bill Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	48	Send accepted bill information
Country code	1 BCD	00-38	Country code (See Table C-5 in Appendix C)
Denomination code	1 BCD	00-19	Bill denomination code (See Table C-6 in Appendix C)
Bill meter	4 BCD	XXXX	Number of accepted bills of this type
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** Older gaming machines that do not send exception 4F may not support long poll 48. It is up to the host to determine whether the gaming machine supports this long poll and to adjust its polling accordingly.

## 7.12 LP 8E: Send Card Information

To request a gaming machine's card information, the host issues a type R long poll with an 8E command code. The gaming machine response is detailed in Table 7.12a.

**Table 7.12a**  
**Send Card Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	8E	Send card information
Hand type	1 binary	00-01	00 - Dealt hand, 01 - Final hand
Hand	5 binary	0000000000-5E5E5E5E5E	Card data with the left most card sent first (see Table 7.12b for codes)
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** On gaming machines with multiple hands or more than five card positions, only the base hand or first five card positions can be reported.

Table 7.12b Card Codes			
Upper Nibble	Definition	Lower Nibble	Definition
0	Spades	0	Two
1	Clubs	1	Three
2	Hearts	2	Four
3	Diamonds	3	Five
4	Joker	4	Six
5	Other	5	Seven
		6	Eight
		7	Nine
		8	Ten
		9	Jack
		A	Queen
		B	King
		C	Ace
		D	Joker
		E	Other

### 7.13 LP 8F: Send Physical Reel Stop Information

The host can obtain a gaming machine's physical reel stop information by issuing a type R long poll with an 8F command code. The gaming machine response is detailed in Table 7.13.

Table 7.13 Send Physical Reel Stop Information Response			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	8F	Send physical reel stop information
Stops	9 binary	????????	Physical reel stop information with the left most reel sent first. Unused bytes are padded with FF.
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** On gaming machines with multiple paylines, the stops should be reported for the center payline. Alternately, the first or top payline may be used, but in any case the reporting should be consistent for all games reporting a particular game ID. If the gaming machine has more than nine reels, only the first nine reels can be reported

**7.14 LP A0: Send Enabled Features**

By issuing a type M long poll with an A0 command code, the host can interrogate numerous features of a gaming machine. The command, detailed in Table 7.14a, specifies the game number of the desired game.

<b>Table 7.14a Send Enabled Features Command</b>			
<b>Field</b>	<b>Bytes</b>	<b>Value</b>	<b>Description</b>
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	A0	Send enabled features
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

Table 7.14b details the gaming machine response.

<b>Table 7.14b Send Enabled Features Response</b>			
<b>Field</b>	<b>Bytes</b>	<b>Value</b>	<b>Description</b>
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	A0	Send enabled features
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Features1	1 binary	00-FF	Feature codes 1 (see Table 7.14c)
Features2	1 binary	00-FF	Feature codes 2 (see Table 7.14d)
Features3	1 binary	00-FF	Feature codes 3 (see Table 7.14e)
Features4	1 binary	00-FF	Feature codes 4 (see Table 7.14f)
Reserved	2 TBD	0's	Reserved.
CRC	2 binary	0000-FFFF	16-bit CRC

<b>Table 7.14c Feature Codes 1</b>	
<b>Bit</b>	<b>Description</b>
0 - Jackpot multiplier	0 = Disabled or not supported, 1 = Enabled
1 - AFT bonus awards	0 = Disabled or not supported, 1 = Enabled
2 - Legacy bonus awards	0 = Disabled or not supported, 1 = Enabled
3 - Tournament	0 = Disabled or not supported, 1 = Enabled
4 - Validation extensions	0 = Not supported, 1 = Supported
6~5 - Validation style	00 = Standard or none 01 = System 10 = Secure Enhanced 11 = reserved
7 - Ticket redemption	0 = Disabled or not supported, 1 = Enabled

**Table 7.14d  
Feature Codes 2**

Bit	Description
1~0 - Meter model flag	00 = Meter model not specified 01 = Won credits metered when won 10 = Won credits metered when played or paid 11 = reserved
2 - Tickets to total drop and total cancelled credits	0 = Not specified, 1 = Included (Note, tickets must always be included in total drop and total cancelled credits)
3 - Extended meters	0 = Not supported, 1 = Supported
4 - Component Authentication	0 = Not supported, 1 = Supported
5 - Jackpot keyoff to machine pay exception	0 = Not supported, 1 = Supported
6 - Advanced Funds Transfer	0 = Not supported, 1 = Supported
7 - Multi-denom extensions	0 = Not supported, 1 = Supported

**Table 7.14e  
Feature Codes 3**

Bit	Description
0 – Maximum polling rate	0 = Not specified, 1 = 40 milliseconds (Note, older gaming machines that support a 40 ms polling rate are not guaranteed to set this bit. Gaming machines conforming to SAS 6.01 or greater must set this bit to 1 if they support a 40 ms rate)
1 – Multiple SAS progressive win reporting (long poll 87)	0 = Not supported, 1 = Supported
2 – Meter change notification	0 = Not supported, 1 = Supported
3 –Reserved	0 (reserved)
4 – Session play	0 = Not supported, 1 = Supported
5 – Foreign currency redemption	0 = Not supported, 1 = Supported
6 – Non-SAS progressive hit reporting	0 = Not supported, 1 = Supported
7– Enhanced progressive data reporting	0 = Not supported, 1 = Supported

**Table 7.14f  
Feature Codes 4**

Bit	Description
0 – Max Progressive Payback	0 = Not enforced, 1 = Enforced
7~1 - –Reserved	0 (reserved)

**Note:** Reserved bits must always be set to zero when transmitting. No assumptions can be made about reserved/undefined bits when receiving.

## 7.15 LP 54: Send SAS Version ID and Gaming Machine Serial Number

To obtain a gaming machine's serial number and the SAS version that it supports, the host can issue a type R long poll with a 54 command code. The variable length gaming machine response, detailed in Table 7.15, will include the data length of the message, supported SAS version and its serial number.

**Table 7.15**  
**Send SAS Version ID and Gaming Machine Serial Number Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine
Command	1 binary	54	Send SAS version ID and gaming machine serial number
Length	1 binary	03-2B	Number of bytes following, not including the CRC
SAS version	3 ASCII	XXX	Implemented SAS version number
Gaming machine serial number	variable ASCII	XXX...	Gaming machine serial number (0 to 40 bytes)
CRC	2 binary	0000-FFFF	16-bit CRC

### 7.16 LP A4: Send Cash Out Limit

The cash out limit, defined as the largest amount, in units of the SAS accounting denom, that the gaming machine can pay from the hopper without locking up in a handpay, can be obtained by the host by issuing a type M long poll with an A4 command code. The command, detailed in Table 7.16a, specifies the game number of the desired game.

**Table 7.16a**  
**Send Cash Out Limit Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	A4	Send cash out limit
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response is detailed in Table 7.16b.

**Table 7.16b**  
**Send Cash Out Limit Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	A4	Send cash out limit
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Cash out limit	2 BCD	0000-9999	Cash out limit in SAS accounting denom units, sent MSB first
CRC	2 binary	0000-FFFF	16-bit CRC

### 7.17 LP 7F: Receive Date and Time

When the host desires to synchronize all gaming machines to the same real time clock, it can use the type G global broadcast detailed in Table 7.17. Gaming machines do not respond to global broadcasts. Long poll 7F can also be sent to any single gaming machine as a type S poll. When received as a type S poll, the gaming machine ACKs or NACKs this message, as detailed in Table 7.4b on page 7-5.

**Table 7.17**  
**Receive Date and Time Command**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Global broadcast or gaming machine address
Command	1 binary	7F	Receive date and time
Date	4 BCD	XXXX	Date in MMDDYYYY format
Time	3 BCD	XXX	Time in HHMMSS 24-hour format
CRC	2 binary	0000-FFFF	16-bit CRC

### 7.18 LP 7E: Send Current Date and Time

The host can issue a type R long poll with a 7E command code to read a gaming machine's current date and time. The response to this long poll is detailed in Table 7.18.

**Table 7.18**  
**Send Current Date and Time Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	7E	Send current date and time
Date	4 BCD	XXXX	Date in MMDDYYYY format
Time	3 BCD	XXX	Time in HHMMSS 24-hour format
CRC	2 binary	0000-FFFF	16-bit CRC

### 7.19 LP 4F: Send Current Hopper Status

At any time, the host can obtain the current hopper status and level by issuing a type R long poll with a 4F command code. The host may use this long poll in response to exception 22 (coin-out tilt) to obtain further information about the hopper tilt status. The gaming machine response, detailed in Table 7.19a, includes the current status and percentage full. It may also optionally include the number of coins in the hopper, if this information is available to the gaming machine.

**Table 7.19a**  
**Send Current Hopper Status Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	4F	Send current hopper status command
Length	1 binary	02, 06	Number of bytes following, not including CRC 02 = only status and % full 06 = status, % full and level
Status	1 binary	00-FF	Hopper status (see Table 7.19b for status codes)
% Full	1 binary	00-64, FF	Current hopper level as 0-100%, or FF if unable to detect hopper level percentage
Level	4 BCD	XXXXXXXX	Current hopper level in number of coins/tokens, only if EGM able to detect (see length byte, above)
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** If a gaming machine reports both % full and level, these values may be derived from separate systems, and have very different resolutions. Therefore, these two values should not be used to calculate how many coins it takes to fill the hopper.

**Table 7.19b**  
**Hopper Status Code Values**

Code (binary)	Status
00	Hopper OK
01	Flooded optics
02	Reverse coin
03	Coin too short
04	Coin jam
05	Hopper runaway
06	Optics disconnected
07	Hopper empty
08-FE	Reserved for future use
FF	Other



## 7.20 LP AA: Enable/Disable Game Auto Rebet

To configure a game to auto rebet (play continuously without customer interaction), the host issues the type S long poll detailed in Table 7.20. The gaming machine ACKs or NACKs this message, as detailed in Table 7.4b on page 7-5.

**Table 7.20**  
**Enable/Disable Game Auto Rebet Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine
Command	1 binary	AA	Enable/Disable game auto rebet feature
Enable/Disable	1 binary	00-01	00 – Disable auto rebet feature 01 – Enable auto rebet feature
CRC	2 binary	0000-FFFF	16-bit CRC

The following sequence will be necessary to start auto play:

- Auto Rebet command from host with Enable/Disable set to 01.
- The player starts a game before any of the auto rebet termination events occur. Auto rebet games play at the same bet options as the initiating game.

The following conditions must terminate Auto Rebet:

- Auto Rebet command from host with Enable/Disable set to 00.
- Link Down.
- Cashout button pressed. (Pressing the cashout button while auto play is active must not perform a cashout. The player must press cashout again after auto play stops to receive a cashout.)
- Attempt to perform a cashless lock or a transfer. (Cashless lock should go pending, a transfer request while auto rebet is active must be rejected.)
- Any handpay or other event requiring attendant attention.

Games that require user input to initiate bonus play should wait for and accept the user input, but not disable auto rebet.

## 7.21 LP 6F, AF: Send Extended Meters for Game N

To better address modern metering needs in the gaming industry, such as those presented by multi-denomination gaming machines, the following method is provided to communicate cumulative meters to the host that are up to 18 decimal digits in length. Existing long polls that communicate eight digit meters must continue to send the least significant eight digits of the requested meter. A gaming machine indicates its support of extended meters by setting Features2 bit 3 to one in its long poll A0 response.

Two different long poll codes can be used to access the exact same meter data. Two different codes are provided to allow a host to perform consecutive meter polls and still provide a proper implied acknowledgement in accordance with Section 3.1. Using the type M long poll 6F, Send Extended Meters, or long poll AF, Send Extended Meters (Alternate), the host can obtain up to 12 meters per poll. For ultimate flexibility, the host can select from the list of meters detailed in Table C-7. The length of the meters is not fixed as with long poll 2F. It is, however, recommended that meters accumulate at least as many digits as implied by the size column in Table C-7. Long polls 6F and AF are defined as multi-denom-aware polls (see long poll preamble B0, Section 16.1), so some meters may also be retrieved for all games at a specific denomination. These variable length commands are detailed in Table 7.21a.

**Table 7.21a**

**Send Extended Meters for Game N Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	6F, AF	Send extended meters command
Length	1 binary	04-1A	Number of bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
Requested meter code	2 binary	0000-FFFF	Meter code for first requested meter (see Table C-7 in Appendix C for codes)
...	variable	...	Additional meter codes (maximum 11 additional meter codes per command)
CRC	2 binary	0000-FFFF	16-bit CRC

The variable length gaming machine response is detailed in Table 7.21b.

**Table 7.21b**  
**Send Extended Meters for Game N Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	6F, AF	Send extended meters command
Length	1 binary	05-nn	Number of bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
Meter code	2 binary	0000-FFFF	Meter code for first meter (see Table C-7 in Appendix C for codes)
Meter size	1 binary	00-09	Meter size in number of bytes
Meter value	x BCD	???	Meter value for first meter (0 to 9 bytes)
...	variable	...	Code/size/value for additional meters
CRC	2 binary	0000-FFFF	16-bit CRC

To obtain terminal-wide meters, use game number 0000. It is possible that not all meters will be supported on all platforms, and that some meters that are supported on a terminal-wide basis may not be supported for individual games or denominations.

Meters are transmitted as three to 12 bytes per meter. The first two bytes are the meter code from Table C-7, indicating the specific meter being transmitted. The third byte indicates the size of the meter in number of bytes. The size not only indicates the number of bytes of meter data being transmitted, but also implies the maximum number of digits in the meter, i.e. meter rollover. The size will be zero if the meter requested is not supported by the gaming machine. In this case no meter data is included. Unlike long poll 2F, the response to 6F or AF must include at a minimum the meter code and a size byte for every meter specified in the 6F or AF command, unless including the meter would cause the maximum number of meters or the maximum message length to be exceeded. Also note that meter codes beyond FF are not available using long poll 2F.

In allowing the gaming machine to specify its meter size, it is important to understand meter rollover. It is expected that cumulative meters by nature have a maximum capacity, and the potential to roll over. It is also expected that the maximum capacity of any meter can be expressed as a string of decimal nines, for example 99,999,999 for an eight digit meter. It is further expected that the maximum capacity is reasonably fixed by game design issues, and will not change dynamically. A meter that rolls over at 99,999,999 is said to have a maximum capacity of eight digits, and would therefore always have a size

byte of 04, and be transmitted as 4 BCD bytes of meter data. A meter with a maximum capacity of 12 digits would always have a size byte of 06 and be transmitted as six BCD bytes.

Please note, if a meter's current value is 99,999,999, for example, and adding one to that meter would result in a value of 100,000,000, the meter obviously does not roll over at eight digits. Meters must always roll over at a value that is an even number of all nines. The size of the meter must always reflect the maximum number of digits before the meter rolls over.

To accommodate future protocol revisions, gaming machines must not attempt to enforce the 12 meter limit of the 6F or AF command by ignoring or NACKing the command. If a host requests more than 12 meters, the gaming machine must respond with meters to the best of its ability. It is permitted to ignore meter codes beyond the 12 meter limit, or respond with all requested meters so long as the maximum length of the response is not exceeded. If for any reason the maximum length of the response would be exceeded by including all requested meters, or the gaming machine is unable to format more than 12 meters in the required response time, extra meter codes that would cause the length or time to be exceeded may be ignored.

If the gaming machine transmits meters with more than 9 bytes of meter data, the host is free to ignore the extra most significant bytes.

## 7.22 LP B3: Send Token Denomination

The host may use the type R long poll B3 to determine what the current coin mechanism and/or hopper denomination is. The gaming machine response to long poll B3 is detailed in Table 7.22.

Table 7.22 Send Token Denomination Response			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B3	Send token denomination
Token denomination	1 binary	00-3F	Binary number representing the token denomination (see Table C-4 in Appendix C) 00 = no configured token value
CRC	2 binary	0000-FFFF	16-bit CRC

## 7.23 LP B5: Send Extended Game N Information

The type M long poll B5 allows a host to retrieve additional data for the gaming machine or a specific game, as a supplement to the legacy game information long polls 1F and 53. Long poll B5 is defined as a multi-denom-aware poll (see long poll preamble B0, Section 16.1), so this information may also be retrieved for all games at a specific denomination, or a specific game at a specific denomination. Because the Max Bet response is 2 BCD bytes, larger max bet values can be accommodated than is possible with long polls 1F and 53. The B5 command, detailed in Table 7.23a, specifies the game number of the desired game.

**Table 7.23a**  
**Send Extended Game N Information Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B5	Send extended game n information
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll B5 is detailed in Table 7.23b.

**Table 7.23b**  
**Send Extended Game N Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B5	Send extended game n information
Length	1 binary	09-nn	Total length of the bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Max bet	2 BCD	0000-9999	Max bet for game n, in units of game credits
Progressive group	1 binary	00-FF	SAS progressive group for game n
Progressive levels	4 binary	00000000-FFFFFFFF	SAS progressive levels enabled for game n (lsb = level 1, msb = level 32, bit set for each SAS progressive level enabled)
Game name length	1 binary	00-14	Length of game n name text
Game name	X ASCII	???	Optional ASCII name of game n or game family
Paytable name length	1 binary	00-14	Length of paytable name text
Paytable name	X ASCII	???	Optional ASCII name of paytable or collection of paytables
Wager categories	2 BCD	0000-9999	Number of wager categories supported
Num player denoms	1 binary	00-nn	Number of player denominations that game n can be configured for.
Player denom 1	1 binary	00-FF	First player denomination, if num player denoms is not zero (See Table C-4 in Appendix C)
...	variable	...	Additional player denominations, to num player denoms (See Table C-4 in Appendix C)
CRC	2 binary	0000-FFFF	16-bit CRC

When the game number is zero, all games supported by the gaming machine are considered in the response to long poll B5, whether they are currently enabled or not. When the denomination is not specified or is zero, all supported denominations are considered.

When the response is based on multiple games and/or multiple denominations, the progressive group is the SAS group number if any game considered is configured for a SAS progressive level, and the progressive levels will have bits set for all active SAS progressive levels for all games considered. The max bet field will contain the largest max bet value available to the player, which is zero for games and denominations that are not enabled.

The progressive levels field must never have any bits set for levels other than SAS progressive levels controlled by the polling host. If the progressive group field is zero, the progressive levels field will be zero.

The game name is an optional ASCII string identifying the game theme. For any response which considers multiple paytables, the response may identify the common game theme, if any, or the overall cabinet theme, if any.

The payable name is an optional ASCII string identifying a specific payable. For any response which considers multiple paytables, the payable name should identify the entire collection of paytables.

The wager categories field indicates how many wager categories within one payable have individual Coin In meters. If a payable has only one wager category, this field may be 0 or 1. For any response which considers multiple paytables, this field should be 0. A wager category of greater than 0 indicates long poll B4 is supported for that payable. See Section 7.24 for more information about wager categories.

The num player denoms field specifies the total number of player denominations that the payable can be enabled for. For game number 0000, the response specifies the total number of player denominations that can be enabled on the gaming machine. The response does not consider any multi-denom preamble to the long poll B5 or which denominations are currently enabled for play, but is based only on what is available to be configured by an operator. If num player denoms is not zero, the actual denominations are then listed, using denomination codes from Table C-4.

## 7.24 Weighted Average Theoretical Payback Percentage

If any single payable has a difference between the minimum and maximum theoretical payback percentage which exceeds some amount, the gaming machine may be required by jurisdictional rules to provide a calculated weighted average theoretical payback percentage for the system, or provide the data necessary for the system to calculate this value.

### 7.24.1 Calculated By Gaming Machine

Meter 007F in Table C-7 provides a method for the host to obtain the weighted average theoretical payback percentage as calculated by the gaming machine. Weighted average theoretical payback percentage is calculated by dividing the amount wagered at each different theoretical base payback percentage for the payable by the total amount wagered on that payable, multiplying the individual theoretical base payback percentage by this value, then summing the results (see calculation below). The value is returned in meter 007F as a percentage in hundredths of a percent.

WATP%<sub>Paytable</sub>: Weighted Average Theoretical Payback Percentage of a payable.  
 CI<sub>WCn</sub>: Wagers placed in wager category 'n'.  
 P<sub>WCn</sub>: Wager category 'n' payback percentage.  
 CI<sub>Paytable</sub>: Total sum of all wagers on a payable.  
 m: Total number of wager categories on a payable.

$$WATP \%_{Paytable} = \frac{\sum_{n=1}^m (CI_{WCn} \times P_{WCn})}{CI_{Paytable}}$$

If there is no jurisdictional requirement to report weighted average theoretical payback percentage, the gaming machine may optionally report the theoretical base payback percentage for max bet for that payable, or choose to not support meter 007F for that payable.

Gaming machines may also optionally report the overall weighted average theoretical payback percentage of a multi-game gaming machine. For a multi-game gaming machine, the weighted average theoretical payback percentage of the gaming machine may be calculated by dividing the amount wagered on each game by the total amount wagered on the gaming machine, multiplying the maximum theoretical base payback percentage or calculated weighted average theoretical payback percentage of the game by this value, then summing the results (see calculation below).

WATP%<sub>Cabinet</sub>: Weighted Average Theoretical Payback Percentage of the cabinet.  
 CI<sub>WC<sub>ni</sub></sub>: Wagers placed in wager category 'n' on Paytable 'i'.  
 P<sub>WC<sub>ni</sub></sub>: Wager category 'n' on payable 'i' payback percentage.  
 CI<sub>Cabinet</sub>: Total sum of all wagers on the cabinet.  
 j: Total number of paytables on the cabinet.

$$WATP \%_{Cabinet} = \frac{\sum_{i=1}^j \sum_{n=1}^m (CI_{WC_{ni}} \times P_{WC_{ni}})}{CI_{Cabinet}}$$

#### 7.24.2 LP B4: Send Wager Category Information

Long poll B4 allows the host to obtain the individual Coin In meters for each different payback percentage or each different number of credits wagered. Gaming machines may optionally maintain multiple wager categories even if there is little or no difference in payback percentages, or no jurisdictional requirement. If a gaming machine has one or more paytables with multiple wager categories, it will report the number of wager categories supported by each of those paytables in its long poll B5 response. The type M long poll B4 allows a host to retrieve the payback percentage and Coin In meter for each specific wager category. The B4 command, detailed in Table 7.24a, specifies the game number and wager category.

**Table 7.24a**  
**Send Wager Category Information Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B4	Send wager category information
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
Wager Category	2 BCD	0000-9999	Wager category (0000=total Coin In for game n)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll B4 is detailed in Table 7.24b.

**Table 7.24b**  
**Send Wager Category Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	B4	Send wager category information
Length	1 binary	09-nn	Number of bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
Wager Category	2 BCD	0000-9999	Wager category (0000=total Coin In for game n)
Payback percentage	4 ASCII	??.??	Theoretical payback percentage for the wager category in ASCII for game n. The decimal is implied and NOT transmitted.
Coin In meter size	1 binary	00-09	Coin In meter size in number of bytes
Coin In meter value	x BCD	???	Coin In meter value (0 to 9 bytes)
CRC	2 binary	0000-FFFF	16-bit CRC

For wager category 0, the payback percentage is the theoretical percentage for maximum wager, and the Coin In meter is the total Coin In for the payable. If the requested wager category is not supported for the requested game number, the payback percentage will be all nulls (0) and the Coin In meter size will be 0.





## SECTION 8 ADVANCED FUNDS TRANSFER

With the introduction of the SAS Advanced Funds Transfer Protocol (AFT), the EFT/ECT protocol first introduced in SAS 3.x has been removed from this specification. This does not prohibit a gaming machine that supports this version of SAS from supporting 3.x compatible EFT/ECT transactions. Note that EFT/ECT does not provide transactional control as AFT does, and is not recommended for managing a player's funds. If a gaming machine is configured with AFT enabled, it should ignore SAS EFT/ECT polls 22 through 25, 29, 62 through 65, 67, and 69.

AFT provides a robust, secure, highly auditable method for transferring funds between a host and a gaming machine. All transfers require that a non-zero "asset number," or house ID, be configured on the gaming machine by an operator to uniquely identify every gaming machine on a system, and every transfer must include the correct asset number. All transfers must also include a unique transaction ID. Transaction IDs must be composed of only printable ASCII characters in the range of 20 hex through 7E hex. It is the responsibility of the system to construct transaction IDs such that all transactions within a property can be uniquely identified and traced over a reasonable period of time.

Please note that the term "promotional," as used in this protocol, refers to amounts that are given to a player as incentive or reward. The intent is to distinguish these amounts from a player's regular funds for the purpose of casino accounting, in that promotional awards that are wagered may be taxed differently from those promotional amounts that are not wagered.

The term "restricted," as used in this protocol, refers to promotional amounts that are not redeemable for cash, and must be wagered. This is equivalent to what has been called "promotional" in the SAS EFT/ECT protocol. Restricted promotional amounts may only be removed from a gaming machine by methods that preserve the restricted status of the amounts, such as by transferring restricted amounts to the host or printing restricted tickets. They must never be cashed out on a normal cashout ticket, from the hopper, or by an attendant handpay.

The term "nonrestricted," as used in this protocol, refers to promotional amounts that may be redeemed for cash but have special accounting requirements. Nonrestricted promotional amounts that are played are tracked separately from those that are cashed out without being played. Whenever nonrestricted promotional amounts are cashed out by a method that cannot preserve the nonrestricted promotional status, they become regular cashable amounts.

The term "cashable," as used in this protocol, refers to amounts that may be redeemed for cash and have no special accounting requirements. Amounts described as cashable do not include nonrestricted promotional amounts. Note that the term cashable may also be used generically in this protocol to refer to the redeemable status of funds.

The term "regular cashable" may be used in this protocol to further clarify that the amounts do not include debit or nonrestricted promotional amounts. Note that debit transfers to the gaming machine always become regular cashable funds when they are added to the credit meter.

The term "total cashable" is used in this protocol when referring to the combined total of debit, cashable and nonrestricted promotional amounts. Note that when cashing out on a ticket, for example, nonrestricted amounts are automatically converted to cashable.

If any combination of restricted promotional amounts, nonrestricted promotional amounts and cashable amounts are in a gaming machine's credit meter at the same time, the restricted amounts must be played first, then the nonrestricted amounts, and finally the cashable amounts. Separate meters must track the cumulative restricted amount played, and the cumulative nonrestricted amount played.

Transfers are separated into three general categories: in-house, bonus and debit. Bonus transfers are win amounts awarded to a player by an external bonusing system. Debit transfers are those transactions where funds are transferred from a player's external bank account. All other transfers are considered in-house. The term "in-house" is not meant to be arbitrarily limiting. The protocol is not concerned with whether the funds are managed locally by the casino, or transferred to and from some form of wagering account maintained on behalf of the player, potentially accessible from more than one casino. A gaming machine may individually prohibit in-house, bonus and/or debit transfer types even if AFT has been enabled, in order to meet specific jurisdictional or system requirements.

Debit transfers (withdrawals from a player's bank account over an external financial network) require a gaming machine to be properly registered by the host. Registration for debit transfers includes a host-supplied non-zero AFT registration key and a non-zero Point of Sale terminal ID (POS ID). Gaming machines must be registered by the host before they can perform debit transfers. Debit transfer requests must include the current valid registration key.

Although system bonus awards use the funds transfer process, they are metered as game win rather than transfers to the gaming machine. They are not metered as part of "total transfers to the gaming machine" or "total in." Bonus amounts are handled exactly the same as normal game win amounts, and are always fully cashable. Bonus amounts transferred as nonrestricted, that are paid to the credit meter and are subsequently wagered, are metered in the Total Nonrestricted Played meter, and otherwise follow all of the standard rules for nonrestricted. Note that bonus transfers using AFT should have the same general player "look and feel" as legacy bonus transfers.

To perform AFT transfers, the gaming machine must maintain a one element buffer to track the current or most recent transfer request. In addition, the gaming machine must also maintain a circular history buffer of a maximum of 127 elements to store the most recent successfully completed transfers of a non-zero amount. Some systems or jurisdictions may have specific requirements for minimum history buffer size. The gaming machine indicates how many buffer positions it supports in its response to long poll 74, AFT Game Lock and Status Request. In lieu of other guidelines, a minimum of 70 positions is recommended.

Gaming machines that are configured with Advanced Funds Transfer enabled will set Features2 bit 6 to one in the long poll A0 response.

## 8.1 LP 73: AFT Register Gaming Machine

---

**Note:** *The registration process allows a debit system to associate a gaming machine with a POS terminal. Registration should not be used for in-house and bonus transactions. These transactions should use a registration key of all zeros.*

*The registration process is custom to each debit application. Please consult the systems provider that will be using the debit transfer functionality to obtain specific instructions on implementing long poll 73. Registration should not be used for AFT transactions other than debit.*

---

Before the host instructs a gaming machine to perform debit transfers, it must register the gaming machine using long poll 73. The variable length type S long poll 73 is detailed in Table 8.1a.

---

**Table 8.1a**

---

**AFT Register Gaming Machine Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine
Command	1 binary	73	AFT register gaming machine
Length	1 binary	01, 1D	Number of bytes following, not including CRC
Registration code	1 binary	nn	00 = Initialize registration 01 = Register gaming machine 40 = Request operator acknowledgement 80 = Unregister gaming machine FF = Read current registration
Asset number	4 binary	nnnnnnnn	Gaming machine asset number or house ID
Registration key	20 binary	nn...	Registration key
POS ID	4 binary	nnnnnnnn	Point of Sale terminal ID (0000 = no POS ID, FFFFFFF = no change)
CRC	2 binary	0000-FFFF	16-bit CRC

A gaming machine must maintain the current registration status and registration data in non-volatile memory. If the host requests to read the registration data or unregister the gaming machine, it will set the registration code to FF or 80 respectively, and omit the remaining fields. The gaming machine response to long poll 73 is detailed in Table 8.1b.

**Table 8.1b**  
**AFT Register Gaming Machine Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	73	AFT register gaming machine
Length	1 binary	1D	Number of bytes following, not including CRC
Registration status	1 binary	nn	00 = Gaming machine registration ready 01 = Gaming machine registered 40 = Gaming machine registration pending 80 = Gaming machine not registered
Asset number	4 binary	nnnnnnnn	Gaming machine asset number or house ID
Registration key	20 binary	nn...	Registration key
POS ID	4 binary	nnnnnnnn	Point of Sale terminal ID (0 = no POS ID)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response includes the current registration status, the current operator-entered asset number, the current or most recently received registration key, and the current or most recently received POS ID. If the gaming machine has not yet been configured with an asset number, the asset number field will be zero. If the gaming machine has not yet received a registration key or POS ID from the host, those fields will be zero. A gaming machine may not be registered until a valid non-zero asset number has been configured by an operator. A gaming machine may not perform debit transfers unless it has been properly registered and received a non-zero POS ID. A POS ID is never required for in-house transfers.

At any time, the host may interrogate the current registration status by setting the registration code to FF and omitting the remaining fields. The host may unregister the gaming machine or cancel a registration cycle at any time by sending long poll 73 with a registration code of 80 and omitting the remaining

fields. The gaming machine may choose to set its registration status to 80 for other reasons, such as a memory error detected, or an operator changing the asset number or AFT setup parameters or unregistering the gaming machine through a setup option provided for that purpose. The gaming machine must still respond with the most recently received registration key, even if it is not currently registered. Once registered, a gaming machine must not set its registration status to 80 simply because communication with the host has been lost, or a funds transfer operation has failed.

If the gaming machine, in response to an action by an operator, wants to initiate a registration cycle, it may issue exception 6C, AFT request to register. The host may deny registration by sending long poll 73 with a registration code 80. The host may initiate a registration cycle without first receiving exception 6C.

A registration cycle begins when the host sends long poll 73 with a registration code of 00. The asset number specified by the host must exactly match the current operator-entered non-zero value. The gaming machine then transitions to registration status 00. The host may set the registration key and/or POS ID to zero, which will cause any value previously stored by the gaming machine to be deleted.

After the gaming machine has responded with a registration status of 00, the host may simply complete the registration by sending long poll 73 with a registration code of 01, the gaming machine's correct asset number, and the desired registration key and POS ID. The final registration key must be non-zero. The gaming machine then sets its registration status to 01. Note that if the POS ID is zero, the gaming machine may be registered, but will be unable to perform debit transfers.

If the registration cycle was initiated by the gaming machine issuing exception 6C, indicating an operator is at the gaming machine, the host may optionally request an acknowledgement from the operator by sending long poll 73 with a registration code of 40. After receiving a valid acknowledgement request, the gaming machine will respond with a registration status of 40, registration pending, and wait for operator acknowledgement. The current gaming machine registration status must be 00 for it to transition to status 40. The registration data in the acknowledgement request is not required to be the same as any previously sent. Like the initiation poll, the acknowledgement request must always include the correct gaming machine asset number.

When the operator performs the acknowledgement step, the gaming machine will change its registration status back to 00 and issue exception 6D, AFT registration acknowledged. The host may then complete the registration by sending long poll 73 with a registration code of 01, the gaming machine's correct asset number, and the desired final registration key and POS ID.

At any time during the registration cycle, if the gaming machine receives a long poll 73 with a registration code of 80, unregister gaming machine, or determines that the link has gone down, or receives an operator request to cancel registration, the registration cycle will be cancelled and the registration status set to 80.

If the gaming machine receives a registration long poll 73 with an asset number that is not valid, or any long poll 73 is received that does not conform to the proper sequence and data, the gaming machine will set its registration status to 80 and preserve its existing registration data.

Whenever a gaming machine transitions from a state where the registration status is a value other than 80 to a state where the registration status is 80, it will issue exception 6E, AFT registration cancelled. This includes all conditions where a valid registration is cancelled or a registration in progress is cancelled, including when the cancellation is initiated by the host.

Please note that exceptions 6C, 6D and 6E are normally only issued to a host if the gaming machine is configured to perform AFT transactions with that host. If AFT is disabled on a gaming machine that is currently registered, or the AFT configuration is changed in any way that affects AFT behavior, the registration must be cancelled and exception 6E issued to the host that performed the registration.

If the gaming machine is configured to print registration reports, it may need to provide an option for an operator to print the report whenever a gaming machine is properly registered. See Section 8.11, Transaction Receipts, for details of the format of a registration report. Please contact your systems provider for specific information on any operator interface design requirements for the registration process.

## 8.2 LP 74: AFT Game Lock and Status Request

The host may interrogate the current AFT availability status at any time using the type S long poll 74. Long poll 74 may also be used to request a lock of gaming machine operation, in order to prevent the player from changing the gaming machine state until an AFT transfer can be initiated. The long poll 74 command is detailed in Table 8.2a.

Table 8.2a AFT Lock and Status Request Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine to poll
Command	1 binary	74	AFT lock and status request
Lock code	1 binary	nn	00 = Request lock 80 = Cancel lock or pending lock request FF = Interrogate current status only
Transfer condition	1 binary	00-FF	Bit For bit = 1, lock when condition true
			0 Transfer to gaming machine OK
			1 Transfer from gaming machine OK
			2 Transfer to printer OK
			3 Bonus award to gaming machine OK
			4 Leave as 0
			7~5 TBD (leave as 0)
Lock timeout	2 BCD	0000-9999	Lock expiration time in hundredths of a second
CRC	2 binary	0000-FFFF	16-bit CRC

If the host simply interrogates the current status (lock code = FF), the gaming machine will ignore the transfer condition and lock timeout fields. An interrogation poll must have no effect on any current or pending lock.

The gaming machine response to long poll 74 is detailed in Table 8.2b.

**Table 8.2b**  
**AFT Lock and Status Request Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	74	AFT lock and status request
Length	1 binary	23	Number of bytes following, not including CRC
Asset number	4 binary	nnnnnnnn	Gaming machine asset number or house ID
Game lock status	1 binary	nn	00 = Game locked 40 = Game lock pending FF = Game not locked
Available transfers	1 binary	00-FF	Bit Description
			0 1 = Transfer to gaming machine OK
			1 1 = Transfer from gaming machine OK
			2 1 = Transfer to printer OK
			3 1 = Win amount pending cashout to host
			4 1 = Bonus award to gaming machine OK
			6~5 TBD (leave as 0)
			7 Lock After Transfer request supported (deprecated)
Host cashout status	1 binary	00-FF	Bit Description
			0 0 = Cashout to host forced by gaming machine 1 = Cashout to host controllable by host
			1 0 = Cashout to host currently disabled 1 = Cashout to host currently enabled
			2 0 = Host cashout mode currently soft 1 = Host cashout mode currently hard (only valid if cashout to host is enabled)
			7~3 TBD (leave as 0)
			0 1 = Printer available for transaction receipts
			1 1 = Transfer to host of less than full available amount allowed
AFT status	1 binary	00-FF	2 1 = Custom ticket data supported
			3 1 = AFT registered
			4 1 = In-house transfers enabled
			5 1 = Bonus transfers enabled
			6 1 = Debit transfers enabled
			7 1 = Any AFT enabled
Max buffer index	1 binary	46-7F	Maximum transactions in history buffer

... Table 8.2b continued next page ...

**Table 8.2b - continued**  
**AFT Lock and Status Request Response**

Field	Bytes	Value	Description
Current cashable amount	5 BCD	XXXXXX	Current cashable amount on gaming machine, in cents
Current restricted amount	5 BCD	XXXXXX	Current restricted amount on gaming machine, in cents
Current non-restricted amount	5 BCD	XXXXXX	Current nonrestricted amount on gaming machine, in cents
Gaming machine transfer limit	5 BCD	XXXXXX	Maximum amount that may currently be transferred to the credit meter, in cents
Restricted expiration	4 BCD	XXXX	Current restricted expiration date in MMDDYYYY format or 0000NNNN days format, if restricted amount non-zero
Restricted pool ID	2 binary	0000-FFFF	Current restricted pool ID, if restricted non-zero
CRC	2 binary	0000-FFFF	16-bit CRC

All responses to long poll 74 include the current gaming machine status at the time of the response. If AFT is disabled on the gaming machine, the available transfers and AFT status fields will be zero. Otherwise, the AFT status bits are set according to the current configuration of the gaming machine, and each of the available transfer bits will be one if the gaming machine is currently in a state where it might be able to accept the particular transfer type. The gaming machine must always set these bits, and the amounts, based on its current state. For example, the “transfer to printer OK” bit would be one if the gaming machine is currently configured with a printer, the printer is currently in a state where it can be used as a cashout device, and the gaming machine is currently in a state where it can perform or escrow a cashout request. This in no way guarantees, however, that the states and amounts will be the same when a subsequent transfer request is received. In addition, the current registration status is not considered in determining currently available transfers.

The host may request a gaming machine lock prior to performing an AFT transfer, to hold the gaming machine in a state where it is reasonably expected to be able to perform the transfer. From the host perspective, the game lock status may be in one of three states:

FF = Not locked (gaming machine not locked, no lock pending)  
 40 = Lock pending (gaming machine not locked, lock request in process)  
 00 = Locked (gaming machine currently locked)

The gaming machine must never enter the locked state if AFT is disabled, an AFT transfer cycle is currently in progress, the gaming machine is in a condition where it cannot be played and cannot accept any transfers such as door open, operator menu, tilt, disabled, waiting for handpay, etc., or a lock is requested that is not possible in the current configuration. If the gaming machine goes into such a condition or enters a link down state while locked or lock pending, it must transition to the not locked state.

If the host requests a gaming machine lock (lock code = 00), it must specify a transfer availability condition to be met in order for the lock to occur. When a lock request is received while the gaming machine is not locked, the gaming machine may reject the lock and respond with lock status FF, it may immediately enter the lock state and respond with lock status 00, or it may transition to lock status 40, lock pending. This allows the gaming machine to actually process the lock request after its initial response. If the game is playable but the requested transfer availability condition is not met, the gaming machine may stay in the lock pending state indefinitely. It is also possible that some transfer conditions

may never be available, for example “lock on transfer to printer OK” for a gaming machine without a printer. In this case the gaming machine may either refuse to enter the lock pending state, or return to the not locked state once the request has been processed. If any AFT transfer cycle is initiated while the gaming machine lock is pending, the transfer will be processed as a transfer request while not locked. A subsequent lock status request (lock code = FF) will return lock status FF, not locked.

When a lock is pending and the requested transfer type is available, the gaming machine will transition into the locked state. Once the gaming machine is locked, it sets the lock status to 00, game locked, and starts a timer for the lock timeout duration specified in the lock command. While locked, the gaming machine should indicate to the player that a transaction is pending with a message such as “Please Wait.” If the lock timer expires, the gaming machine must exit the locked state. If any AFT transfer request (long poll 72) is received while the gaming machine is locked, the gaming machine should remain locked for the purpose of the transfer. However, because the lock request is a separate process from the transfer request, the lock status, as reported to a long poll 74 Lock and Status request, will return status FF, not locked, once the transfer request is received.

If a gaming machine receives a lock request while in a locked or lock pending state, the transfer condition and lock timeout values from the new lock request take precedence over any previous values. If the gaming machine is in a locked state when it receives the new lock request, it must not exit the locked state simply because it has not yet processed the new lock request. When the gaming machine evaluates the lock request while already locked, if the requested transfer condition is already met; the gaming machine must refresh its lock timer with the new timeout value and remain locked. In this way, the host can actively maintain a lock indefinitely. Otherwise the gaming machine will process the new request and transition to the appropriate state.

At its next opportunity to respond with an exception code after entering the game lock state, unless a higher priority exception is pending, the gaming machine will respond with exception 6F, game locked. Exception 6F is a priority exception, and must not be inserted in the exception queue. It must only be issued if the gaming machine is currently locked. Exception 6F must be reissued every five seconds as long as the gaming machine is locked.

If the host requests the lock to be cancelled (lock code = 80), or issues any lock code the gaming machine does not support, the transfer condition and lock timeout fields will be ignored. The gaming machine must respond with status FF, not locked, and cancel any current or pending lock.

### **8.3 LP 72: AFT Transfer Funds**

The host may use long poll 72 to transfer funds to or from the gaming machine, or instruct the gaming machine to print a ticket for a specified amount. For any of these transfers, the host may request the gaming machine to print a transaction receipt. The host may also use long poll 72 to award a bonus win amount. The variable length type S long poll 72 is detailed in Table 8.3a.



**Table 8.3a**  
**AFT Transfer Funds Initiate Command**

Field	Bytes	Value	Description	
Address	1 binary	01-7F	Address of gaming machine to poll	
Command	1 binary	72	AFT transfer funds	
Length	1 binary	01-nn	Number of bytes following, not including CRC	
Transfer code	1 binary	nn	00 = Transfer request, full transfer only 01 = Transfer request, partial transfer allowed 80 = Cancel transfer request	
Transaction index	1 binary	00	Only “current” transaction may be initiated	
Transfer type	1 binary	nn	Transfer type (see Table 8.3d)	
Cashable amount	5 BCD	XXXXX	Cashable transfer amount requested, in cents	
Restricted amount	5 BCD	XXXXX	Restricted transfer amount requested, in cents	
Nonrestricted amount	5 BCD	XXXXX	Nonrestricted transfer amount requested, in cents	
Transfer flags	1 binary	00-FF	Bit	Description
			0	Host cashout enable control 0 = Ignore bits 1 and 2 1 = set host cashout per bits 1 and 2
			1	0 = Host cashout disable 1 = Host cashout enable (ignore if bit 0 = 0)
			2	0 = Host cashout mode soft 1 = Host cashout mode hard (ignore if bit 0 = 0)
			3	1 = Cashout from gaming machine request
			4	Lock After Transfer request (deprecated)
			5	1 = Use custom ticket data (long poll 76)
			6	1 = Accept transfer only if locked
			7	1 = Transaction receipt request
Asset number	4 binary	nnnnnnnn	Gaming machine asset number or house ID	
Registration key	20 binary	nn...	Registration key (0 = registration not required)	
Transaction ID length	1 binary	01-14	Length of message transaction ID	
Transaction ID	x ASCII	???	Transaction ID ASCII text (1 to 20 bytes)	
Expiration	4 BCD	XXXX	Expiration date in MMDDYYYY format or 0000NNNN days format	
Pool ID	2 binary	0000-FFFF	Restricted pool ID	
Receipt data length	1 binary	nn	Number of bytes of receipt data following (Length zero if no data provided. Data may be provided even if no receipt is requested. Note that maximum overall message length must not be exceeded.)	
Receipt data	X bytes	???	Transaction receipt data (see Table 8.3f)	
Lock timeout	2 BCD	0000-9999	Lock expiration time in hundredths of a second, Only used for Lock After Transfer request (deprecated).	
CRC	2 binary	0000-FFFF	16-bit CRC	

**Note:** Please see Section 15, Validation, for details on expiration, pool ID, and the rules for combining restricted credits from different sources.

When initiating a transfer, the transaction index must be zero. The host may not specify the index for a new transaction.

Transfer requests may specify “full transfer only,” or “partial transfer allowed.” Full transfer requests require the gaming machine to either perform the entire transfer for the exact amount specified or reject it. With partial transfer allowed, the gaming machine is permitted to perform a transfer for any amount equal to or less than each specified amount. Due to jurisdictional or other considerations, some gaming machines may refuse to perform partial transfers even if the host specifies partial transfer allowed.

The total requested transfer amount is the sum of the cashable, restricted and nonrestricted amounts. Table 8.3d indicates which amount types are allowed for each supported transfer type. Amount fields must be zero for amount types not permitted for the selected transfer type.

The host may request a transaction receipt for transfers from the host to the gaming machine, transfers from the gaming machine to the host, and transfers from the host to a ticket. There is never a receipt for bonus award transfers. If the host requests a transaction receipt, it should provide additional information as necessary for the receipt. Some receipt information differs for in-house vs. debit transactions. Please see Table 8.3f. If the host requests a receipt and the gaming machine knows it will be unable to produce a receipt before it begins the transfer, it will reject the transfer request. The host may also specify that a transfer must only be accepted if the gaming machine is currently locked (using long poll 74) at the time the transfer is initiated. It does not matter which lock condition was requested, only that the gaming machine is locked and able to perform the requested transfer.

The host may also use long poll 72 to interrogate the status of the current or most recently completed transfer requests. Long poll 72 also allows the host to retrieve from the history buffer up to 127 of the most recent funds transfers that were successfully completed for a non-zero amount. The variable length interrogation long poll 72 is detailed in Table 8.3b.

Table 8.3b AFT Transfer Funds Interrogation Poll			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine to poll
Command	1 binary	72	AFT transfer funds
Length	1 binary	02	Number of bytes following, not including CRC
Transfer code	1 binary	FE, FF	Identify poll as interrogation request
Transaction index	1 binary	00, 01-7F, 81-FF	00 = current or most recent transaction 01-7F = absolute history buffer position 81-FF = relative history index
CRC	2 binary	0000-FFFF	16-bit CRC

Transfer code FE is identical to FF, except a response to an FE interrogation does not in any way affect the current transfer cycle, even when reporting the current transaction. Systems designers should be aware that gaming machines not supporting an FE transfer code will respond with a Transfer Status C1, unsupported transfer code.

The gaming machine response to long poll 72 is detailed in Table 8.3c.

**Table 8.3c**  
**AFT Transfer Funds Response**

Field	Bytes	Value	Description																		
Address	1 binary	01-7F	Address of gaming machine responding																		
Command	1 binary	72	AFT transfer funds																		
Length	1 binary	02-nn	Number of bytes following, not including CRC																		
Transaction buffer position	1 binary	00-FF	Specific transaction history buffer position (0 = current or most recent transaction, not in history buffer)																		
Transfer status	1 binary	nn	Gaming machine transfer status code (see Table 8.3e)																		
Receipt status	1 binary	nn	Transaction receipt status code (see Table 8.3g)																		
Transfer type	1 binary	nn	Transfer type (see Table 8.3d)																		
Cashable amount	5 BCD	XXXXX	Actual or pending cashable transfer amount, in cents																		
Restricted amount	5 BCD	XXXXX	Actual or pending restricted transfer amount, in cents																		
Nonrestricted amount	5 BCD	XXXXX	Actual or pending nonrestricted transfer amount, in cents																		
Transfer flags	1 binary	00-FF	<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>0 = Cashout to host forced by gaming machine 1 = Cashout to host controllable by host</td></tr><tr><td>1</td><td>0 = Cashout to host currently disabled 1 = Cashout to host currently enabled</td></tr><tr><td>2</td><td>0 = Host cashout mode currently soft 1 = Host cashout mode currently hard (only valid if cashout to host is enabled)</td></tr><tr><td>3</td><td>0 = Host did not request cashout from gaming machine 1 = Host requested cashout from gaming machine</td></tr><tr><td>4</td><td>0 = no Lock After Transfer request 1 = Lock After Transfer requested (deprecated)</td></tr><tr><td>5</td><td>Custom ticket data requested</td></tr><tr><td>6</td><td>0 = Host does not require lock 1 = Host requested transfer only if locked</td></tr><tr><td>7</td><td>0 = No transaction receipt 1 = Transaction receipt requested</td></tr></table>	Bit	Description	0	0 = Cashout to host forced by gaming machine 1 = Cashout to host controllable by host	1	0 = Cashout to host currently disabled 1 = Cashout to host currently enabled	2	0 = Host cashout mode currently soft 1 = Host cashout mode currently hard (only valid if cashout to host is enabled)	3	0 = Host did not request cashout from gaming machine 1 = Host requested cashout from gaming machine	4	0 = no Lock After Transfer request 1 = Lock After Transfer requested (deprecated)	5	Custom ticket data requested	6	0 = Host does not require lock 1 = Host requested transfer only if locked	7	0 = No transaction receipt 1 = Transaction receipt requested
Bit	Description																				
0	0 = Cashout to host forced by gaming machine 1 = Cashout to host controllable by host																				
1	0 = Cashout to host currently disabled 1 = Cashout to host currently enabled																				
2	0 = Host cashout mode currently soft 1 = Host cashout mode currently hard (only valid if cashout to host is enabled)																				
3	0 = Host did not request cashout from gaming machine 1 = Host requested cashout from gaming machine																				
4	0 = no Lock After Transfer request 1 = Lock After Transfer requested (deprecated)																				
5	Custom ticket data requested																				
6	0 = Host does not require lock 1 = Host requested transfer only if locked																				
7	0 = No transaction receipt 1 = Transaction receipt requested																				
Asset number	4 binary	nnnnnnnn	Gaming machine asset number or house ID																		
Transaction ID length	1 binary	01-14	Length of message transaction ID																		
Transaction ID	x ASCII	???	Transaction ID ASCII text (1 to 20 bytes)																		
Transaction date	4 BCD	XXXX	Date transaction completed in MMDDYYYY format																		
Transaction time	3 BCD	XXX	Time transaction completed in HHMMSS 24-hour format																		

... Table 8.3c continued next page ...

**Table 8.3c - continued**  
**AFT Transfer Funds Response**

Field	Bytes	Value	Description
Expiration	4 BCD	XXXX	Expiration date for transfer to ticket or restricted amount in MMDDYYYY or 0000NNNN days format
Pool ID	2 binary	0000-FFFF	Restricted pool ID (0 if no restricted amount)
Cumulative cashable amount meter size	1 binary	00-09	Length of cumulative cashable amount meter for transfer type, after transfer complete (0 until complete)
Cumulative cashable amount meter	x BCD	???	Cumulative cashable amount meter for transfer type, in cents (0 to 9 bytes)
Cumulative restricted amount meter size	1 binary	00-09	Length of cumulative restricted amount meter for transfer type, after transfer complete (0 until complete)
Cumulative restricted amount meter	x BCD	???	Cumulative restricted amount meter for transfer type, in cents (0 to 9 bytes)
Cumulative nonrestricted amount meter size	1 binary	00-09	Length of cumulative nonrestricted amount meter for transfer type, after transfer complete (0 until complete)
Cumulative nonrestricted amount meter	x BCD	???	Cumulative nonrestricted amount meter for transfer type, in cents (0 to 9 bytes)
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 8.3d**  
**AFT Transfer Types**

Code (binary)	Transfer type	Allowed amount types		
		Cashable	Restricted	Nonrestricted
00	Transfer in-house amount from host to gaming machine	X	X	X
10	Transfer bonus coin out win amount from host to gaming machine	X		X
11	Transfer bonus jackpot win amount from host to gaming machine (force attendant pay lockup)	X		X
20	Transfer in-house amount from host to ticket (only one amount type allowed per transfer)	X	X	
40	Transfer debit amount from host to gaming machine	X		
60	Transfer debit amount from host to ticket	X		
80	Transfer in-house amount from gaming machine to host	X	X	X
90	Transfer win amount (in-house) from gaming machine to host	X	X	X

**Table 8.3e**  
**Gaming Machine AFT Transfer Status Codes**

Code (binary)	Transfer Status (Note, 3 MSbits can be used to determine category of status code)
<b>Binary codes 000xxxxx indicate transfer successful</b>	
00	Full transfer successful
01	Partial transfer successful
<b>Binary codes 010xxxxx indicate transfer pending</b>	
40	Transfer pending (not complete)

... Table 8.3e continued next page ...

**Table 8.3e - continued**  
**Gaming Machine AFT Transfer Status Codes**

<b>Code (binary)</b>	<b>Transfer Status</b> (Note, 3 MSbits can be used to determine category of status code)
<b>Binary codes 100xxxxx indicate transfer failed</b>	
80	Transfer cancelled by host
81	Transaction ID not unique (same as last successful transfer logged in history)
82	Not a valid transfer function (unsupported type, amount, index, etc.)
83	Not a valid transfer amount or expiration (non-BCD, etc.)
84	Transfer amount exceeds the gaming machine transfer limit
85	Transfer amount not an even multiple of gaming machine denomination
86	Gaming machine unable to perform partial transfers to the host
87	Gaming machine unable to perform transfers at this time (door open, tilt, disabled, cashout in progress, etc.)
88	Gaming machine not registered (required for debit transfers)
89	Registration key does not match
8A	No POS ID (required for debit transfers)
8B	No won credits available for cashout
8C	No gaming machine denomination set (unable to perform cents to credits conversion)
8D	Expiration not valid for transfer to ticket (already expired)
8E	Transfer to ticket device not available
8F	Unable to accept transfer due to existing restricted amounts from different pool
90	Unable to print transaction receipt (receipt device not currently available)
91	Insufficient data to print transaction receipt (required fields missing)
92	Transaction receipt not allowed for specified transfer type
93	Asset number zero or does not match
94	Gaming machine not locked (transfer specified lock required)
95	Transaction ID not valid
9F	Unexpected error
<b>Binary codes 110xxxxx indicate incompatible or unsupported poll</b>	
C0	Not compatible with current transfer in progress
C1	Unsupported transfer code
<b>Binary codes 111xxxxx indicate no transfer information available</b>	
FF	No transfer information available

**Table 8.3f**  
**Transaction Receipt Fields**

<b>Code (binary)</b>	<b>Description</b>	<b>Data</b>	<b>Predefined Label Text</b>
00	Transfer source/destination (in-house or debit)	Variable ASCII text (22 max)	None
01	Date and time (in-house or debit)	7 BCD (date in MMDDYYYY format followed by time in HHMMSS 24-hour format)	None
10	Patron name (in-house transaction only)	Variable ASCII text (22 max)	None
11	Patron acct# (required!) (in-house transaction only)	Variable ASCII text (16 max)	"Acct:"
13	Account Balance (in-house transaction only)	5 BCD (cents) (value BEFORE transaction!)	"Acct Bal" (print value + total transaction)
41	Debit card# (required!) (debit transaction only)	2 BCD (NNNN = last 4 digits of card#)	"Acct: xxxxxxxxxxxx"
42	Transaction fee (debit transaction only)	5 BCD (cents)	"Transaction Fee"
43	Total debit amount (debit transaction only)	5 BCD (cents)	"Total Debit"

Each transaction receipt data entry in the long poll 72 command consists of the code, a length byte, and the data. A length byte must always be included, even with fixed length data, to enable a gaming machine to skip over receipt data codes it does not understand.

**Table 8.3g**  
**Transaction Receipt Status Codes**

<b>Code (binary)</b>	<b>Transaction Receipt Status</b>
00	Receipt printed
20	Receipt printing in progress (not complete)
40	Receipt pending (not complete)
FF	No receipt requested or receipt not printed

A transfer cycle begins when the gaming machine receives an initiating 72 long poll, and ends when the host retrieves the final transfer and transaction receipt completion status codes. During a transfer cycle, the host may interrogate the current status at any time, and may request that the transfer be cancelled. (This does not mean, however, that a request can always be cancelled. Cancellation is always at the discretion of the gaming machine.)

To initiate a transfer cycle, the host issues long poll 72 with a transfer code of 00 or 01 indicating whether the gaming machine must transfer the full amount or if a partial transfer is allowed, and a transaction index of zero. The remaining fields must be filled in appropriately to describe the specific transfer being

requested. The transaction ID must be different from the transaction ID of the most recently completed transfer for a non-zero total amount (the last transfer placed in the history buffer).

If an initiating long poll 72 contains a transfer code that the gaming machine does not support, the gaming machine responds with a transfer status of C1, unsupported transfer code, and omits the remaining fields. The gaming machine should not attempt to parse the rest of the message.

If the gaming machine otherwise determines that the initiating long poll is not valid, or that it is unable to perform the requested transfer, or that it is unable to print a requested transaction receipt, it may respond immediately with a transfer status code from Table 8.3e indicating the reason the transfer was rejected. When determining which transfer failure code to report from Table 8.3e, the gaming machine is permitted to evaluate the error conditions in any order it chooses. For example, if a transfer to the gaming machine has an invalid transfer amount and the gaming machine is in a tilt condition, it may respond with either transfer status 83 or 87, depending on the order it evaluates the transfer poll. Because receipts are only printed for successful transfers for a non-zero total amount, the receipt status is set to FF when a transfer is rejected, fails or successfully completes for a total amount of zero. At this point, the transfer cycle is complete and no further processing is necessary.

If the gaming machine expects to be able to complete the transfer, or is unable to determine whether it is able to perform the transfer in time to respond to the initiating long poll 72 with the transfer completion status, it will respond to the initiating long poll with transfer status 40, transfer pending. If a transaction receipt has been requested, the receipt status code is also set to 40, receipt pending. If no receipt was requested, the receipt status code is set to FF. The gaming machine is then considered to be in a transfer cycle until the transfer and receipt (if any) are both complete or the transfer has been rejected, and the host has retrieved the final transfer and receipt completion status codes. It is important to understand that a transfer cannot be reported as complete until all affected meters have been updated, any requested receipt has been printed, and the transfer has been logged in history if appropriate.

Whenever the transfer status in the gaming machine response indicates the transfer is pending, each transfer amount should be the expected transfer amount. If partial transfer is allowed but the gaming machine has not yet determined the actual amounts that will be transferred, if any, it should report the full or maximum amounts while the transfer is pending, until it has determined the actual amounts, even though the actual transfer may be for partial amounts. When the gaming machine reports the transfer is complete, the amount fields must indicate the actual amounts transferred, or zero if the transfer failed.

The date and time fields should be all zeros while the transfer is pending. Once the transfer is complete, whether successful or failed, the date and time fields must indicate the time, according to the gaming machine clock, that the transfer completed.

The pool ID is ignored if the transfer does not include a restricted amount. Expiration is valid for all restricted amount transfers and all transfers from the host to ticket. Expiration is ignored otherwise. If the host uses long poll 72 to transfer an amount to a ticket, it may set the expiration to 00000000 to instruct the gaming machine to use its default expiration, or specify a valid expiration value that is not already expired. The gaming machine must not consider whether the expiration date is already expired in determining whether to accept a transfer of a restricted amount to the gaming machine.

When restricted amounts are transferred from the gaming machine to the host, the host should leave the pool ID and expiration fields in the initiating long poll 72 set to all zeros. The gaming machine's long poll 72 response will indicate the pool ID and expiration currently associated with any restricted amounts being transferred. The gaming machine must not consider whether the amounts are currently expired in determining whether to honor the request to transfer restricted amounts to the host. It is the responsibility of the host to check the current expiration using long poll 74 if it wants to limit transfer of expired restricted amounts.



At any time the host may use the interrogation form of long poll 72 to interrogate the current or most recent transfer and receipt status codes by setting the transfer code to FF and the transaction index to 00. All fields in the gaming machine response are set to the actual status of the current or most recent transfer. If the host requests the current transfer status and there has been no current or previous transfer, the response will have a transfer buffer position of 00, a transfer status of FF and a receipt status of FF, and the remaining fields are omitted.

Once the host has sent an initiating long poll 72 and the gaming machine has responded with a transfer pending status response 40, the host may not send another initiating long poll 72 until the current transfer cycle is complete and the host has received and acknowledged (see Section 3.1) the final transfer and receipt status codes using the interrogation long poll 72 with transfer code FF, as described above. The host may choose to poll for the transfer status at any time or wait for exception 69, AFT transfer complete. Remember that a response to an interrogation poll with transfer code FE does not count as reporting the transfer completion status.

At any time while a transfer is pending, the host is allowed to attempt to cancel the transfer by sending long poll 72 with a transfer code of 80 and omitting the remaining fields. If the gaming machine has not already irretrievably committed to performing the transfer it should abort the transfer request and respond with a transfer status of 80, transfer cancelled by host. If the gaming machine has irretrievably committed to the transfer, or has already completed the transfer, it may effectively ignore the cancel request and simply respond with the current pending or completion status codes. It is entirely up to the gaming machine to decide whether a pending transfer may be cancelled.

If the host sends any long poll 72 during a transfer cycle, other than an interrogation poll or request to cancel, that poll must have no effect on the current transfer cycle. If the long poll 72 contains a transfer code that the gaming machine does not support, the gaming machine will respond with a transfer status of C1, unsupported transfer code, and omit the remaining fields. Otherwise, the gaming machine will respond with a transfer status of C0, and omit the remaining fields.

While the transfer is pending, the cumulative amount meters are reported each with a size byte of zero. Once the transfer is complete, the cumulative amount meters report the total meters for the transfer type just completed (i.e. the same values as reported by the AFT-specific meters in Table C-7). These meters must include the values from the transfer just completed.

If the transfer has completed successfully and was logged in history, the transaction buffer position will indicate the position where the transaction has been stored in the transaction history.

If the host performs a transfer from the host to a ticket, that amount must be metered in any “total transfer in” meter as well as any “total ticket out” meter.

Transaction receipts are only printed for successful transfers for a non-zero amount. If a transfer fails for any reason, or successfully completes for an amount of zero, the transfer status will be set to the appropriate value from Table 8.3e and the receipt status will be set to FF if it is not already FF. If the transfer has successfully completed for a non-zero amount and a receipt was requested, the transfer status will be set to the appropriate completion code (00 or 01), and the receipt status will be set to 20 to indicate the receipt is being printed.

If the gaming machine responded to the initiating long poll 72 with transfer pending status 40, the gaming machine is responsible (to the best of its ability) to make sure the host retrieves the transfer and receipt completion status. When either the transfer status transitions from pending to complete (successful or not), or the receipt status transitions from printing to complete, and the host has not received and acknowledged the completion status, the gaming machine must issue exception 69, AFT transfer complete. The host will then issue the interrogation long poll 72 with the transaction index set to 00 to obtain the completion status. If the host does not respond with a proper interrogation poll, the gaming

machine must reissue exception 69 every 15 seconds until the host polls for and acknowledges the completion status.

Please note, exception 69 is not issued unless the gaming machine has responded to the initiating long poll 72 with transfer pending status of 40, and has not subsequently responded to an interrogation long poll 72 (transfer code FF) with both a transfer and receipt completion status. Exception 69 is not issued if the host polls for and acknowledges the completion status before the gaming machine can issue exception 69. Exception 69 is not issued if the gaming machine responded with a transfer status other than 40 to the initiating long poll 72.

Exception 69 is a priority exception, and is NOT inserted into the exception queue. It must be issued dynamically based on the current completion state. Exception 69 may be issued even though other exceptions are pending in the queue. It is the responsibility of the host to read the completion status. The gaming machine must not accept a new transfer until the current transfer cycle is complete and the host has polled for **and acknowledged** the final transfer and receipt completion status. The transfer cycle is not complete until both the transfer and receipt status flags have been read and acknowledged by the host with their final completion values.

## 8.4 Accepting Transfers

The gaming machine must reject all transfers or ignore transfer polls if it is not enabled for AFT. It must not accept transfers when it is in an unplayable state, such as door open, operator menu, tilt, disabled, waiting for handpay, etc., except if cash out is allowed to occur from a tilt or disable state. If the host has requested that a transfer only be accepted if the gaming machine is “locked,” the gaming machine must reject the transfer if it is not currently locked using long poll 74, and able to accept the requested transfer type in that lock state. Otherwise, the gaming machine will escrow transfers received during game play or at any other time while waiting for player input, and perform the transfer at its next available opportunity. In this case, the gaming machine will respond with transfer pending status 40.

If, before the gaming machine is able to perform the transfer, it transitions to a state where it would normally have rejected the transfer, such as a tilt occurring or a door opening, it should then reject the transfer and report the transfer complete. The transfer completion status will indicate the reason for rejection.

The gaming machine may perform an in-house or debit transfer from the host whenever it would normally allow money to be accepted or credits to be wagered. Understanding that gaming machines may enforce a transfer limit, for example due to a maximum allowable transfer amount or a credit meter limit, the gaming machine may reject the entire transfer when full transfers are required, or the portion that would exceed the limit if partial transfer is allowed. When multiple amounts are specified and less than the full total amount can be transferred, the gaming machine must transfer from the restricted amount first, if possible, then the nonrestricted amount, then the cashable amount, until the limit is reached. Transfers from the host to a ticket may have other restrictions, such as not allowed when the gaming machine is disabled, or when other rules would prevent the printer from being used as a cashout device.

The gaming machine may perform transfers from the gaming machine to the host at any time it would otherwise normally allow the player to cash out. One method for the host to request a transfer of all available credits to the host is to set all amounts to 999999999 and the transfer code to partial transfer allowed. A gaming machine is not required to allow cashouts from the credit meter of less than the full available amount for each type. If there are no credits on the gaming machine when the host requests a transfer of all available credits to the host, if possible the gaming machine should perform a successful transfer for a total amount of zero. This allows the host to use the cashout process to set the Host Cashout Enable state, and begin or end a cashless session, even if there are no credits currently on the gaming machine.

When the host requests a transfer from the gaming machine, it may also request that any amounts in the gaming machine credit meter greater than the amounts specified in the transfer request be cashed out by the gaming machine. This is accomplished by setting the transfer flag bit 3 to 1. One use for this feature is for the host to effectively press the cashout button on the gaming machine, by requesting a transfer type of transfer in-house amount from gaming machine to host, with all amounts set to zero and the transfer flag bit 3 set to 1. If possible, the gaming machine should perform a cashout by whatever means it normally would if the player had pressed the cashout button and cashout to host were not an option. Once the cashout has been performed, the transfer would be reported as completing successfully for a total amount of zero. Another example is for the host to cash out all restricted promotional credits to the host, while causing all cashable credits to be cashed out of the gaming machine, by setting the restricted amount to 999999999 and the cashable and nonrestricted amounts to zero.

## 8.5 Bonus Awards

Bonus award transfers differ from all other types of transfers in that they are considered to be game win, and contribute to the total gaming machine hold and yield calculations. Because bonus award transfers can be paid to the credit meter, or by hopper, ticket, handpay, etc., they are not limited by the gaming machine's credit limit or maximum transfer limit. Bonus award transfers must always be performed for the full requested amount, if at all. Bonus award transfers are accepted, and possibly escrowed, by the same rules as in-house and debit transfers above, and performed when the gaming machine is in a state where the player would normally be allowed to cash out. Bonus award transfers must never be accepted or performed when the gaming machine is in a disabled state, even if the player may cash out from this state.

The host may choose to transfer the bonus award as a "bonus coin out" or "bonus jackpot" type. The gaming machine may pay the bonus coin out award to the credit meter, or by hopper, ticket, handpay, etc., by the same rules it would use for a normal game win. If the award results in a jackpot handpay, the bonus award is metered in the Total Attendant Paid External Bonus Win meter and reported to the host in the long poll 72 transfer complete response as a bonus jackpot. Otherwise, the bonus award is metered in the Total Machine Paid External Bonus Win meter and reported to the host as a bonus coin out transfer.

The host may use the bonus jackpot transfer type to force a bonus win of any amount, including an amount of zero, to cause the game to lock up in a jackpot handpay state requiring attendant intervention. If the gaming machine implements a "W2-G Reset to Credit Meter" feature, the gaming machine's jackpot limit is not considered in determining whether the jackpot bonus win is eligible to be reset to the credit meter. If paid to the credit meter or other method besides attendant handpay, the bonus award is metered in the Total Machine Paid External Bonus Win meter and reported to the host as a bonus coin out transfer.

In addition to metering bonus awards in the Total Machine Paid External Bonus Win or Total Attendant Paid External Bonus Win meter, they are added to the Total Coin Out or Total Jackpot meter as appropriate. Total Coin Out and Total Jackpot are the meters reported by long polls 0F, 12, 14, 19, 1C, and 52, and Table C-7 meter codes 01 and 02 for long polls 2F, 6F and AF. New meters have been added for base payable win and progressive win, to allow for proper calculations of the base gaming machine hold and yield percentages. External bonus awards must **never** be added to these base meters.

Please refer to Section 13 for more details on bonusing, particularly enabling and disabling, reporting active players, and differences between legacy and AFT bonusing.

## 8.6 Transaction History

The gaming machine must maintain a circular buffer of the most recent successfully completed transfers for a non-zero total amount, and all successful bonus transfers, up to a maximum of 127. The gaming machine indicates how many buffer positions it supports in its long poll 74 response. Note that some

jurisdictions or other regulations may require a minimum number of transfers to be buffered in order to allow a gaming machine to perform AFT transfers. In lieu of other specific guidelines, a minimum of 70 positions is recommended.

Once a transfer for a non-zero total amount, or any bonus transfer, has completed successfully and the transaction receipt (if any) has been printed, the gaming machine will copy the transfer record to the next available location in the transaction history buffer. Buffer positions are numbered, starting with position 1. The first transaction copied to the history buffer goes in buffer position 1, and the buffer is filled sequentially until the last buffer position is filled. The next transaction then overwrites the transaction in buffer position 1, and so forth.

The host may use the interrogation form of long poll 72 to retrieve transactions from the history buffer using either an absolute buffer position number or a relative transaction index. Relative transaction index FF references the transaction most recently copied to the history buffer, index FE references the transaction copied prior to that, etc. Transaction indexes 01 through 7F reference absolute buffer positions. Once a transaction has been copied to the history buffer, it must remain at the same buffer position until overwritten by a newer transaction. The long poll 72 response must always indicate the absolute buffer position where the transaction data is stored when responding with transaction data that is stored in the buffer. This includes the most recently completed transaction, if it has successfully completed and been stored in the buffer. All data in the response, including the asset number, must be as it existed at the time the transfer completed.

Note that only a current unfinished transfer can have a transfer status of 40. The transfer status for completed transfers must always be the final completion status. If the transaction index refers to a transaction older than the oldest transaction currently buffered by the gaming machine, or a buffer position that is empty or greater than the maximum number of buffer positions on the gaming machine, the response will have a transfer status and receipt status of FF, and the remaining fields are omitted. The transaction buffer position in the response will be the requested absolute position or relative index.

## 8.7 Host Cashout Enable

When the host initiates a transfer using long poll 72, it may specify a requested Host Cashout Enable state. The gaming machine response to long poll 72 always indicates the current Host Cashout Enable state, unless the response data is from the history buffer. The current Host Cashout Enable state is set to the state requested by the host only when a transfer has successfully completed (transfer status 00 or 01).

The host may optionally perform a transfer for the sole purpose of setting the Host Cashout Enable state by setting the transfer amounts to zero. The gaming machine should accept a transfer request with a total amount of zero whenever possible, unless there is already currently a transfer cycle in progress. A successful transfer for a total amount of zero is not copied to the history buffer. Note that it is permissible for the gaming machine to override the host's requested Host Cashout Enable state, for example due to an operator configuration requiring that all cashouts go to the host.

When host cashouts are enabled, the gaming machine should treat the host as an available cashout device. Note, it is allowable that some cashouts may not be eligible to be cashed out to the host, for example if there is a maximum transfer limit. Whenever the gaming machine is requested to perform a cashout from the credit meter, such as when the player presses the cashout button, if host cashouts are enabled and the cashout is eligible to be cashed out to the host, the gaming machine will issue exception 6A, AFT request for host cashout. The host should then send a long poll 72 to initiate a transfer amount to host. If the host cashout mode is set to soft, gaming machines may choose to perform a cashout to a device other than the host, for example in response to a selection by the player.

The amount fields may all be set to 9999999999, with partial transfer allowed, to transfer the entire cashout amount. Optionally, all amounts may be set to zero to instruct the gaming machine to perform the cashout by whatever other means are available. The host may not perform any other type of transfer

while a cashout to host is pending, such as a transfer to the gaming machine or a transfer to a ticket. If requested to do so, the gaming machine will respond with status code 87, unable to perform transfers at this time. This must not cause the cashout in progress to be aborted.

If host cashouts are enabled, some systems or jurisdictions may require a gaming machine to operate by specific rules, such as establishing a limit over which all wins are cashed out to the host. Please contact your systems provider for details. Some wins may simply need to be cashed out rather than be paid to the credit meter, for example if the win amount would cause the gaming machine's credit meter limit to be exceeded. Please note that some wins may not be eligible to be cashed out to the host, for example due to a jackpot limit or maximum transfer limit.

When a win is ready to be cashed out to the host, the gaming machine issues exception 6B, AFT request for host to cash out win. The host will then send a long poll 72 with the transfer type set to 90, transfer win amount to host. Note that a transfer win amount to host is metered as an in-house transfer to the host. The amount fields may be set to 9999999999, with partial transfer allowed, to transfer the entire win amount. The amounts may be set to zero to instruct the gaming machine to either pay the win to the credit meter or cash out the win by whatever other means are available. If the host attempts a transfer for less than the full amount, the gaming machine may optionally perform a cashout to the host for the requested amount and cash out the remainder using whatever other means are available, or pay the entire win by a means other than to the host.

Exceptions 6A and 6B are priority exceptions, and are NOT inserted into the exception queue. They must be issued dynamically based on the current cashout pending state. Exceptions 6A and 6B may be issued even though other exceptions are pending in the queue. As long as a cashout to the host is pending, the gaming machine will reissue the exception 6A or exception 6B every 800 milliseconds. If the host sends long poll 74, AFT Game Lock And Status Request, the gaming machine resets its 800 millisecond timer. If the host fails to perform or deny the transfer within 8 seconds, the gaming machine must perform a cashout to host failure recovery process.

If the communications link is down at the time of the cashout, or is determined to be down during this period (see Section 4.3), the gaming machine should go ahead with its recovery process immediately, rather than waiting the full 8 seconds. If the host cashout mode is set to soft, the cashout to host failure recovery process is to go ahead and perform the cashout by whatever method would have been selected by the gaming machine if cashout to the host had not been an option to begin with.

If the host cashout mode is set to hard, a cashout to host failure should cause a "cashout to host failure" tilt. The gaming machine should then provide a mechanism for an attendant to select an alternate cashout method. It is preferred that the exception 6A or 6B continue to be issued, or be issued as soon as the host comes back on line, while waiting for an attendant action. If the host performs the cashout while the gaming machine is waiting for the attendant, the tilt should be cleared. Once the attendant has initiated any action, such as turning the attendant key, the host is not allowed to perform the cashout unless the attendant returns the gaming machine to the state where it is waiting for the host to perform the cashout.

## 8.8 Cash Out Button Pressed

If AFT transfers to host are enabled, the gaming machine will report exception 66 whenever the player presses the cash out button. This exception is reported regardless of the credit amount or type and will be reported even during game play and tilt conditions. If a gaming machine forces a cashout from the credit meter on behalf of the player, for example due to terminal disable, it should issue exception 66 the same as if the player had pressed the cashout button.

---

**Note:** *There is no need to delay 800 milliseconds when AFT Host Cashout is not enabled. Attempting to intercept cashouts from the host is inherently unreliable. The host must*

---

*use AFT Host Cashout Enable in order to be guaranteed an opportunity to process gaming machine cashouts.*

---

## **8.9 Lock After Transfer**

The AFT Lock After Transfer feature that was added to SAS 6.02 is flawed and cannot be implemented as defined. It is recommended that this feature not be used, and it has been deprecated in SAS 6.03. In the long poll 74 response, the Available Transfers bit 7 should be left as 0, and in the long poll 72 command, the Transfer Flags bit 4 should be ignored.

## **8.10 AFT Meters**

Gaming machines that support AFT must keep track of the cumulative value and the total number of transfers performed for each type of transfer supported. The host can obtain these meters by issuing type M long poll 2F, Send Selected Meters For Game N, or by issuing type M long poll 6F or AF, Send Extended Meters For Game N. The game number must always be 0000 for AFT meters, as AFT transfers are never tracked on a per game basis. AFT-specific meter code values have been added to Table C-7, starting with code A0.

## **8.11 Transaction Receipts**

In order to print transaction receipts and registration reports, a gaming machine must be equipped with a printer capable of at least 24 lines of 22 ASCII characters per line. Three basic receipt types are defined; in-house transfers to the gaming machine, debit transfers to the gaming machine, and in-house transfers to the host, including wins. Some lines are the same for all types of receipts, and some lines vary based on the type of transaction.

To provide reasonably consistent transaction receipts across gaming machines supplied by multiple manufacturers, a great deal of specific text is recommended by the protocol. It is understood that other considerations, such as foreign language support, may make it undesirable to follow these recommendations. It is highly encouraged that other manufacturers contact IGT for guidance whenever possible when deviating from these recommendations.

**8.11.1 LP 75: Set AFT Receipt Data**

Using the set AFT receipt data command, the host can configure a variety of data that may be printed on registration reports and transaction receipts. For ultimate flexibility, the host can select from a list of fields to configure. The number of fields that can be configured in one poll is limited only by the maximum length of the poll. This variable length type S command is detailed in Table 8.11.1a. The gaming machine ACKs or NACKs this message, as detailed in Table 7.4b on page 7-5. Long poll 75 may be issued to a specific gaming machine address, or as a variable length type G global broadcast by setting the address to 00. When long poll 75 is sent as a type G global broadcast, the gaming machine sets its values according to the poll but does not respond.

**Table 8.11.1a**  
**Set AFT Receipt Data Command**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Global broadcast or gaming machine address
Command	1 binary	75	Set AFT receipt data command
Length	1 binary	02-nn	Number of bytes following, not including the CRC
Data code	1 binary	nn	Code indicates data element type following (see Table 8.11.1b)
Data length	1 binary	nn	Length of data element following
Data	x bytes	???	Data element (see Table 8.11.1b)
...	variable	...	Additional data code/length/data elements
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 8.11.1b**  
**Transaction Receipt Data Fields**

Data code (binary)	Description	Data
00	Location	Variable ASCII text (22 max)
01	Address 1	Variable ASCII text (22 max)
02	Address 2	Variable ASCII text (22 max)
10	In-house line 1	Variable ASCII text (22 max)
11	In-house line 2	Variable ASCII text (22 max)
12	In-house line 3	Variable ASCII text (22 max)
13	In-house line 4	Variable ASCII text (22 max)
20	Debit line 1	Variable ASCII text (22 max)
21	Debit line 2	Variable ASCII text (22 max)
22	Debit line 3	Variable ASCII text (22 max)
23	Debit line 4	Variable ASCII text (22 max)

Variable ASCII text data consists of a length byte followed by up to max ASCII bytes. Specifying a data code followed by a length byte of zero will cause the field to revert to any default value. To print a blank line for a specific field, set the ASCII text to one or more ASCII blanks (hex 20).

### 8.11.2 Transaction Receipt Layout

In the interest of providing consistent transaction receipts from all machines connected to a system, the following guidelines are recommended for receipt layout. Jurisdictional, language or other considerations may require receipts to be formatted differently from these guidelines. Please consult your systems provider for details.

Following is documentation of each line of the receipt, and the source of the text. Any line for which data has not been provided should be left blank. If a receipt is requested and data has not been provided for a required line, the transfer must be rejected.

Line 1: Location

Source: Operator entry or long poll 75 data

Line 2: Address1

Source: Operator entry or long poll 75 data

Line 3: Address2

Source: Operator entry or long poll 75 data

Line 4: Blank

Line 5: Transfer description

Source: Long poll 72 transfer type (see Table 8.11.2)

Line 6: Transfer source/destination

Source: Long poll 72 print data (ASCII text as received, or blank)

Line 7: Blank

Line 8: Date and time

Source: Long poll 72 print data, or date and time transfer completed if not specified by host

Line 9: Blank

Line 10: Asset number

Source: Set in gaming machine

Line 11: Blank (in-house) or POS ID (debit)

Source: Debit = POS ID from long poll 73

Line 12: Patron name (in-house) or blank (debit)

Source: In-house = long poll 72 print data (ASCII text as received, or blank)

Line 13: Patron acct# (in-house) or Debit card# (debit)

Source: In-house = "Acct: " followed by long poll 72 print data

Debit = "Acct: xxxxxxxxxxxx" followed by long poll 72 print data

Line 14: Blank



Line 15: Transaction ID

Source: Long poll 72 transaction ID

Line 16: Total cashable transfer amount

Source: Descriptive text based on transfer type (see Table 8.11.2), followed by total of cashable and nonrestricted transfer amounts from long poll 72 response (leave line blank if total cashable amount is zero)

Line 17: Restricted transfer amount (in-house) or blank (debit)

Source: In-house = descriptive text based on transfer type (see Table 8.11.2), followed by restricted transfer amount from long poll 72 response (leave line blank if restricted amount is zero)

Line 18: Blank (in-house) or transaction fee (debit)

Source: Debit = "Transaction Fee" followed by long poll 72 print data, or blank

Line 19: Account balance (in-house) or total debit (debit)

Source: In-house = "Acct Bal" followed by sum or difference of long poll 72 print data and total transfer amount, or blank

Debit = "Total Debit" followed by long poll 72 print data, or calculated total (debit transfer amount plus fee) if total is not provided but transaction fee is provided, or blank

Line 20: Blank

Line 21: In-house text 1 (in-house) or debit text 1 (debit)

Source: Long poll 75 data (ASCII text as received, or blank)

Line 22: In-house text 2 (in-house) or debit text 2 (debit)

Source: Long poll 75 data (ASCII text as received, or blank)

Line 23: In-house text 3 (in-house) or debit text 3 (debit)

Source: Long poll 75 data (ASCII text as received, or blank)

Line 24: In-house text 4 (in-house) or debit text 4 (debit)

Source: Long poll 75 data (ASCII text as received, or blank)

**Table 8.11.2**  
**Transfer Descriptive Text**

<b>Transfer type</b> (binary)	<b>Transfer description</b> (line 5)	<b>Total cashable amount label</b> (line 16)	<b>Restricted amount label</b> (line 17)
00	TRANSFER TO GAME	Cash In	Promo In
20	TRANSFER TO GAME	Cash Ticket	Promo Ticket
40	DEBIT CARD WITHDRAWAL	Debit In	
60	DEBIT CARD WITHDRAWAL	Debit Ticket	
80	TRANSFER FROM GAME	Cash Out	Promo Out
90	TRANSFER FROM GAME	Cash Out	

### 8.11.3 Sample Transaction Receipts

Following are some example transaction receipts.

AFT Transfer To Gaming Machine

```

1  LUCKY LARRY'S CASINO
2  777 ADRIAN WAY
3  RENO, NV 89511
4
5  TRANSFER TO GAME
6
7
8  10/15/2001  11:16:32
9
10 EGM 123456
11
12 Freddy Reels spinner
13 Acct: 615902814
14
15 12345678901234567890
16 Cash In      $180.00
17 Promo In     $20.00
18
19
20
21 In-house text line 1
22 In-house text line 2
23 In-house text line 3
24 In-house text line 4

```

## AFT Transfer From Gaming Machine

```
1  LUCKY LARRY'S CASINO
2    777 ADRIAN WAY
3    RENO, NV 89511
4
5    TRANSFER FROM GAME
6
7
8    10/15/2001  12:34:56
9
10   EGM 341256
11
12   Johnny W. Jackpot
13   Acct: 777-12345-6789
14
15   23456789012345678901
16   Cash Out      $1234.50
17
18
19   Acct Bal      $54321.25
20
21   In-house text line 1
22   In-house text line 2
23   In-house text line 3
24   In-house text line 4
```

## Debit Card Withdrawal

```
1  LUCKY LARRY'S CASINO
2    777 ADRIAN WAY
3    RENO, NV 89511
4
5    DEBIT CARD WITHDRAWAL
6    FROM PRIMARY ACCOUNT
7
8    10/15/2001  14:13:12
9
10   EGM 456123
11   POS 2105439876
12
13   Acct: xxxxxxxxxxxx1248
14
15   13579246809753186420
16   Debit Ticket   $500.00
17
18   Transaction Fee $1.75
19   Total Debit     $501.75
20
21   Debit text line 1
22   Debit text line 2
23   Debit text line 3
24   Debit text line 4
```

## Debit Registration Report

```

1  LUCKY LARRY'S CASINO
2  777 ADRIAN WAY
3  RENO, NV 89511
4
5  REGISTRATION REPORT
6
7
8  05/15/2001  08:09:10
9
10 EGM 456123
11 POS 2105439876
12
13
14
15
16
17
18
19
20
21 Debit text line 1
22 Debit text line 2
23 Debit text line 3
24 Debit text line 4

```

**8.12 LP 76: Set Custom AFT Ticket Data**

The type S long poll 76, Set Custom AFT Ticket Data, provides support for custom text and graphics on tickets generated using the AFT transfer to ticket functionality. The host may set custom text and graphics for AFT tickets, while leaving standard text in place for tickets generated due to normal cashout activity.

The host may use long poll 76 to specify custom data elements for tickets that are printed as a result of an AFT transfer to ticket. The variable length long poll 76 command is detailed in Table 8.12a.

**Table 8.12a**  
**Set Custom AFT Ticket Data Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	76	Set custom AFT ticket data command
Length	1 binary	01-nn	Number of bytes following, not including the CRC
Function	1 binary	nn	00 = set data elements (or interrogate only, if no data elements specified) 80 = clear all data parameters (omit remaining fields)
Data code	1 binary	nn	Code indicates data element type following (see Table 8.12c)
Data length	1 binary	nn	Length of data element following
Data	x bytes	???	Data element (see Table 8.12c)
...	variable	...	Additional data code/length/data elements
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine responds with a list of currently configured elements as detailed in Table 8.12b.

**Table 8.12b**  
**Set Custom AFT Ticket Data Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	76	Set custom AFT ticket data
Length	1 binary	00-nn	Number of bytes following, not including the CRC
Data codes	n binary	??	Codes for each custom data element currently configured, if any (see Table 8.12c)
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 8.12c**  
**Ticket Data Elements**

Data code (binary)	Description	Data
00	Custom AFT location	Variable ASCII text (40 max)
01	Custom AFT address 1	Variable ASCII text (40 max)
02	Custom AFT address 2	Variable ASCII text (40 max)
03	Custom AFT graphics selector	Variable ASCII text (3 max)
10	Custom AFT ticket title	Variable ASCII text (16 max)

A gaming machine must maintain the data sent using long poll 76 in non-volatile memory. Variable ASCII text data consists of a length byte followed by the corresponding number of ASCII bytes. Specifying a data code followed by a length byte of zero will cause the selected element to be cancelled, or unset. To set a blank line for a specific element, set the ASCII text to one or more ASCII blanks (hex 20). All custom elements may be cancelled by setting the function code to 80 and omitting the remaining fields. If a gaming machine receives a function code it does not support, it will ignore any data for that function code.

The gaming machine response will indicate which elements, if any, are currently set. The response includes a list of all codes from Table 8.12c for the elements that currently have data assigned to them. If no elements are currently set, the response will have a length of zero.

When the host requests an AFT transfer to ticket using long poll 72, it indicates that custom ticket data should be used on the ticket by setting the Transfer Flags bit 5 to one. If bit 5 is set to zero, the standard or default ticket data will be used. If bit 5 is set to one, the following rules must be observed when printing the AFT ticket.

For each data element that can be set using long poll 76, if a custom parameter has been set, that parameter is used. If no custom parameter has been set, the corresponding standard parameter is used. If no standard parameter has been set, the gaming machine's default or operator-entered parameter will be used.

The custom graphics selector is passed to the printer in the format provided for by the communications interface between the gaming machine and the printer. The specific effect of any graphics selector is

entirely dependent on the printer firmware. One anticipated use is to instruct the printer to print custom graphics on a ticket, such as a birthday cake and balloons. However, because the interpretation of the graphics selector is left to the printer firmware, neither the SAS protocol nor the gaming machine need be concerned with the actual meanings of these parameters.

The custom ticket title is printed on the tickets where text such as “CASHOUT TICKET” is printed on normal cashout tickets. Please use caution in setting this string, that the promotional or cashable nature of the specific ticket is clearly indicated.

A gaming machine will indicate that it supports custom ticket data in its long poll 74 response by setting AFT Status bit 2 to one.

## **SECTION 9 RESERVED**

This section intentionally left blank.

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH





## SECTION 10 PROGRESSIVES

SAS progressive support allows the SAS host to provide progressive amounts to the gaming machine. The gaming machine must be configured with a non-zero Group ID to enable progressive control by the SAS host. SAS also supports reporting of limited progressive data for non-SAS progressives (link, standalone, WAP, a different SAS host, etc.). Any progressive wins on a gaming machine not administered by the SAS host which is polling for the data are considered non-SAS progressives for that host, even if they are administered by a different SAS host.

### 10.1 LP 80, 86: Progressive Broadcasts

Using the global broadcast format defined in Section 2, the host can send progressive information to the gaming machines. For gaming machines that are configured for a small number of progressive levels, the host can issue the progressive broadcast detailed in Table 10.1a. To accommodate gaming machines that are configured for many progressive levels, the host can issue the variable length progressive broadcast detailed in Table 10.1b and send up to 32 progressive levels to each group. However, gaming machines are not required to support 32 progressive levels. Also, some platforms may have limits on the maximum number of bytes for any one SAS message. If the length of this message exceeds the number of bytes that a gaming machine can receive, that gaming machine ignores this message. Gaming machines do not respond to global broadcasts. Long polls 80 and 86 can also be sent to any single gaming machine as a type S poll. When received as a type S poll, the gaming machine ACKs or NACKs the message, as detailed in Table 7.4b on page 7-5.

**Table 10.1a**  
**Single Level Progressive Broadcast Format**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Global broadcast or gaming machine address
Command	1 binary	80	Progressive broadcast
Group	1 binary	01-FF	Group ID for this broadcast
Level	1 binary	01-20	Progressive level
Amount	5 BCD	0000000000 - 9999999999	Level amount in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 10.1b**  
**Multiple Level Progressive Broadcast Format**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Global broadcast or gaming machine address
Command	1 binary	86	Multiple progressive broadcast
Length	1 binary	07-C1	Length of data to follow, not including the message CRC
Group	1 binary	01-FF	Group ID for this broadcast
Level	1 binary	01-20	Progressive level
Amount	5 BCD	0000000000 - 9999999999	Level amount in units of cents
...	Variable	...	Optional additional level/amount pairs

---

CRC	2 binary	0000-FFFF	16-bit CRC
-----	----------	-----------	------------

---

### 10.1.1 Group

This field identifies the group to which the level and amount of the broadcast belong. By grouping progressive levels together, a single host can act as the progressive controller for multiple, mutually exclusive sets of progressive gaming machines. Group ID 00 is reserved for non-SAS progressives.

### 10.1.2 Level

The level field allows multiple progressive amounts to be configured under a single group. Level 01 represents the top progressive award for the group, level 02 is the next highest progressive award, etc.

### 10.1.3 Amount

This is the amount of the progressive level in units of cents.

## 10.2 Timing

Progressive broadcasts are issued by the host as needed to update the gaming machines. However, a gaming machine configured for SAS progressives must receive updates to its configured levels (the configured levels for the currently selected game on a multi-game gaming machine) in a timely manner. The gaming machine must receive a progressive broadcast for each configured level within five seconds from the last time a broadcast for that level was received. In order to more easily meet this timing requirement, the multiple progressive broadcast may be used to send all active progressive levels in one message.

If a gaming machine does not receive a progressive broadcast within the required time frame, it reports exception 53 (no progressive information has been received for five seconds). Note that this exception is only reported once when communication is lost, not every five seconds while not receiving progressive data. It is the responsibility of the gaming machine manufacturer at the time of implementation to determine the gaming machine action after reporting exception 53. It is the responsibility of the host to broadcast the progressive levels in such a way that the gaming machine can obtain the current progressive amounts in a timely manner.

## 10.3 Contributions

There are several ways for the host to obtain progressive coin in contribution amounts. When the gaming machine is operating in the real time event reporting mode, the credits wagered amount from the game start message can be used. The host can also request the gaming machine's coin in meter and calculate a delta amount. For a gaming machine with a configured max bet of 10 or less, the coin/credit wagered exception can be used.

## 10.4 LP 84: Send Progressive Win Amount

When a progressive win occurs on a gaming machine, the gaming machine reports exception code 54 for a cashout device/credit paid win or 51 for handpay pending. Upon receiving exception code 51, the host will normally issue the send handpay information long poll. The gaming machine response to the send handpay information long poll is detailed in Section 7 on page 7-13.

If a pending handpay includes progressive amounts, and some or all of those amounts are actually machine paid, for example due to a W2-G Reset to Credit Meter key-off, both exceptions 51 and 54 must be issued.

When the gaming machine issues exception 51 and/or 54, the most recent progressive win information is available through the send progressive win amount long poll. For the gaming machine response to the send progressive win amount long poll, see Table 10.4 below.

**Table 10.4**  
**Send Progressive Win Amount Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	84	Send progressive win amount
Group	1 binary	00-FF	Group ID of the progressive
Level	1 binary	01-20	Progressive level
Amount	5 BCD	0000000000-9999999999	Win amount in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

Long poll 84, Send Progressive Win Amount, normally reports the group, level and amount for the last progressive win on a gaming machine. However, it is possible to hit more than one progressive level in one game cycle, and it is not possible to report a progressive win until the pay method is determined at the end of the game. In order to prevent the loss of progressive information, it is essential that long poll 84 report the TOTAL progressive win amount for that game cycle. In the case of multiple progressive wins, the group and level reporting should follow the same rules laid out for long poll 1B.

In the case where a game cycle includes amounts from more than one progressive win, the progressive win amount reported by long poll 84 will be the total progressive amount for that game cycle. The progressive group and level fields are filled in based on the highest contributing progressive hit.

If a progressive win results in a handpay lockup that is subsequently reset to the credit meter, exception 51 will be issued to indicate a handpay is pending, and exception 54 must be issued when the handpay lockup is reset to indicate the progressive win was machine paid.

## 10.5 SAS-Controlled Progressives

To support SAS-controlled progressives, the gaming machine must maintain an n-entry first in/first out queue of SAS progressive win data. This queue must be deep enough to hold the maximum number of progressive levels that can be hit in any one game cycle. When a SAS progressive level is hit, the level and amount are placed in this queue and exception 56 (SAS progressive level hit) is reported. This exception is reported in addition to any exception 51 (handpay is pending) or exception 54 (progressive jackpot cashout device/credit paid). This exception is not reported for non SAS progressives.

Exception 56 is a priority exception. While records remain in the progressive win queue, the gaming machine reissues exception 56 every fifteen seconds. Two methods are supported for retrieving data from the progressive win queue.

**Note:** "SAS-controlled progressives" refers to progressive levels where this SAS host is providing the progressive level amounts. If a gaming machine supports more than one SAS host, progressive levels controlled by a different host are not considered SAS-controlled from the perspective of this host.

### 10.5.1 LP 85: Send SAS Progressive Win Amount

Upon receiving exception 56, the host may request the progressive win amount one record at a time by sending a type R long poll with an 85 command code. The gaming machine response is detailed below in Table 10.5.1.

**Table 10.5.1**  
**Send SAS Progressive Win Amount Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	85	Send SAS progressive win amount
Group	1 binary	01-FF	SAS group number
Level	1 binary	01-20	Progressive level
Amount	5 BCD	0000000000-9999999999	Win amount in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

When the gaming machine responds to long poll 85 with data from the queue and the response is acknowledged, the record is deleted from the queue. If additional records remain in the queue, exception 56 is reissued and the process repeats. If no records are in the queue when the gaming machine receives long poll 85, it will respond with the group, level, and amount fields set to zero.

### 10.5.2 LP 87: Send Multiple SAS Progressive Win Amounts

Upon receiving exception 56, the host may alternatively request all progressive win amounts in the queue by sending a type R long poll with an 87 command code. The variable length gaming machine response is detailed below in Table 10.5.2.

**Table 10.5.2**  
**Send Multiple SAS Progressive Win Amounts Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	87	Send multiple SAS progressive win amounts
Length	1 binary	02-C2	Number of bytes following, not including CRC
Group	1 binary	01-FF	SAS group number
Number of levels	1 binary	00-20	Number of levels following (00 if queue empty)
Level	1 binary	01-20	Progressive level of first entry
Amount	5 BCD	0000000000-9999999999	Win amount of first level in units of cents
...	variable	...	Additional level/amount data sets
CRC	2 binary	0000-FFFF	16-bit CRC

The response to long poll 87 includes all data in the progressive win queue, up to 32 records. When the response is acknowledged, the reported records are deleted from the queue. Note that a maximum of 32 records can be reported at one time. If game design issues require a queue larger than 32 elements, and additional records remain in the queue, exception 56 is reissued and the process repeats. If no records are in the queue when the gaming machine receives long poll 87, it will respond with a length of 02 and the SAS group number. The number of levels field will be set to zero, and no level/amount data will be included.

A gaming machine indicates that it supports long poll 87 by setting Features3 bit 1 to one in its long poll A0 response.

**Note:** Exception 54 and long poll 84 do not adequately support SAS progressives on all platforms. The host MUST issue long poll 85 or 87 in response to exception 56 but should still issue long poll 1B in response to exception 51.

## 10.6 Resetting Progressive Levels

Once the host has received the progressive win information for a SAS progressive win, it should immediately broadcast the reset amount for the hit progressive. This allows gaming machines in that progressive group to update their amounts and displays in a timely manner.

## 10.7 LP 83: Send Cumulative Progressive Wins

Each time a gaming machine awards a progressive win, either by cashout device/credit pay or handpay, it converts the progressive win amount to credits and adds them to the Cumulative Progressive Wins meter. For multi-game gaming machines, this may be done on a per game level as well as a gaming machine level. The host can obtain this information by issuing a type M long poll with command code 83. The command, detailed in Table 10.7a, specifies the game number of the desired game.

**Table 10.7a**  
**Send Cumulative Progressive Wins Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	83	Send cumulative progressive wins
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to this long poll is detailed in Table 10.7b.

**Table 10.7b**  
**Send Cumulative Progressive Wins Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	83	Send cumulative progressive wins
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Cumulative progressive wins	4 BCD	00000000-99999999	4-byte BCD meter in SAS accounting denom units
CRC	2 binary	0000-FFFF	16-bit CRC



## SECTION 11 TOURNAMENT

### 11.1 LP 8C: Enter/Exit Tournament Mode

Tournament mode configuration allows the host to remotely configure gaming machines that support one or more tournament mode(s). This includes configuring the time and/or credits along with tournament pulses functionality.

The host configuration message, detailed in Table 11.1, specifies the game to enable/disable tournament mode on, time in minutes and seconds of the tournament game, starting credits for the tournament game, and whether or not tournament pulses are enabled or disabled. Gaming machines that do not support SAS-controlled tournament mode will ignore this long poll.

**Table 11.1**  
**Enter/Exit Tournament Mode Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	8C	Enter/exit tournament mode
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Time	2 BCD	0000-9959	MSB = minutes for the tournament time LSB = seconds for the tournament time
Credits	4 BCD	00000000-99999999	Starting credit amount for the tournament session
Pulses	1 binary	00-01	00 - Tournament pulses disabled 01 - Tournament pulses enabled
CRC	2 binary	0000-FFFF	16-bit CRC

To configure a gaming machine for a 'time only' tournament session, this message is sent with a zero amount in the credits field. Likewise, for 'credit only' tournament session, this message is sent with a zero in the time field. The host can terminate tournament mode by issuing this message with zero amounts in the time and credit field.

### 11.2 Entering Tournament Mode

When a gaming machine receives the enter/exit tournament long poll, it will first complete any game, funds transfer, or bill transaction prior to entering tournament mode. If the gaming machine is in a tilt or handpay condition, it will wait until the condition is reset to enter tournament mode. If tournament mode has been configured with a time limit, the timer should not start until the start of the first tournament game.

### 11.3 Tournament Accounting

Gaming machines that support SAS-controlled tournament mode must account for the number of tournament games played, games won, credits wagered, and credits awarded per tournament session. For multi-game gaming machines, this accounting should be done for every tournament capable game. To obtain this information, the host can issue a type M long poll with the command code 95, 96, 97, 98, and 99 (see Appendix B for details).





## SECTION 12 REAL TIME EVENT REPORTING

For situations where real time event reporting is desired, the gaming machine can be configured to report events in response to long polls as well as general polls. This allows events such as reel stops, coins in, game end, etc., to be reported in a timely manner. Gaming machines must default to the polling and response structure detailed in Sections 2 and 3 on initial power up and when recovering from a power down condition.

### 12.1 LP 0E: Enable/Disable Real Time Event Reporting

To configure a gaming machine for real time event reporting or to disable real time event reporting on a gaming machine, the host issues the type S long poll detailed in Table 12.1. The gaming machine ACKs or NACKs this message as detailed in Table 7.4b on page 7-5.

Table 12.1 Enable/Disable Real Time Event Reporting			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	0E	Enable/Disable real time event reporting
Enable/disable	1 binary	00-01	00 - Disable 01 - Enable
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.2 Polling Method

The polling format defined in Section 2 is used by the host to obtain meter information. However, the polling rate can be increased to 40 ms in order to better approximate real time reporting.

### 12.3 Priority

Event reporting takes priority over long poll responses. If a gaming machine has any outstanding events to report when it receives a long poll, it reports the event.

### 12.4 Host/Gaming Machine Acknowledgment

In the event that the host receives an invalid event response to a long poll, it NACKs the message by reissuing the original long poll. The gaming machine must not change its response type when it receives a NACK. If the gaming machine responded with a normal long poll response to the first poll and then an event occurs between the first and second poll, the gaming machine must respond to the second poll with the normal long poll response.

### 12.5 Event Response Format

When configured for real time event reporting, gaming machines no longer report exceptions as single byte codes, even when responding to a general poll. All exceptions are reported using the event message detailed in Table 12.5. Some exceptions, detailed in Sections 12.5.1 through 12.5.8, contain additional data. The gaming machine only sends this data when it is configured for real time event reporting.

**Table 12.5**  
**Real Time Event Reporting Message Format**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Exception code	1 binary	00-FF	Exception code (see Appendix B)
Data	X varies	???...	Any additional data
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.1 Exc 4F: Bill Accepted

This message, detailed in Table 12.5.1, includes the country code, denomination code, and the number of accepted bills of this type.

**Table 12.5.1**  
**Bill Accepted Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Bill accepted	1 binary	4F	Bill accepted exception code
Data	6 BCD	00-99	Country code (See Table C-5 in Appendix C)
		00-99	Denomination code (See Table C-6 in Appendix C)
		00000000	Number of accepted bills of this type
		-	
		99999999	
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.2 Exc 7C: Legacy Bonus Pay Was Awarded

When a system initiated legacy bonus or multiplied jackpot is awarded by the gaming machine, it reports the multiplier and multiplied win amount, if any, and the tax status and bonus amount, if any. The message format is detailed in Table 12.5.2.

**Table 12.5.2**  
**Legacy bonus Pay Was Awarded Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Legacy bonus pay	1 binary	7C	Legacy bonus pay was awarded exception
Multiplier	1 binary	00-0A or 81-8A	1 byte binary multiplier (bit 7: 1 = non-deductible, 0 = deductible) 00 = no multiplied win
Multiplied win	4 BCD	00000000- 99999999	Multiplied win amount, not including the original win, in SAS accounting denom units
Tax status	1 binary	00-02	Tax status of the legacy bonus award 00 – Deductible or no award 01 - Non-deductible 02 - Wager match
Bonus	4 BCD	00000000- 99999999	Legacy bonus award amount (from long poll 8A) in SAS accounting denom units
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.3 Exc 7E: Game Start

When a game is initiated, the gaming machine sends the game start message detailed in Table 12.5.3. Included with this message is the number of wagered credits for the current game, coin in meter, wager type, and progressive group for the current game.

**Table 12.5.3**  
**Game Start Event Message**

Field	Bytes	Value	Description	
Address	1 binary	01-7F	Gaming machine address	
Event identifier	1 binary	FF	Real time event message identifier	
Game start	1 binary	7E	Game start	
Credits wagered	2 BCD	0000-9999	Credits wagered for the current game, in units of game denomination	
Total coin in meter	4 BCD	00000000-99999999	Total gaming machine coin in meter after credits wagered, in SAS accounting denom units	
Wager type	1 binary	00-FF	Bit	Description
			5~0	Denomination of game played, from Table C-4, or 0 if not multi-denom
			6	0 = Not multi-denom 1 = Multi-denom machine
			7	0 = Max bet not wagered 1 = Max bet wagered
Progressive group	1 binary	00-FF	Progressive group for this game (only if this game is SAS progressive)	
CRC	2 binary	0000-FFFF	16-bit CRC	

Note that the credits wagered field is in units of actual game credits wagered, independent of any denomination. The Total Coin In meter, and all of the meters in other RTE responses, are in units of the SAS accounting denomination.

#### 12.5.4 Exc 7F: Game End

After the final game outcome evaluation, the gaming machine reports the game end event detailed in Table 12.5.4. Included with this event is any game win amount, not including bonus awards.

**Table 12.5.4**  
**Game End Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Game end	1 binary	7F	Game end
Game win	4 BCD	00000000-99999999	Game win, in SAS accounting denom units. Does not include SAS bonus awards.
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.5 Exc 88: Reel N Has Stopped

The reel N has stopped message, detailed in Table 12.5.5, includes the reel number and physical stop. This event is sent only if real time event reporting is enabled.

**Table 12.5.5**  
**Reel N Has Stopped Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Reel n stopped	1 binary	88	Reel n has stopped exception
Reel number	1 binary	01-09	Reel number of stopped reel
Physical stop	1 binary	00-FF	Physical stop
CRC	2 binary	0000-FFFF	16-bit CRC

If the game has more than 9 reels, only the first 9 reels can be reported. In the event that the gaming machine has multiple win lines, the stops positions reported must correspond to the center or first line. If a single credit wager applies to more than one line, then a “center” line should be defined and documented for that gaming machine.

If a reel has more than 255 stops, any stop greater than 255 must be reported as 255. Games with more than 9 reels and/or more than 255 stops per reel may optionally not report reel stops.

### 12.5.6 Exc: 8A: Game Recall Entry Displayed

When an attendant views a game recall entry on a gaming machine, this event message, detailed in Table 12.5.6, is sent. Specified in this message is the multi-game game number of the recalled game and the recall entry index, with 0000 being the most recently played game on the gaming machine, 0001 the next most recent, etc.

For a game cycle with multiple play steps, exception 8A is only sent when first entering the game recall entry. The exception is not sent and the recall index is not modified when moving between play steps for a single game cycle.

**Table 12.5.6**  
**Game Recall Entered Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Game recall entry displayed	1 binary	8A	Game recall entry displayed exception
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Recall index	2 BCD	0000-9999	Recall entry index for the game
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.7 Exc 8B: Card Held/Not Held

Table 12.5.7 details the card held/not held message. This message indicates the card number and whether it was held or not held. On multi-hand card games only the first or base hand can be reported. This event is sent only if real time event reporting is enabled.

<b>Table 12.5.7 Card Held/Not Held Event Message</b>			
<b>Field</b>	<b>Bytes</b>	<b>Value</b>	<b>Description</b>
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Card held/not held	1 binary	8B	Card held/not held exception
Card	1 binary	00-04 or 80-84	Card number and status Left most card = 0, right most card = 4 Bit 7: 0 = not held, 1 = held
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.8 Exc 8C: Game Selected

On a multi-game gaming machine, whenever the player selects a new game the gaming machine reports a game selected exception. Table 12.5.8 below details its format.

<b>Table 12.5.8 Game Selected Event Message</b>			
<b>Field</b>	<b>Bytes</b>	<b>Value</b>	<b>Description</b>
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Game selected	1 binary	8C	Game selected exception
Game number	2 BCD	0000-9999	Selected game number (0000 = in game menu)
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.9 Exc 9E: Session Start

When a session is initiated, the gaming machine sends the variable length session start message detailed in Table 12.5.9. Included with this message are the game number, denomination, total session cost, and number of games in the session (see Section 18.2).

**Table 12.5.9**  
**Session Start Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Session start	1 binary	9E	Session start
Length	1 binary	09-nn	Number of bytes following, not including the CRC
Game number	2 BCD	0000-9999	Game number (0000 if not multigame)
Denomination	1 binary	01-3F	Binary number representing the player denomination of the game (see Table C-4 in Appendix C)
Session wager	4 BCD	00000000-99999999	Total cost for the session, in SAS accounting denom units
Games in session	2 BCD	0000-9999	Total number of games in the session
CRC	2 binary	0000-FFFF	16-bit CRC

### 12.5.10      **Exc 9F: Session End**

After the final game of the session has been played and reported, or if a session is ended early, the gaming machine reports the variable length session end event detailed in Table 12.5.10. Included with this event are the actual number of games played in the session, total sessions played meter for the game, total win amount for this session, and any residual amount added to the Coin In meter if a session is ended early.

**Table 12.5.10**  
**Session End Event Message**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Event identifier	1 binary	FF	Real time event message identifier
Session end	1 binary	9F	Session end
Length	1 binary	0E-nn	Number of bytes following, not including the CRC
Games played	2 BCD	0000-9999	Actual games played in the session
Sessions played	4 BCD	00000000-99999999	Total number of sessions played on this game (including this session)
Session win	4 BCD	00000000-99999999	Total session win in SAS accounting denom units
Extra coin in	4 BCD	00000000-99999999	If a session is terminated early, indicates any additional amount added to the Coin In meter
CRC	2 binary	0000-FFFF	16-bit CRC

## 12.6 No Activity Exceptions

When configured for real time event mode operation, gaming machines do not report exception codes 00 (no activity) and 1F (no activity and waiting for player input) in response to a long poll. No activity

on the gaming machine is implied when the gaming machine does not send a real time event in response to a long poll.

## **12.7 ROM Signature Response**

When real time event reporting is enabled the gaming machine must respond with a ROM signature response in response to a long poll if a ROM signature response is pending. However, unlike real time event reporting, ROM signature responses do not include the event identifier byte 0xFF. This distinguishes a ROM signature response from a coin in tilt exception response.



## SECTION 13 BONUSING

With the introduction of the Advanced Funds Transfer (AFT) protocol, SAS supports two different forms of bonusing. Each bonus process has its own procedures and meters. For the purpose of differentiation, the direct bonus award and multiplied jackpot features described in this chapter are referred to as “legacy” bonusing. All SAS bonus awards, regardless of source, are metered in the Total Machine Paid External Bonus meter or Total Attendant Paid External Bonus meter, as appropriate. However, only AFT bonus awards are metered in the AFT-specific meters, and only legacy bonus awards, including multiplied jackpots, are metered in the specific meters described in this section.

### 13.1 Enabling/Disabling Bonusing

Gaming machines that support any form of SAS bonusing, including AFT bonusing, must have a secure method for enabling and disabling it at a gaming machine level. AFT bonusing and legacy bonusing should each be able to be configured separately. When a gaming machine is configured with legacy bonusing disabled, it ignores all legacy bonusing commands, including long poll 8A, Initiate Legacy Bonus, long poll 8B, Initiate Multiplied Jackpot Mode, and long poll 2E, Game Delay.

In the event of a SAS communications failure, the gaming machine disables multiplied jackpots. Once communication with the host is reestablished, the host must assume that multiplied jackpots are disabled and can, if desired, enable multiplied jackpots on the gaming machine.

If a gaming machine has received and ACKed one or more legacy bonus award long polls and is waiting until the completion of the current game to award them when a communications failure with the host is detected, it still awards the pending bonuses and places the 7C exception in its exception queue. Exception 7C is never reported as a result of an AFT bonus award.

### 13.2 Reporting Active Players

A SAS host can be configured to award ‘active’ players with additional bonuses. An active player is defined as a person initiating and completing a game within a specified time period. The host determines the active player status by starting a timer when it receives a game start exception. If a second game start exception is received before the timer expires, that player is deemed active.

Gaming machine conditions where it is not waiting for user input are not considered in the determination of active players. These include, but are not limited to, hopper pays, handpays, tilts, door open, etc. In order to distinguish between gaming machine conditions where it is waiting for user input and conditions where it is not waiting for user input, exception code 1F has been added. Exception 1F is reported only if the gaming machine has AFT bonusing or legacy bonusing enabled, is not in tournament mode, and is waiting for the player to act before continuing. This includes being in the idle state, waiting for a player to insert coins, play credits, press start, hold cards, enter/exit double up, etc. Conversely, exception 00 is used to indicate that the gaming machine is not waiting for the player to act in order to continue. Situations such as self test, display meters, evaluations, handpay conditions, paying coins from the hopper, tilts, etc., all result in exception code 00 being reported.

---

**Note:** *Exception 1F has not been implemented consistently, and has not proven to be useful in the field. It is recommended that gaming machines do not implement exception 1F, and that systems treat this exception as equivalent to exception 00, no activity.*

---

### 13.3 LP 8A: Initiate Legacy Bonus Pay

The host can instruct a gaming machine to award a bonus to a player. This is accomplished by sending the type S long poll detailed in Table 13.3 and specifying the credit amount and tax status of the bonus. The gaming machine ACKs or NACKs this message as detailed in Table 7.4b on page 7-5.

**Table 13.3**  
**Initiate Legacy Bonus Pay Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	8A	Initiate a legacy bonus pay
Bonus amount	4 BCD	00000000-99999999	Bonus amount in SAS accounting denom units
Tax status	1 binary	00-02	00 – Deductible 01 - Non-deductible 02 - Wager match
CRC	2 binary	0000-FFFF	16-bit CRC

#### 13.3.1 During Game Play

When a gaming machine receives a legacy bonus during game play, it holds that bonus in escrow until the end of that game. If additional legacy bonuses are received before the end of the game, they are added to the current bonus escrow amount and held until the end of that game.

On the completion of the game, the gaming machine reports the game end exception, delays if needed (see 13.7), then awards the escrowed bonus amount to the player.

In the event that the communications link between the gaming machine and host is lost when the gaming machine has an escrow bonus amount, the gaming machine will award the escrow bonus amount as detailed in the preceding paragraph.

#### 13.3.2 During Idle

Any legacy bonus received by the gaming machine while it is in the idle state is paid immediately. If the gaming machine is processing an event such as a cash out request or bill insertion, it will finish processing the event before awarding the bonus.

#### 13.3.3 During a Handpay

When a gaming machine receives a legacy bonus while a handpay is already pending, it escrows the bonus until such time as the handpay has been reset. If additional bonuses are received before the handpay is reset, they are added to the escrow.

#### 13.3.4 During Player Screens

Certain video gaming machines possess additional player screens such as ‘help’, ‘paytable’, ‘menu’, etc., that the player can select. Any legacy bonus received by the gaming machine while it is displaying a player screen must be acknowledged and can either be awarded immediately or escrowed until such time as the player screen is exited.

### 13.3.5 During a Malfunction, Door Open, or Maintenance

In the event that a legacy bonus is received by the gaming machine when it is in a tilt condition, door opened, maintenance mode, or game recall mode, it should not be escrowed. The gaming machine indicates its inability to fulfill the bonus award by issuing the game busy response (see Section **Error! Reference source not found.** on page 4-1).

## 13.4 LP 8B: Multiplied Jackpot Mode (obsolete)

**Note:** Any implementation of the Multiplied Jackpot feature is very dependent on game design issues, making it impossible to achieve a consistent implementation. It is strongly recommended that this feature not be implemented or used. The following documentation is maintained only for backwards compatibility.

Through multiplied jackpots, the host is able to instruct the gaming machine to multiply all wins within a specified range by a specified value. Detailed in Table 13.4, this type S host message consists of the minimum and maximum win, multiplier/tax status, enable/disable flag, and wager type. The gaming machine ACKs or NACKs this message as detailed in Table 7.4b on page 7-5.

**Table 13.4**  
**Multiplied Jackpot Mode Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	8B	Multiplied jackpot mode
Minimum win	4 BCD	00000000-99999999	Minimum win, inclusive, that is eligible for a multiplied jackpot
Maximum win	4 BCD	00000000-99999999	Maximum win, inclusive, that is eligible for a multiplied jackpot
Multiplier/tax status	1 binary	01-0A/ 81-8A	Multiplier Bit 7: 0 = deductible, 1 = non-deductible
Enable/disable	1 binary	00-01	00 – Enable 01 – Disable
Wager type	1 binary	00-01	00 – All wagers are eligible 01 – Only max bet wagers are eligible
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** Any multiplied jackpot message received during game play won't take effect until the completion of the current game.

### 13.4.1 Multiplied Jackpots and Multi-Line Gaming Machines

Multi-line gaming machines are gaming machines configured with multiple winning lines, each of which may have its own independent wager. For example, the gaming machine could be a “9 line” game with up to 5 coins/credits wagered on any line for a total maximum bet of 45 coins/credits.

When a multiplied jackpot message is received with the wager type field indicating that only max bet wagers are eligible for the multiplier, this implies that the total max bet, i.e., 45 coins/credits for the above example, must be bet in order for that game cycle to be eligible. In the event that a game cycle on a multi-line gaming machine results with wins on multiple lines, the individual wins are added together and the multiplied jackpot is applied to the sum.

### 13.4.2 Multiplied Jackpots and Bonus Awards

External bonus awards, from any source, are not considered in the evaluation of the base game win limits or included in the multiplication of multiplied jackpots. Therefore, escrowed bonus awards, i.e., bonus awards received during game play, are not added to the base gaming machine win and are not eligible to be multiplied.

### 13.4.3 Multiplied Jackpots and Progressive Wins

Progressive wins are considered part of the base gaming machine win and as such are eligible for multiplied jackpots. It is the responsibility of the system to either not configure progressive gaming machines for multiplied jackpots or to set the minimum and maximum wins for the multiplied jackpot accordingly if potentially large multiplied win are to be avoided.

Wide Area Progressives (WAP) wins are never eligible for a multiplied jackpot, even if the gaming machine is configured for it. This provides an additional level of security against an extremely large win amount being multiplied.

## 13.5 LP 90: Send Legacy Bonus Win Amount

When the gaming machine awards a legacy bonus, it reports exception code 7C. In response, the host can issue a type R long poll and request the bonus award information. The gaming machine response is detailed in Table 13.5 below.

**Table 13.5**  
**Send Legacy Bonus Win Amount Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	90	Send legacy bonus win amount
Multiplier	1 binary	00-0A/ 81-8A	1 byte binary multiplier (bit 7: 1 = non-deductible, 0 = deductible)
Multiplied win	4 BCD	00000000-99999999	Multiplied win amount not including the original win, in SAS accounting denom units
Tax status	1 binary	00-02	Tax status of the legacy bonus 00 – Deductible 01 – Non-deductible 02 – Wager match
Bonus	4 BCD	00000000-99999999	Legacy bonus win amount in SAS accounting denom units
CRC	2 binary	0000-FFFF	16-bit CRC

It is the responsibility of the host to obtain this information in a timely manner as the gaming machine only reports the most recent bonus/multiplied jackpot award. Once this award information has been sent to the host and has been acknowledged, it is cleared from the gaming machine's memory. If the host again requests this information, the gaming machine will respond with zero amounts.

### 13.6 LP 9A: Send Legacy Bonus Meters

Gaming machines must account for all deductible, non-deductible, and wager match legacy bonus awards and all multiplied jackpots. For multi-game gaming machines, this accounting may be done on a per game level as well as on a gaming machine level. By issuing a type M long poll with a 9A command code, the host can request the legacy bonus meters from the gaming machine. The command, detailed in Table 13.6a, specifies the game number of the desired game.

**Table 13.6a**  
**Send Legacy Bonus Meters Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	9A	Send legacy bonus meters
Game number	2 BCD	0000-9999	Game number (0000=gaming machine)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine will respond as detailed in Table 13.6b.

**Table 13.6b**  
**Send Legacy Bonus Meters Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	9A	Send legacy bonus meters
Game number	2 BCD	0000-9999	Game number (0000 = gaming machine)
Deductible	4 BCD	00000000-99999999	Deductible bonus meter in SAS accounting denom units
Non-deductible	4 BCD	00000000-99999999	Non-deductible bonus meter in SAS accounting denom units
Wager Match	4 BCD	00000000-99999999	Wager match bonus meter in SAS accounting denom units
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** Meters reported using long poll 9A must include only amounts awarded using SAS legacy bonus polls. They do not include amounts awarded using any other process or protocol.

### 13.7 LP 2E: Game Delay

It is possible for a gaming machine to have such a fast game cycle that the host, after receiving the game start exception, cannot issue a bonus pay before the end of the game. To remedy this, the game delay command, detailed in Table 13.7, is available.

**Table 13.7**  
**Game Delay Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	2E	Game Delay
Buffer amount	2 binary	0000-FFFF	Delay time in units of 100ms
CRC	2 binary	0000-FFFF	16-bit CRC

Gaming machines configured for game delay will, after determining the final game outcome and sending the game end exception, start the delay. During the game delay, any initiate legacy bonus commands received by the gaming machine will be added to the bonus escrow and awarded after the delay.

Any new game delay command received by the gaming machine while it is currently performing a game delay will replace the remaining current delay time and take affect immediately. This allows the host to extend the delay time or to cancel it by configuring the gaming machine for a delay time of zero.

Once configured for a game delay, the gaming machine will delay for all future games.

## SECTION 14 JACKPOT HANDPAY RESET METHODS

In many gaming jurisdictions there is a win threshold where all wins from a single game that exceed the threshold must be reported. Gaming machines typically have a configurable ‘handpay’, or, ‘jackpot’ limit that allows an attendant to configure this limit as needed based on a particular gaming jurisdiction’s win threshold.

For high denomination gaming machines, this win threshold often results in wins of only several credits being hand paid. For example, a gaming machine with a \$100.00 denomination in a gaming jurisdiction where the win threshold is \$1,200.00 would require handpays for all wins of 12 or more credits.

To reduce the number of attendant payouts on these high denomination gaming machines, an alternate method of resetting handpays has been developed. Under this new method, the gaming machine still enters a handpay condition for wins that exceed the win threshold and an attendant is still required to reset the gaming machine. However, before the gaming machine jackpot handpay condition is reset, the player can opt to have it reset onto the gaming machine credit meter rather than be paid by the attendant.

To reset a jackpot handpay to the gaming machine credit meter, the attendant first must obtain authorization before filling out a credit receipt for the player. Once authorization has been obtained and the required paperwork has been completed, the handpay condition is reset. If the jackpot handpay cannot be reset to the credit meter, the attendant will not be given authorization and therefore must proceed with the standard jackpot handpay procedures.

### 14.1 Attendant Authorization

When a jackpot handpay condition occurs on a gaming machine, it checks the win against the upper jackpot limit. The upper jackpot limit provides an upper limit for jackpot handpays to be eligible to be reset to the credit meter. Any win that is greater than or equal to this limit is not eligible to be reset to the credit meter.

If the win is not greater than or equal to the upper jackpot limit, the gaming machine then determines whether or not it can add the win to the current credit amount without exceeding the gaming machine credit limit. Any single win that when added to the current credit amount would exceed the gaming machine credit limit is not eligible to be reset to the credit meter.

Jackpot handpays that are determined to be eligible for a reset to the credit meter are reported to the host by loading the Reset ID field with a 01 in the gaming machine response to the 1B long poll (see page 7-13). Ineligible jackpot handpays and handpays that have already been reset are reported to the host by loading the Reset ID field with 00.

When the host receives the handpay jackpot pending exception from the gaming machine, it issues the 1B long poll to obtain the handpay information. When the attendant on the gaming floor requests authorization to reset the handpay to the gaming machine credit meter, the host attendant may use the reported Reset ID to determine whether or not to enable a jackpot handpay reset method and authorize the handpay request.

### 14.2 LP 8A: Enable Jackpot Handpay Reset Method

Before the reset jackpot handpay to the credit meter request can be authorized by the host attendant, the appropriate jackpot handpay reset method must be enabled on the gaming machine. This is accomplished through the use of the type S long poll detailed in Table 14.2a. If SAS is not enabled to control the jackpot handpay reset method, long poll A8 will be ignored.

**Table 14.2a**

Enable Jackpot Handpay Reset Method Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	A8	Enable jackpot handpay reset method
Reset method	1 binary	00-01	00 – Standard handpay 01 – Reset to the credit meter
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response, detailed in Table 14.2b, contains a single byte acknowledgment code. If the host attempts to enable a reset method on a gaming machine when it is not in a handpay condition, it responds with acknowledgment code 02. If the gaming machine is in a handpay condition but cannot comply with the host request, it responds with acknowledgment code 01. If the gaming machine is able to comply with the host request, it responds with acknowledgment code 00.

Table 14.2b Enable Jackpot Handpay Reset Method Response			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	A8	Enable jackpot handpay reset method
ACK code	1 binary	00-02	00 – Reset method enabled 01 – Unable to enable reset method 02 – Not currently in a handpay condition
CRC	2 binary	0000-FFFF	16-bit CRC

The enable jackpot handpay reset method long poll affects only the pending handpay. Once that handpay is reset, the gaming machine reverts back to its standard handpay reset method. Additionally, only a single jackpot reset handpay method can be active on a gaming machine at any given time. A gaming machine that has been configured to reset a jackpot handpay to its credit meter cannot perform a standard reset for that jackpot handpay unless another enable jackpot handpay reset method is received enabling the standard handpay reset method.

**Note:** *Jackpot handpay resets to the credit meter are accounted for by the gaming machine as machine pay, not attendant pay.*



## SECTION 15 VALIDATION AND TICKET REDEMPTION

SAS provides support for three types of cashout validation; standard, secure enhanced and system. Only one method of validation can be supported at any time. Selection between validation modes may be provided by an operator setup option, most likely protected by a setchip or other secure access method.

Standard validation provides for a gaming machine generated eight-digit (4 BCD) validation number. An example standard validation number algorithm is described in Section 15.14. Standard validation lacks sufficient security to allow automatic redemption of a cashout ticket at a gaming machine. To address the security and accountability requirements of modern Ticket In/Ticket Out systems, secure enhanced validation and system validation methods have been defined. Secure enhanced validation places many of the security requirements on the gaming machine to allow more autonomous operation and support of handpay validation, whereas system validation places most of the responsibility for security on the host. Because they share many polls and processes, secure enhanced validation and system validation are referred to collectively as enhanced validation.

Secure enhanced validation provides for a gaming machine generated 16-digit (8 BCD) validation number. The secure enhanced validation number algorithm is described in Section 15.12. To create this validation number, the gaming machine needs to maintain a gaming machine validation ID number and a validation sequence number in non-volatile memory. The validation system ID for secure enhanced validation is always 00. Secure enhanced validation requires the gaming machine to disable itself and not allow game play until the gaming machine validation ID and starting sequence number have been configured by the host, unless handpay validation has been disabled. A gaming machine enabled for secure enhanced validation that does not have a valid gaming machine validation ID and sequence number must report exception 3F (validation ID not configured) at power-up, and every fifteen seconds until configured.

System validation allows the host to provide a 16-digit (8 BCD) validation number plus a 2-digit (1 BCD) non-zero validation system ID for cashout tickets at the time of the cashout. With system validation enabled, a gaming machine will issue exception 57, system validation request, when a cashout requiring validation is pending, and wait up to ten seconds for the host to provide a validation number. The gaming machine reissues exception 57 every 800 milliseconds while waiting, until it receives a long poll 57 or 58.

When a gaming machine prints a cash out ticket or a handpay or jackpot receipt, it reports exception 3D (a cash out ticket has been printed). Secure enhanced validation also allows that the gaming machine validate handpays where no receipt is printed. If a validated handpay does not result in a receipt being printed, such as with hopper-only machines, the gaming machine will report exception 3E (handpay has been validated) after the handpay has been reset. A gaming machine must never report both exception 3D and exception 3E for the same handpay event. Note that for all intents and purposes, exceptions 3D and 3E are functionally equivalent. The host should use the data from long poll 4D to determine the validation type.

It is important to understand the difference between a cash out ticket and a handpay receipt. A cash out ticket is printed and delivered directly to a player. For the purposes of metering, the ticket is the cash out. A handpay receipt is printed when an attendant resets a jackpot or cancelled credit handpay. The receipt is not the cash out, the handpay is the cash out, and is metered the same regardless of whether a receipt is printed or not. The option of whether or not a gaming machine prints a handpay receipt, and how a receipt is used, is up to the operator and/or the jurisdiction. System validation does not support jackpot or handpay receipts, or handpay validation, since a handpay must be allowed to occur whether the system is able to validate it or not.

If a sequential ticket number is printed on a ticket or receipt, do NOT use the secure enhanced validation sequence number provided by the host. The gaming machine must maintain its own sequence number for this purpose.

For a gaming machine to support secure enhanced validation, it must maintain a circular buffer of ticket/receipt/handpay validation records for at least five and not more than 31 cash out tickets, handpay receipts and/or handpays. Buffer positions are numbered, starting with one, to enable the host to re-acquire previously read validation records. Initially, the first record is put in buffer position number one, and the buffer is then filled sequentially. When the buffer is full, each new record overwrites the oldest record.

The gaming machine must disable itself before the buffer becomes full of records that have not been read by the host in order to prevent loss of validation information. If it is possible for the player to cash out while the gaming machine is disabled, the disable must occur while there is still room in the buffer for all final cash out records.

When operating in secure enhanced validation mode, if the link is down (see Section 4.3) or *any* unread validation records remain in the validation buffer for more than 10 seconds, the gaming machine may not use the printer to print cashout tickets for the player. In this way only jackpot and handpay events, with or without receipts, will need to be validated.

System validation utilizes the same buffer mechanism as secure enhanced validation. However, because jackpot and handpay events do not require validation, and cashout tickets will not be printed when unread validation records remain in the buffer, it is reasonable to expect there would never be more than one unread record in the buffer.

For the validation controller, exceptions 3D and 3E are priority exceptions, and must not be inserted in the exception queue. The gaming machine must reissue the 3D or 3E exception to the validation controller every fifteen seconds as long as unread records remain in the validation buffer. If the gaming machine is also communicating to a SAS host that is not the validation controller, exceptions 3D and 3E are treated as normal exceptions, and are inserted into the exception queue for that host once for each associated event. Validation records are not buffered for the non-validating host. Only the most recent validation amount is available.

SAS also provides a method to redeem tickets that have been printed by a gaming machine that supports secure enhanced or system validation. Ticket redemption is **not** supported in conjunction with standard validation. ***Gaming machines that support secure enhanced or system validation must also support a 40 ms polling rate.*** Due to the time-critical nature of ticket redemption, the host is not required to wait 40 ms to respond to a priority exception, provided that poll is the next poll following the priority exception response.

Extended validation support provides enhancements to better support restricted and nonrestricted promotional tickets under secure enhanced and system validation. Extended validation support includes improved ticket expiration support and pool IDs for restricted promotional credits. Extended validation status is available using long poll 7B, and extended ticket data may be set using long poll 7C.

Long poll A0, Send Enabled Features, can be used to determine if a gaming machine is operating in standard, secure enhanced or system validation mode, if it supports ticket redemption, and if it supports validation extensions.

## 15.1 Improved Ticket Expiration Support

Extended validation support provides improved functionality for setting and reporting ticket expiration values. Using long poll 7B, the host can set the expiration to be used for cashable tickets and handpay receipts to “n” days using a 2 byte BCD field, allowing expiration values of greater than 255 days. A separate default expiration for restricted tickets may also be set to “n” days using a separate 2 byte BCD field in long poll 7B.

When redeeming a restricted ticket, the host may override the default restricted ticket expiration by providing an expiration to use for the specific restricted amounts as part of the redemption poll. Whenever the credit meter has no restricted amounts, the gaming machine reverts to the default expiration. The specific expiration is set using a 4 byte BCD field appended to the long poll 71 command. The expiration may be set to “n” days or to a specific date. The field will either be MMDDYYYY or 0000NNNN days. It can be set to 00000000, or omitted, to specify the default expiration.

The expiration date is not evaluated when the ticket is redeemed. It is only used by the gaming machine when processing a cashout request for any remaining restricted credits. When set to a specific date, the restricted amounts may not be cashed out on a restricted ticket if the current date is later than the expiration date. The host may easily disable cash out of specific restricted amounts by setting an expiration date prior to the current date (01011901 for example). The gaming machine is not responsible for checking for legal dates. It is the host’s responsibility to never set an expiration date such as 02312002. The important distinction between the two formats is that a specific date sets the expiration relative to when the amounts are transferred. An “n” days expiration is always relative to when the ticket is printed. Expiration values are always used as sent. The gaming machine must never alter them based on the passage of time.

After printing any ticket, the gaming machine will tell the host what expiration was printed on the ticket in the long poll 4D response, as described in Section 15.3 below.

## 15.2 LP 7B: Extended Validation Status

Long poll 7B, Extended Validation Status, allows the host to control several gaming machine parameters associated with validation and ticket printing. The host may also use this long poll to inquire the current status of these parameters. The variable length type S long poll 7B command is detailed in Table 15.2a. Long poll 7B may be issued to a specific gaming machine address, or as a variable length type G global broadcast by setting the address to 00.

**Table 15.2a**  
**Extended Validation Status Command**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Global broadcast or gaming machine address
Command	1 binary	7B	Extended validation status
Length	1 binary	08	Number of bytes following, not including CRC
Control mask	2 binary	0000-FFFF	Set bit to 1 to allow control of corresponding function in control bits (See Table 15.2c)
Status bit control states	2 binary	0000-FFFF	Bit = 1 to enable function, 0 to disable function, if corresponding mask bit = 1 (See Table 15.2c)
Cashable ticket and receipt expiration	2 BCD	0000-9999	Number of days before cashable tickets and handpay receipts expire (0000 = do not change, 9999 = never expire)
Restricted ticket default expiration	2 BCD	0000-9999	Default number of days before restricted tickets expire (0000 = do not change, 9999 = never expire)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to the type S long poll 7B is detailed in Table 15.2b. When long poll 7B is sent as a type G global broadcast, the gaming machine sets its values according to the poll but does not respond.

**Table 15.2b**  
**Extended Validation Status Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	7B	Extended validation status
Length	1 binary	0A	Number of bytes following, not including CRC
Asset number	4 binary	nnnnnnnn	Gaming machine asset number or house ID
Status bits	2 binary	0000-FFFF	Bit = 1 if function currently enabled, 0 if function currently disabled (see Table 15.2c)
Cashable ticket and receipt expiration	2 BCD	0001-9999	Number of days before cashable tickets and handpay receipts expire (9999 = never expire)
Restricted ticket default expiration	2 BCD	0001-9999	Default number of days before restricted tickets expire (9999 = never expire)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response includes the current operator-entered gaming machine asset number (if any), the current status states (after processing the host's request to set any states), the expiration to be used for printing cashable tickets and handpay receipts, and the default expiration for restricted tickets.

The asset number is an operator-entered value used to uniquely identify the gaming machine. If no asset number has been assigned, the asset number field is set to zero. It is recommended that if an operator has entered a non-zero asset number, that the asset number be printed on a cashout ticket in the location where the host ID would normally be printed. Please consult your systems provider for specific details.

Each expiration value is set as "n" days. A value of 0000 specifies that the current expiration is to be left in place. The host can set a value to 9999 to say "never expires." While the protocol allows up to 9998 days, it is expected that the system will impose a more "practical" limit. The gaming machine response always indicates what the expiration values are currently set to. Values set using long poll 7B take precedence over the expiration field in long poll 7D. Once expiration values have been set using long poll 7B, the expiration field in long poll 7D must be ignored, unless valid enrollment is cancelled.

**Table 15.2c**  
**Validation Control/Status Bits**

Byte	Bit	Description	Control	Status
LSB	0	Use printer as cashout device	0 = Do not allow 1 = Allow	0 = Currently not available 1 = Currently available
	1	Use printer as handpay receipt device	0 = Do not allow 1 = Allow	0 = Currently not available 1 = Currently available
	2	Validate handpays and handpay receipts	0 = Do not allow 1 = Allow	0 = Currently not configured 1 = Currently configured
	3	Print restricted tickets	0 = Do not allow 1 = Allow	0 = Currently not allowed 1 = Currently allowed
	4	Tickets for foreign restricted amounts	0 = Do not allow 1 = Allow	0 = Currently not allowed 1 = Currently allowed
	5	Ticket redemption	0 = Do not allow 1 = Allow	0 = Currently not allowed 1 = Currently allowed
	7~6	Reserved	Set to 0	Returns 0
MSB	6~0	Reserved	Set to 0	Returns 0
	7	Secure enhanced validation configuration	0 = Cancel configuration 1 = No change (use long poll 4C to configure)	0 = Configuration not set 1 = Currently configured (using long poll 4C)

LSB bit 0 allows the host to specify whether the printer may be used as a cashout device. This affects all tickets that would be printed directly for the player without attendant intervention, including all cashable and restricted tickets. For backwards compatibility, this option should initially default to “allow.” There are other reasons the printer may not be available as a cashout device, such as gaming machine not configured with a printer or other operator option to disable the printer as a cashout device, printer malfunction, validation ID not configured, unread validation records in the buffer, etc. The gaming machine response always indicates whether a printer is currently available as a cashout device at the time of the response.

LSB bit 1 allows the host to specify whether the gaming machine prints a handpay receipt following a handpay. For backwards compatibility, this option should initially default to “allow.” There are other reasons the printer may not be available as a receipt device, such as gaming machine not configured with a printer or other operator option to disable the printer as a receipt device, printer malfunction, validation ID not configured, etc. The status response indicates whether the gaming machine will currently print handpay receipts, taking into account the current setting of this control, the current status of the printer, etc. Note that handpay receipts are not supported in the system validation mode.

LSB bit 2 allows the host to specify whether the gaming machine validates handpays. For backwards compatibility, this option should initially default to “allow.” There are other reasons handpays may not be validated, such as an operator option to disable handpay validation or system validation mode enabled. If handpays are never validated, no receipt can ever be printed following a handpay. Therefore, if bit 2 is set to zero, the host setting for bit 1 is ignored and handpay receipts are not printed. If the gaming machine does not validate handpays, it does not need to disable itself in secure enhanced validation mode when it does not have valid configuration data from a long poll 4C. It simply does not use the printer as a cashout device. The gaming machine must still issue exception 3F in secure enhanced validation mode when it does not have valid configuration data. The status response indicates whether the gaming machine is currently configured to validate handpays.

LSB bit 3 allows the host to specify whether the gaming machine is allowed to print a restricted cashout ticket for restricted amounts. For backwards compatibility, this option should initially default to “allow.” There are other reasons why the printer may not be used to cash out restricted amounts, such as an operator option to disable this capability. The status response indicates whether the gaming machine is currently allowed to print restricted tickets.

LSB bit 4 allows the host to specify whether the gaming machine is allowed to print a restricted cashout ticket for restricted amounts from “foreign” sources, that is, from any source other than ticket in. For backwards compatibility, this option should initially default to “allow.” If the gaming machine is not allowed to cash out foreign restricted amounts, those amounts must not be combined with restricted amounts that may be cashed out. The status response indicates whether the gaming machine is currently allowed to print restricted tickets from foreign sources.

LSB bit 5 allows the host to specify whether a gaming machine configured for secure enhanced or system validation is allowed to perform ticket redemption. For backwards compatibility, this option should initially default to “allow.” There are other reasons why ticket redemption may not be allowed, such as an operator option to disable this capability. If the gaming machine is not allowed to perform ticket redemption, it will reject all tickets without issuing any exception 67. The status response indicates whether the gaming machine is currently enabled for ticket redemption.

MSB bit 7 allows the host to cancel any validation ID previously sent using long poll 4C, Set Secure Enhanced Validation ID. In addition to the host being able to cancel validation configuration, the gaming machine should cancel validation configuration if any operator configurations are changed that affect communications between the validating host and the gaming machine. A gaming machine is also permitted to provide an operator option specifically to allow the operator to cancel validation configuration.

Whenever validation configuration is cancelled for any reason, the gaming machine should revert to its defaults or operator configurations for all data previously sent using long polls 7B, 7C and 7D. In secure enhanced validation mode, the gaming machine must issue exception 3F whenever the validation configuration has been cancelled. Even though it can not use the values for creating validation numbers, the gaming machine must, to the best of its ability, still respond with the most recent gaming machine validation ID and sequence number if the host sends long poll 4C with a gaming machine validation ID of 0000.

All other bits are currently reserved. The host should never attempt to control reserved bits, unless they have been defined in a future revision to this protocol. A gaming machine should ignore attempts to control undefined reserved bits, and respond with a status of zero.

Please note that it’s possible that changes requested while the gaming machine is currently performing a cashout may not take effect until after the current cashout is completed.

### 15.3 LP 7C: Set Extended Ticket Data

Using the set extended ticket data command, the host can configure a variety of data that may be printed on cashout tickets and handpay receipts. For ultimate flexibility, the host can select from a list of fields to configure. The number of fields that can be configured in one poll is limited only by the maximum length of the poll. This long poll can be issued to a single gaming machine as a type S poll by using a non-zero polling address. A host can optionally broadcast this data to all gaming machines on a loop as a type G poll by setting the polling address to zero. Long poll 7C can also be used to interrogate whether a gaming machine has received a previous long poll 7C with valid data, by setting the length byte to 00 and omitting all data fields. This variable length poll is detailed in Table 15.3a.

**Table 15.3a**

### Set Extended Ticket Data Command

Field	Bytes	Value	Description
Address	1 binary	00-7F	Gaming machine address
Command	1 binary	7C	Set extended ticket data command
Length	1 binary	00-nn	Number of bytes following, not including the CRC
Data code	1 binary	nn	Code indicates data element type following (see Table 15.3c)
Data length	1 binary	nn	Length of data element following
Data	x bytes	???	Data element (see Table 15.3c)
...	variable	...	Additional data code/length/data elements
CRC	2 binary	0000-FFFF	16-bit CRC

As a type S poll, the gaming machine responds as detailed in Table 15.3b. Gaming machines do *not* respond to type G polls.

**Table 15.3b**  
**Set Extended Ticket Data Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	7C	Set extended ticket data
Ticket data status flag	1 binary	00-01	00 = Flag currently false 01 = Flag currently true
CRC	2 binary	0000-FFFF	16-bit CRC

A gaming machine must maintain the data sent using long poll 7C in non-volatile memory. It must also maintain a ticket data status flag in non-volatile memory. When a long poll 7C with valid data is received, this flag is set to true (before the response, if any). When a long poll 7C with invalid data is received, this flag is set to false (before the response, if any). Whenever a gaming machine is able to respond to long poll 7C (i.e. not sent as a Type G broadcast message), the current state of this flag is returned, and the flag is reset to false when the implied ACK from the host is received. Note that when long poll 7C is sent as a type S poll with data, the response is essentially an ACK/NACK flag indicating whether valid data was received.

**Table 15.3c**  
**Ticket Data Fields**

Data code (binary)	Description	Data
00	Location	Variable ASCII text (40 max)
01	Address 1	Variable ASCII text (40 max)
02	Address 2	Variable ASCII text (40 max)
10	Restricted ticket title	Variable ASCII text (16 max)
20	Debit ticket title	Variable ASCII text (16 max)

Data set using this long poll always takes precedence over any data set using long poll 7D.

Variable ASCII text data consists of a length byte followed by the corresponding number of ASCII bytes. Specifying a data code followed by a length byte of zero will cause the field to revert to any default value. To set a blank line for a specific field, set the ASCII text to one or more ASCII blanks (hex 20).

Note that the host is allowed to set a title for restricted tickets and a title for debit tickets. These are the ASCII strings printed on the tickets where text such as “CASHOUT TICKET” is printed on normal cashout tickets. The preferred default string for restricted tickets is “PLAYABLE ONLY.” The preferred default string for debit tickets is “DEBIT TICKET.”

#### 15.4 LP 7D: Set Ticket Data (legacy)

Several data fields on a ticket or handpay receipt are likely to be the same for all gaming machines connected to a particular host. Long poll 7D allows the host to send this data to multiple gaming machines, relieving an attendant from the task of entering this text manually at each individual gaming machine. This long poll can be issued to a single gaming machine as a type S poll by using a non-zero polling address. A host can broadcast this data to all gaming machines on a loop as a type G poll by setting the polling address to zero. Long poll 7D can also be used to interrogate whether a gaming machine has received a previous long poll 7D with valid data, by setting the length byte to 00 and omitting all data fields. This variable length poll is detailed in Table 15.4a.

**Table 15.4a**  
**Set Ticket Data Command**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Address of gaming machine
Command	1 binary	7D	Set ticket data
Length	1 binary	00, 02-7E	Number of bytes following, not including the CRC
Host ID	2 binary	0000-FFFF	Host identification number
Expiration	1 binary	00-FF	Number of days before ticket expires (00 = never expires)
Location length	1 binary	00-28	Length of location name data (00 = do not change)
Location data	x bytes	???	Location ASCII text data (0 to 40 bytes)
Address 1 length	1 binary	00-28	Length of address 1 data (street addr) (00 = do not change)
Address 1 data	x bytes	???	Address 1 ASCII text data (0 to 40 bytes)
Address 2 length	1 binary	00-28	Length of address 2 data (city/state/zip) (00 = do not change)
Address 2 data	x bytes	???	Address 2 ASCII text data (0 to 40 bytes)
CRC	2 binary	0000-FFFF	16-bit CRC

To send data to one or more gaming machines, the minimum length is 02, which means at least the Host ID data must be provided. All other fields are optional, except that to send Address 1 data, for example, the Expiration, Location Length and Location Data fields would need to be included. Note that any text data field may be omitted by setting the associated length field to 00. A gaming machine must maintain this data in non-volatile memory. It must also maintain a ticket data status flag in non-volatile memory. When a long poll 7D with valid data is received, this flag is set to true (before response, if any). When a long poll 7D with invalid data is received, this flag is set to false (before response, if any). Whenever



a gaming machine is able to respond to long poll 7D (i.e. not sent as a Type G broadcast message), the current state of this flag is returned, then the flag is reset to false.

The gaming machine response to long poll 7D, when sent as a Type S poll, is detailed in Table 15.4b. Gaming machines do **not** respond to type G polls.

Table 15.4b Set Ticket Data Response			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	7D	Set ticket data
Ticket data status flag	1 binary	00-01	00 = Flag currently false
			01 = Flag currently true
CRC	2 binary	0000-FFFF	16-bit CRC

Hosts utilizing extended validation support will likely use long polls 7B and 7C instead of long poll 7D. Data sent using 7B and 7C always takes precedence over data sent using long poll 7D.

## 15.5 LP 3D: Send Cash Out Ticket Information

When a gaming machine is configured for standard validation or communicating with a host that is not the validating controller, it will issue exception 3D (a cashout ticket has been printed) or 3E (handpay has been validated) to inform the host that a validation has been performed. Note that for all intents and purposes, exceptions 3D and 3E are functionally equivalent. The host may issue a type R long poll with a 3D command code to request the cash out ticket information. The gaming machine response, detailed in Table 15.5, includes an eight-digit (4 BCD) ticket validation number and the amount of the cash out in cents. If a gaming machine is configured to perform secure enhanced or system validation, it should not respond to long poll 3D to the validation controller. If it does respond to long poll 3D, it must not mark the validation record as having been read. When responding to a host that is not the validation controller, the gaming machine must return all zeros in the Validation Number field.

**Table 15.5**  
**Send Cash Out Ticket Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	3D	Send cash out ticket information
Validation number	4 BCD	XXXX	Standard validation number (calculated by the gaming machine)
Ticket amount	5 BCD	XXXXX	Ticket amount in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

## 15.6 LP 4C: Set Secure Enhanced Validation ID

For a gaming machine to perform secure enhanced ticket/receipt/handpay validation, the host must use the type S long poll detailed in Table 15.6a to set a gaming machine's validation ID number and initial validation sequence number. The host may also use this long poll to retrieve the current gaming machine validation ID and validation sequence number by issuing the 4C command with a gaming machine validation ID of zero. If a gaming machine is not configured to perform secure enhanced validation, or is responding to a host that is not the validation controller, it ignores this long poll.

**Table 15.6a**  
**Set Secure Enhanced Validation ID Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	4C	Set secure enhanced validation ID
Machine ID	3 binary	000000-FFFFF	Gaming machine validation ID number
Sequence number	3 binary	000000-FFFFF	Starting sequence number (incremented before being assigned to each event)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll 4C is detailed in Table 15.6b.

**Table 15.6b**  
**Set Secure Enhanced Validation ID Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	4C	Set secure enhanced validation ID
Machine ID	3 binary	000000- FFFFFF	Gaming machine validation ID number
Sequence number	3 binary	000000- FFFFFF	Current sequence number
CRC	2 binary	0000-FFFF	16-bit CRC

If the host resends the exact same gaming machine validation ID and sequence number that it most recently previously sent, and the gaming machine has since incremented the sequence number, the gaming machine must not reset the sequence number to the value sent but continue to use the current incremented value.

## 15.7 LP 57: Send Pending Cashout Information

When a gaming machine is configured for system validation, the host should be given an opportunity to provide the validation number for a pending cashout. When the gaming machine is ready to print a cashout ticket, it issues exception 57, system validation request. Exception 57 is a priority exception, and is sent at the next opportunity to respond to the host with an exception, even if other exceptions are pending. It must never be sent if the gaming machine is not waiting for system validation at the time it is polled. If the host does not respond with a long poll 57 or 58, the gaming machine reissues exception 57 every 800 milliseconds until the cashout is no longer waiting for system validation, such as the ten second timer expiring or link down detected.

When the host receives exception 57, it uses the type R long poll with a 57 command code to request the pending cashout information. The gaming machine response is detailed in Table 15.7a.

**Table 15.7a**  
**Send Pending Cashout Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	57	Send pending cashout information
Cashout type	1 binary	See table	Type of cashout (see Table 15.7b)
Amount	5 BCD	XXXXXX	Cashout amount in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

**Table 15.7b**  
**Cashout Type Code Values**

Code (binary)	Cashout Type
00	Cashable ticket
01	Restricted promotional ticket
80	Not waiting for system validation

## 15.8 LP 58: Receive Validation Number

After polling for the pending cashout information, the host may then issue the type S long poll with a command code 58 to provide the validation number, as detailed in the Table 15.8a. The host may also use long poll 58 following the exception 57 or long poll 57 to deny system validation.

**Table 15.8a**  
**Receive Validation Number Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine
Command	1 binary	58	Receive validation number
Validation System ID	1 BCD	XX	Validation system ID code (00 = system validation denied)
Validation number	8 BCD	XXXXXXXX	Validation number to use for cashout (not used if validation denied)
CRC	2 binary	0000-FFFF	16-bit CRC

If the Validation System ID field is 00, the system validation is denied. In this case the validation number field is not used. If system validation is denied the gaming machine must not print the cashout ticket. The gaming machine must then use another means to perform the cashout or abort it.

Note that the host may use long poll 58 to deny system validation without first issuing a long poll 57. A long poll 58 that specifies a valid validation number must be preceded by a valid long poll 57 within the same cashout.

The gaming machine response to long poll 58 is detailed in Table 15.8b.

**Table 15.8b**  
**Receive Validation Number Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	58	Receive validation number
Status	1 binary	00, 80-81	00 = command acknowledged 80 = Not in cashout 81 = Improper validation rejected
CRC	2 binary	0000-FFFF	16-bit CRC

If the link is down or the host does not issue a long poll 58 within ten seconds after the gaming machine begins its cashout process, or the host issues a long poll 58 specifying a validation number without first issuing a proper long poll 57, the gaming machine will proceed as though the system validation had been denied.

## 15.9 System Validation Examples

To demonstrate how system validation works, two examples are presented. The first will show a cashout ticket being validated by the system, and the second will show a cashout ticket being denied.

### 15.9.1 Example 1, Host validates cashout ticket

The player presses the cashout button with \$47.50 worth of cashable credits on the gaming machine. The gaming machine determines that it should print a cashout ticket for \$47.50. The gaming machine starts a ten second timer, and responds to the next general poll with exception 57. The host then polls for the cashout amount using the type R long poll 57, and the gaming machine responds as shown in Table 15.9.1a.

Table 15.9.1a Gaming Machine Response to Pending Cashout Information Command			
Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	57	Send pending cashout information
Cashout type	1 binary	00	Cashout type = cashable ticket
Amount	5 BCD	0000004750	Cashout amount in units of cents
CRC	2 binary	6D83	16-bit CRC

The host then calculates a validation number, and sends the long poll 58 command as detailed in Table 15.9.1b.

Table 15.9.1b Host Command to Receive Validation Number			
Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	58	Receive validation number
Validation System ID	1 BCD	01	Validation system ID code
Validation number	8 BCD	1234567890123456	Validation number to use for cashout
CRC	2 binary	349C	16-bit CRC

The gaming machine responds as shown in Table 15.9.1c.

**Table 15.9.1c**  
**Gaming Machine Response to Receive Validation Number Command**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	58	Receive validation number
Status	1 binary	00	Command acknowledged
CRC	2 binary	47EB	16-bit CRC

The gaming machine will then print the cashout ticket using the validation number provided by the host.

### 15.9.2 Example 2, Host refuses to validate cashout ticket

The player presses the cashout button with \$123.45 worth of cashable credits on the gaming machine. The gaming machine determines that it should print a cashout ticket for \$123.45. The gaming machine starts a ten second timer, and responds to the next general poll with exception 57. The host determines it is unable or unwilling to provide a validation number for any cashout, and sends the long poll 58 command in Table 15.9.2a.

**Table 15.9.2a**  
**Host Command to Receive Validation Number**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	58	Receive validation number
Validation System ID	1 BCD	00	System validation denied
Validation number	8 BCD	0000000000000000	Validation number (not used)
CRC	2 binary	BF91	16-bit CRC

The gaming machine responds as shown in Table 15.9.2b.

**Table 15.9.2b**  
**Gaming Machine Response to Receive Validation Number Command**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	58	Receive validation number
Status	1 binary	00	Command acknowledged
CRC	2 binary	47EB	16-bit CRC

At this point, the gaming machine will abort the cashout ticket and proceed with whatever other cashout method is available. That may be a hopper pay or a cancelled credits handpay.

### 15.10 LP 4D: Send Enhanced Validation Information

When a gaming machine is configured for secure enhanced or system validation, it will issue exception 3D (a cashout ticket has been printed) or 3E (handpay has been validated) to inform the host that an

unread validation record is in the buffer. Note that for all intents and purposes, exceptions 3D and 3E are functionally equivalent. The host may issue a type S long poll with a 4D command byte, as detailed in Table 15.10a, to look at or read the oldest unread validation in the buffer. This long poll is also used to retrieve previously read validation information that is still in the gaming machine's buffer. If a gaming machine is not configured to perform secure enhanced or system validation, or is responding to a host that is not the validation controller, it must either not respond to long poll 4D or return all zeros in the Validation Number field.

<b>Table 15.10a</b> <b>Send Enhanced Validation Information Command</b>			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	4D	Send enhanced validation information
Function code	1 binary	00-1F, FF	00 = read current validation info 01-1F = validation info from buffer index n FF = look ahead at current validation info
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll 4D is detailed in Table 15.10b.

**Table 15.10b**  
**Send Enhanced Validation Information Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	4D	Send enhanced validation information
Validation Type	1 binary	See table	Type of validation (see Table 15.13c on page 15-24)
Index number	1 binary	00-1F	Buffer position index number
Date	4 BCD	XXXX	Validation date in MMDDYYYY format
Time	3 BCD	XXX	Time in HHMMSS 24-hour format
Validation number	8 BCD	XXXXXXXX	Validation number (secure enhanced or system)
Amount	5 BCD	XXXXX	Ticket/handpay amount in units of cents
Ticket number	2 binary	0000-270F, FFFF	The sequential number printed on the ticket, starts at 0001, rolls over from 9999 to 0000 (FFFF for validations with no ticket)
Validation System ID	1 BCD	XX	00 = Secure enhanced validation number calculated by gaming machine 01-99 = System ID code (indicates validation number provided by host)
Expiration	4 BCD	XXXX	Expiration date printed on ticket in MMDDYYYY format, or 00000001-00009998 = number of days before ticket expires, 00009999 = never expires, 00000000 = no ticket printed or validation extensions not supported
Pool ID	2 binary	0000-FFFF	Restricted pool ID (0000 if not restricted or pool ID unknown)
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** When a validation number is printed on a ticket, it should include the 2 digit validation system ID followed by the 16 digit validation number, printed as an 18 digit validation number.

The function code supplied by the host controls the gaming machine's response mode.

Function code 00 causes the next unread ticket record to be returned. Once the gaming machine has received an implied ACK, it will mark the record as having been read. If unread records remain in the validation buffer, the gaming machine will then reissue exception 3D or 3E (according to the oldest unread validation). The gaming machine must continue to issue exception 3D or 3E every fifteen seconds as long as the host is reading the exceptions, and unread records remain in the buffer. If no unread records are in the buffer, all fields in the long poll 4D response will be zero, particularly the index number.

Function code FF allows the host to look at the next unread ticket record without marking the record as having been read. If no unread records are in the buffer, all fields will be zero.



For all other function codes, the validation record at the buffer index position corresponding to the function code will be returned. If the validation record was previously unread, it will continue to be considered unread. If the function code does not correspond to a valid buffer index, or the buffer position does not contain a valid record, all fields will be zero.

Please note, the amount does NOT include any partial amounts paid out of the hopper or to the credit meter, even in the case of a progressive handpay.

### 15.11 LP 70: Send Ticket Validation Data

When a ticket is inserted into a validator to be redeemed, in an acceptable condition with a machine readable validation number, the gaming machine issues exception 67 (ticket has been inserted). Note that, because ticket redemption is a time critical task, exception 67 takes priority over any other pending exceptions. If the link is down when the ticket is inserted, it should be returned to the player immediately without issuing exception 67.

When the host receives exception 67, it uses the type R long poll with a 70 command code to request the ticket's validation data. The host may respond to exception 67 with a long poll 70 immediately. The gaming machine variable length response is detailed in Table 15.11a. If a gaming machine is not configured for ticket redemption, it will never issue exception 67, and will ignore long poll 70.

**Table 15.11a**  
**Send Ticket Validation Data Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	70	Send ticket validation data
Length	1 binary	01-27	Number of bytes following, not including the CRC
Ticket status	1 binary	00 or FF	00 = ticket in escrow, data follows FF = no ticket in escrow
Ticket amount	5 BCD	XXXXX	Ticket amount in cents (all zeros if no amount available)
Parsing code	1 binary	00-FF	Validation data parsing code (see Table 15.11b)
Validation data	x bytes	See table	Ticket validation data (32 bytes max) (see Table 15.11b)
CRC	2 binary	0000-FFFF	16-bit CRC

**Note:** If the host sends the 70 long poll when a ticket is not being held in escrow, the gaming machine will respond with a ticket status of FF, and omit the remaining fields.

**Table 15.11b**  
**Validation Parsing Codes**

Code (binary)	Bytes	Parsing instructions
00	9 BCD	BCD-encoded 18 digit decimal validation number. The first two digits are a 2 digit system ID code indicating how to interpret the following 16 digits. System ID code 00 indicates that the following 16 digits represent a SAS secure enhanced validation number.
01	9 bytes	Foreign bill redemption encoded as: 1 byte Reserved 3 ASCII ISO 4217 alphabetic currency code (i.e. USD, CAD, etc.) 5 BCD Bill denomination (expressed in the currency's minor units) (See Section 18.5)

### 15.12 LP 71: Redeem Ticket

After the host has received the ticket validation data using long poll 70, it can authorize or reject the ticket by issuing long poll 71, as detailed in Table 15.12a. If a gaming machine is not configured for ticket redemption, it ignores long poll 71.

**Table 15.12a**  
**Redeem Ticket Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine
Command	1 binary	71	Redeem ticket
Length	1 binary	01-2D	Number of bytes following, not including the CRC
Transfer code	1 binary	See table	Ticket transfer code (see Table 15.12c)
Transfer amount	5 BCD	XXXXX	Ticket transfer amount, in cents
Parsing code	1 binary	00-FF	Validation data parsing code (see Table 15.11b)
Validation data	x bytes	See table	Ticket validation data (32 bytes max) (see Table 15.11b)
Restricted expiration	4 BCD	XXXX	Expiration date in MMDDYYYY format or 0000NNNN days format
Pool ID	2 binary	0000-FFFF	Restricted pool ID
Target ID length	1 binary	nn	Length of Target ID data following
Target ID	x ASCII	???	Target ID of targeted funds
CRC	2 binary	0000-FFFF	16-bit CRC

The restricted expiration and pool ID fields should only be included if the gaming machine has indicated it supports validation extensions in its long poll A0 response and the transfer code indicates a restricted type ticket. If omitted or set to all zeros, the default expiration and pool ID 0000 will be used. Please see Section 15.1, Improved Ticket Expiration Support, for a discussion on how to handle the restricted expiration field.

The Target ID field allows for an ASCII string that identifies the target group. A target group can be any game or group of games that the funds are targeted at, including an individual game, a select group

of games in a single theme or game type, or any other grouping as defined by the application. The Target ID associated with a specific target can be imbedded in each game, or configured by an operator, depending on the application. A game may have more than one Target ID associated with it. Funds from a targeted ticket can only be used for play on games associated with the specified Target ID. If no games matching the Target ID are available for play, the ticket must be returned to the player. The Target ID length must be zero or omitted for regular, non-targeted tickets. See Section 18.2 for more details.

---

**Note:** *The Restricted Expiration and Pool ID fields must always be included whenever the Target ID is included. For cashable and non-restricted funds, these fields should be set to all zeros.*

---

Targeted promotional tickets are metered the same as non-targeted tickets, based on the transfer code included in long poll 71.

The gaming machine response to long poll 71 is detailed in Table 15.12b.

Table 15.12b Redeem Ticket Response			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	71	Redeem ticket
Length	1 binary	01-27	Number of bytes following, not including the CRC
Machine status	1 binary	See table	Gaming machine status code (see Table 15.12d)
Transfer amount	5 BCD	XXXXX	Ticket transfer amount, in cents (all zeros if no amount available)
Parsing code	1 binary	00-FF	Validation data parsing code (see Table 15.11b)
Validation data	x bytes	See table	Ticket validation data (32 bytes max) (see Table 15.11b)
CRC	2 binary	0000-FFFF	16-bit CRC

---

**Note:** *If the communications link is down when the ticket is inserted or determined to be down while the ticket is in escrow, before the host sends the 71 long poll, the ticket should be returned to the player immediately. If the gaming machine does not receive the 70 long poll within ten seconds after the ticket is inserted, or receive the 71 long poll within 30 seconds after the ticket is inserted, the ticket should be returned to the player.*

---

A ticket redemption cycle begins when the gaming machine receives a 70 long poll, and ends when the next ticket's 70 long poll is received. The host may use long poll 71 to request the current ticket status at any time by setting the transfer code to FF and omitting the transfer amount, parsing code and validation data fields. After sending a valid Redeem Ticket long poll 71, the host may also request the current ticket status by resending the exact same command during that ticket redemption cycle. All fields in the response will be set to the current status of the most recent ticket redemption cycle. If there has been no previous ticket redemption cycle, the response will have a machine status of FF, and the transfer amount, parsing code and validation data fields are omitted.

If a host sends a different long poll 71 after having sent a Redeem Ticket command for the current redemption cycle, the poll must have no effect on the current redemption cycle. The gaming machine will respond with a machine status of C0, and omit the transfer amount, parsing code and validation data fields.

A ticket cannot be declared redeemed until it has been properly stacked. When a gaming machine receives a valid transfer that can be accepted, it cannot indicate success until the ticket has been irretrievably stacked. Therefore, it issues its long poll 71 response with status code 40, ticket redemption pending. Then, when the ticket has been either successfully stacked or rejected, the gaming machine issues exception 68, ticket transfer complete. The host should then issue long poll 71 with a transfer code of FF to get the completion status.

The gaming machine must reissue exception 68 every fifteen seconds until the host polls for and ACKs the completion status. Please note, exception 68 is *not* issued unless the gaming machine previously responded to a long poll 71 with status code 40, and has not subsequently responded to a long poll 71 with a status other than 40 and received a proper acknowledgement. It is the responsibility of the host to properly complete a ticket transaction. If another ticket is inserted after the gaming machine has stacked or returned the previous ticket, the gaming machine issues exception 67. If the host sends long poll 70 without polling for the completion status of the previous ticket, that status will be lost.

To redeem a ticket, a gaming machine must be able to accept the entire transfer amount. (Gaming machines with ticket printers may be able to accept ticket transfers that exceed the credit limit or are not an even multiple of the gaming machine denomination by printing a “change” ticket for the excess amount.) When a ticket transaction is rejected for any reason, the ticket must be returned to the player. When a ticket transaction is accepted, the ticket is stacked and the player credited with the ticket amount.

If the gaming machine currently has restricted amounts, it may not accept restricted amounts from a different pool. If the gaming machine refuses to redeem a ticket due to incompatible restricted amounts, the correct machine status code is 87, gaming machine unable to accept transfer at this time.

If the gaming machine currently has restricted amounts, and the host authorizes redemption of a restricted ticket or transfer of restricted amounts from the same pool but with a different expiration, the gaming machine selects an expiration for the combined amounts according to the following rules:

If both expirations are for a specific date, use the later date.

If both expirations are for “n” days, use the larger value of “n”.

If one expiration is for “n” days and the other is for a specific date, use the “n” days expiration.

**Table 15.12c**  
**Ticket Transfer Codes**

<b>Code (binary)</b>	<b>Status</b>
00	Valid cashable ticket
01	Valid restricted promotional ticket
02	Valid nonrestricted promotional ticket
10	Valid targeted cashable ticket (session play)
11	Valid targeted restricted promotional ticket (session play)
12	Valid targeted nonrestricted promotional ticket (session play)
80	Unable to validate (no reason given / other)
81	Not a valid validation number
82	Validation number not in system
83	Ticket marked pending in system
84	Ticket already redeemed
85	Ticket expired
86	Validation information not available
87	Ticket amount does not match system amount
88	Ticket amount exceeds auto redemption limit
90	Ticket not valid at this time
91	Ticket not valid on this gaming machine
92	Player card must be inserted
93	Ticket not valid for current player card
FF	Request for current ticket status

**Note:** Although gaming machines may only print cashable or restricted promotional tickets, support is provided here for redemption of nonrestricted promotional tickets. Please see Section 8, Advanced Funds Transfer Protocol, for details on management and metering of nonrestricted promotional amounts.

**Table 15.12d**  
**Gaming Machine Status Codes**

<b>Code</b> (binary)	<b>Status</b> (Note, 3 MSbits can be used to determine category of status code)
<b>Binary codes 000xxxxx indicate ticket redemption successful</b>	
00	Cashable ticket redeemed
01	Restricted promotional ticket redeemed
02	Nonrestricted promotional ticket redeemed
<b>Binary codes 001xxxxx indicate waiting for long poll 71</b>	
20	Waiting for long poll 71
<b>Binary codes 010xxxxx indicate ticket redemption pending</b>	
40	Ticket redemption pending (not complete)
<b>Binary codes 100xxxxx indicate ticket redemption failed</b>	
80	Ticket rejected by host, or unknown
81	Validation number does not match (response must include correct validation number)
82	Not a valid transfer function
83	Not a valid transfer amount (non-BCD)
84	Transfer amount exceeded the gaming machine credit limit
85	Transfer amount not an even multiple of gaming machine denomination
86	Transfer amount does not match ticket amount
87	Gaming machine unable to accept transfer at this time
88	Ticket rejected due to timeout
89	Ticket rejected due to comm link down
8A	Ticket redemption disabled
8B	Ticket rejected due to validator failure
90	No enabled games matching target group (session play)
91	Player cancelled redemption
<b>Binary codes 110xxxxx indicate incompatible poll</b>	
C0	Not compatible with current redemption cycle (ignored)
<b>Binary codes 111xxxxx indicate no validation information available</b>	
FF	No validation information available

**15.13 LP 50: Send Validation Meters**

Gaming machines that support ticket/receipt validation and/or handpay validation must keep track of the cumulative value in cents and the total number of validations performed for each type of validation supported. Gaming machines that support ticket redemption must keep track of the cumulative value in cents and the total number of tickets redeemed for each type of ticket supported. The host can obtain these meters by issuing a type S long poll with command code 50, as detailed in Table 15.13a.

**Table 15.13a**  
**Send Validation Meters Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	50	Send validation meters
Validation Type	1 binary	See table	Type of validation (see Table 15.13c)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll 50 is detailed in Table 15.13b.

**Table 15.13b**  
**Send Validation Meters Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	50	Send validation meters
Validation Type	1 binary	See table	Type of validation (see Table 15.13c)
Total validations	4 BCD	XXXX	Total number of validations of type
Cumulative amount	5 BCD	XXXXX	Cumulative validation amount in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

These validation meters are also included in Table C-7 in appendix C, starting with code 80.

**Note:** Meters reported using long poll 50 include only validations using the SAS protocol. They do not include amounts from any other process or protocol.

**Note:** *A cashable ticket or promotional ticket is a device delivered directly to the player, without attendant intervention. A handpay receipt is a device that is delivered to the attendant following a handpay. The receipt is not the cashout. The handpay is the cashout, and is metered in the appropriate cancelled credit or jackpot handpay meter. Only validated handpays are metered here, according to whether or not a receipt was printed.*

**Table 15.13c**  
**Validation Type Code Values**

<b>Code (binary)</b>	<b>Validation Type</b>
00	Cashable ticket from cashout or win, no handpay lockup
01	Restricted promotional ticket from cashout
02	Cashable ticket from AFT transfer
03	Restricted ticket from AFT transfer
04	Debit ticket from AFT transfer
10	Cancelled credit handpay (receipt printed)
20	Jackpot handpay (receipt printed)
40	Cancelled credit handpay (no receipt)
60	Jackpot handpay (no receipt)
80	Cashable ticket redeemed
81	Restricted promotional ticket redeemed
82	Nonrestricted promotional ticket redeemed

#### 15.14 Secure Enhanced Validation Algorithm

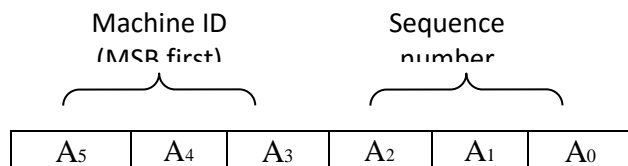
In secure enhanced validation mode, cash out ticket and handpay validation numbers are generated by the gaming machine using seed values provided by the host. The encoded number is calculated using the gaming machine validation ID and the current validation sequence number. The gaming machine validation ID is a 3 byte unsigned value assigned to the gaming machine by the host. The validation sequence number is a 3 byte unsigned value that is initialized by the host. The gaming machine validation ID and initial validation sequence number are provided by long poll 4C, Set Secure Enhanced Validation ID (see page 15-10). The host may change these values at any time it chooses. If the gaming machine is in the process of creating a validation number when new values are sent by the host, it may either finish creating the validation number from the existing values, then save the new values to be used for the next validation, or use the new values to create the current validation number.

The validation sequence number is **not** the sequential ticket number that is printed on every ticket. The validation sequence number is always incremented immediately prior to being used to create each validation number. Therefore, the actual validation sequence number is not used as is in the first validation number calculated following receipt of data in a long poll 4C.

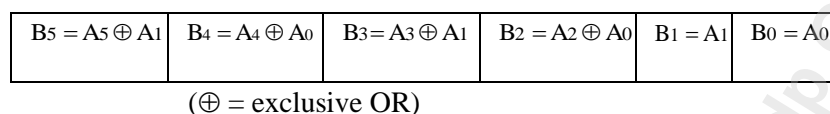
After incrementing the validation sequence number, the six binary bytes composed of the gaming machine validation ID and new validation sequence number are converted by the validation algorithm into a 16-digit BCD number that includes a check-digit. The following steps are employed in the encoding process.



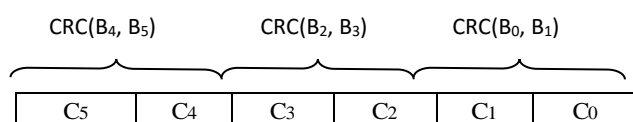
**Step 1:** Place the gaming machine validation ID and the sequence number in an array of 6 bytes.



**Step 2:** Array A gets transformed into array B as follows:

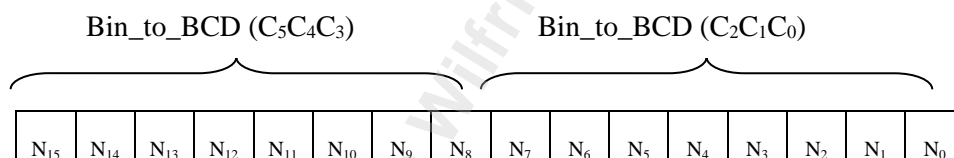


**Step 3:** Array B gets transformed into array C as follows:



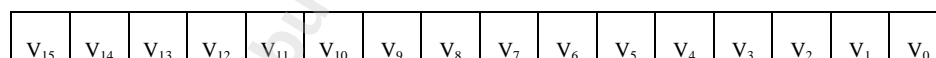
where CRC( B<sub>i</sub>, B<sub>j</sub>) represents a CRC calculation, as per Section 5, with seed 0, over the bytes B<sub>i</sub> and B<sub>j</sub> in the respective order.

**Step 4:** Array C gets transformed into an array of digits N as follows:



Where Bin\_to\_BCD (C<sub>i</sub>C<sub>j</sub>C<sub>k</sub>) represents the conversion from binary to BCD of the number C<sub>i</sub>C<sub>j</sub>C<sub>k</sub>

**Step 5:** The array of digits N gets transformed into the array of digits V as follows:



$$V_7 = N_7 \mid ((\sum_{j=0}^7 N_j) \bmod 5 \ll 1)$$

$$V_{15} = N_{15} \mid ((\sum_{j=8}^{15} N_j) \bmod 5 \ll 1)$$

V<sub>k</sub> = N<sub>k</sub> for all k 0 through 6, and 8 through 14.

The finished packed BCD validation number will be ordered with V<sub>15</sub> as the MSB and V<sub>0</sub> as the LSB.

Example:

Machine ID                      0x654321  
Sequence number                0x000001

Step 1	65	43	21	00	00	01
Step 2	65	42	21	01	00	01

Step 3	41			7D			29			53			19			D8		
Step 4	0	4	2	9	1	8	8	1	0	5	4	4	6	1	0	4		
Step 5	6	4	2	9	1	8	8	1	8	5	4	4	6	1	0	4		

The BCD validation number will be 6 4 2 9 1 8 8 1 8 5 4 4 6 1 0 4.

## SECTION 16 MULTI-DENOM EXTENSIONS

Gaming machines that allow the player to select from more than one denomination, or credit value, present a particular challenge for credit-based accounting protocols such as SAS. Most meters, such as Total Coin In and Total Coin Out, are defined in terms of “credits.” However, it is essential that the unit of measure used for meter reporting must not change dynamically due to player activity. To implement player selectable denominations in SAS, a base “accounting” credit value, or accounting denomination, must be established that can accurately represent any credit transaction that can occur on the gaming machine. All denominations available to the player must be evenly divisible by the accounting denomination. To prevent frequent meter rollover on 8 digit meters, it is recommended that a reasonable dynamic range be maintained between the base accounting value and a single maximum wager (maximum player credit value times max bet). In addition, no single win should be allowed to exceed the 4 BCD meter size.

To avoid confusion, it is important to use consistent terminology when referring to the various denomination values that can exist in a multi-denom gaming machine. The base denomination used for basic gaming machine accounting is called the **accounting denomination**. The denominations available to the player are called the **player denominations** or **game denominations**. The denomination of the coin mechanism and/or hopper is called the **token denomination**.

The base accounting denomination is reported to the host via long polls 1F and 53, and is the denomination to be used for all credit values reported to the host, except for those values specifically defined to be in a different unit of measure, such as cents, tokens, or units of game credits. Game credits refers to wager amounts without regard to denomination, such as “bet 5 credits.” **Please note, when the term “credits” is used in this document without qualifiers, such as “credit meter” or “cancelled credits,” it is not implying any specific method of storage or representation.**

Any gaming machine that reports a denomination via the 1F and 53 long polls that is or could be different from some player denomination must always be considered a multi-denom machine. However, a gaming machine does not actually need to offer the player more than one denomination in order to behave as a multi-denom machine. A multi-denom gaming machine can be configured such that only one player denomination is enabled. Also, a single denomination gaming machine may implement multi-denom extensions and report itself as a multi-denom gaming machine so long as it does so consistently. A gaming machine must report whether it supports multi-denom extensions in the long poll A0 response.

### 16.1 LP B0: Multi-Denom Preamble

The host may use the multi-denom preamble along with certain long polls to obtain player denomination-specific information. Multi-denom-aware long polls are listed in Table 16.1d. The preamble plus the base long poll always take the form of a variable length type S long poll. Table 16.1a shows the generic form of the multi-denom preamble long poll B0. If a gaming machine does not support multi-denom extensions, it ignores this poll.

**Table 16.1a**  
**Multi-Denom Preamble Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B0	Multi-denom preamble
Length	1 binary	02-FF	Total length of the bytes following, not including the CRC
Denomination	1 binary	00-3F	Binary number representing a specific denomination, or 00 for default response (see Table C-4 in Appendix C for denominations, see Table 16.1d for default responses)
Base command	1 binary	01-FF	Command byte for multi-denom-aware long poll (from Table 16.1d)
Data	varies	varies	Data appropriate for base long poll
CRC	2 binary	0000-FFFF	16-bit CRC

The generic form of the response to long poll B0 is shown in Table 16.1b.

**Table 16.1b**  
**Multi-Denom Preamble Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B0	Multi-denom preamble
Length	1 binary	01-FF	Total length of the bytes following, not including the CRC
Denomination	1 binary	00-3F	Binary number representing the requested specific denomination, or 00 for default response (see Table C-4 in Appendix C for denominations, see Table 16.1d for default responses)
Base command	1 binary	00-FF	Command byte for multi-denom-aware long poll response, or 00 if error
Data	varies	varies	Data appropriate for base long poll, or 1 byte binary error code from Table 16.1c
CRC	2 binary	0000-FFFF	16-bit CRC

If the base command cannot be executed or would normally be ignored, the “base command” response byte will be 00, and the data will be an error code from Table 16.1c. If the base command byte does not indicate an error, the data in the response will be whatever data the long poll would respond with if the multi-denom preamble were not present. For long polls that only respond with an ACK/NACK, the data will be the gaming machine’s polling address for ACK, or the polling address OR’d with hex 80 for NACK.

If the gaming machine is processing a time-sensitive task, it may send the gaming machine busy response in place of the normal preamble response, as defined in Section 4.1, without processing the base command. If the gaming machine intends to send the busy response specifically in response to the base command, it is sent as the data field in the preamble response, i.e., as the base command response.

**Note:** When evaluating implied acknowledgement or negative acknowledgement rules (see Section 3), the gaming machine must consider consecutive multidenom preamble polls with different base long polls to be different polls.

**Table 16.1c**  
**Multi-Denom Preamble Error Code Values**

Code (binary)	Error
01	Long poll not supported or ignored
02	Improperly formatted long poll
03	Not a multi-denom-aware long poll
04	Long poll not supported in that format for specific denomination (for example, requesting meter for specific game)
05	Not a valid player denomination

**Table 16.1d**  
**Multi-Denom-Aware Long Polls**

Poll	Description	Default Response (denomination = 00)
09	Enable/disable game n	Enable/disable game for all player denominations
11	Send total coin in meter	Send total coin in meter for gaming machine
12	Send total coin out meter	Send total coin out meter for gaming machine
14	Send total jackpot meter	Send total jackpot meter for gaming machine
15	Send games played meter	Send games played meter for gaming machine
16	Send games won meter	Send games won meter for gaming machine
17	Send games lost meter	Send games lost meter for gaming machine
2F	Send selected meters	Send selected meters for gaming machine
56	Send enabled game numbers	Send enabled game numbers for currently selected player denomination
6F	Send extended meters	Send selected meters for gaming machine
AF	Send extended meters (alternate)	Send selected meters for gaming machine
B5	Send extended game n information	Send game information for all player denominations for gaming machine (game=0000) or all player denominations for specified game

## 16.2 Multi-Denom Preamble Examples

### 16.2.1 Example 1

The normal response to long poll 56, Send Enabled Game Numbers, is a list of games currently available to the player. On a multi-denom gaming machine, the default response is a list of the games available to the player at the currently selected player denomination. The host may request a list of games for any specific denomination by using the multi-denom preamble. The long poll to request a list of games enabled for 5 cent play is detailed in Table 16.2.1a.

**Table 16.2.1a**  
**Host Command to Send Enabled Game Numbers for 5¢**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	B0	Multi-denom preamble
Length	1 binary	02	Number of bytes following, not including the CRC
Denomination	1 binary	02	Denomination code for \$0.05
Base command	1 binary	56	Send enabled game numbers
CRC	2 binary	DB63	16-bit CRC

If the gaming machine has two games enabled, 0003 and 0007, the response would be as shown in Table 16.2.1b.

**Table 16.2.1b**  
**Gaming Machine Response to Send Enabled Game Numbers for 5¢**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	B0	Multi-denom preamble
Length	1 binary	08	Number of bytes following, not including the CRC
Denomination	1 binary	02	Denomination code for \$0.05
Base command	1 binary	56	Send enabled game numbers
Length	1 binary	05	Number of bytes following, not including the CRC
Number of games	1 binary	02	Number of games currently enabled
Game number	2 BCD	0003	2-byte BCD game number
Game number	2 BCD	0007	2-byte BCD game number
CRC	2 binary	4EF3	16-bit CRC

### 16.2.2 Example 2

The normal response to long poll 2F, Send Selected Meters, is a list of requested meters for either the gaming machine or a specific game. On a multi-denom gaming machine, the host may request certain meters for any specific denomination by using the multi-denom preamble. Please note that the host may not combine a request for specific denominations with a request for specific games, i.e., when using the multi-denom preamble with long poll 2F, the game number must be 0000. Table C-7 in Appendix C details which meters are recommended to be supported for the overall gaming machine, which should be supported on a “per game” basis, and which should be supported on a “per denomination” basis. The long poll to request the Total Coin In meter for 25 cent play is detailed in Table 16.2c.

**Table 16.2c**  
**Host Command to Send Selected Meters for 25¢**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	B0	Multi-denom preamble
Length	1 binary	06	Number of bytes following, not including the CRC
Denomination	1 binary	04	Denomination code for \$0.25
Base command	1 binary	2F	Send selected meters
Length	1 binary	03	Number of bytes following, not including the CRC
Game number	2 BCD	0000	Must always be gaming machine
Requested meter	1 binary	00	Total Coin In meter
CRC	2 binary	0C56	16-bit CRC

If the 25 cent Total Coin In meter is 123,456, the response would be as shown in Table 16.2d.

**Table 16.2d**  
**Gaming Machine Response to Send Selected Meters for 25¢**

Field	Bytes	Value	Description
Address	1 binary	01	Gaming machine address
Command	1 binary	B0	Multi-denom preamble
Length	1 binary	0A	Number of bytes following, not including the CRC
Denomination	1 binary	04	Denomination code for \$0.25
Base command	1 binary	2F	Send selected meters
Length	1 binary	07	Number of bytes following, not including the CRC
Game number	2 BCD	0000	Gaming machine
Meter code	1 binary	00	Total Coin In meter
Meter value	4 BCD	00123456	Meter
CRC	2 binary	1952	16-bit CRC

**16.3 LP B1: Send Current Player Denomination**

When a gaming machine supports multi-denom extensions, the denomination currently selected by the player for game play is available via the type R long poll B1. The gaming machine response to B1 is detailed in Table 16.3. If a gaming machine does not support multi-denom extensions, it ignores this poll.

**Table 16.3**  
**Send Current Player Denomination Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B1	Send current player denomination
Current player denomination	1 binary	01-3F	Binary number representing the player denomination currently selected (see Table C-4 in Appendix C)
CRC	2 binary	0000-FFFF	16-bit CRC

With Real Time Event reporting enabled, the Game Start RTE will indicate whether the gaming machine supports multi-denom extensions, and if so, the player denomination the game was started for, the number of credits wagered at that denomination, and whether that game at that denomination is enabled for SAS progressives. See Section 12.5.3.

**16.4 LP B2: Send Enabled Player Denominations**

The host may use the type R long poll B2 to determine which denominations are currently available to the player. The gaming machine response to long poll B2 is detailed in Table 16.4. If a gaming machine does not support multi-denom extensions, it ignores this poll.

**Table 16.4**  
**Send Enabled Player Denominations Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B2	Send enabled player denominations
Length	1 binary	01-80	Total length of the bytes following, not including CRC
Number of denominations	1 binary	00-7F	Number of player denominations currently enabled
Player denomination	X binary	01-3F	Binary number representing the denomination, times the number of player denominations enabled (see Table C-4 in Appendix C)
CRC	2 binary	0000-FFFF	16-bit CRC



## SECTION 17 COMPONENT AUTHENTICATION PROTOCOL

The SAS Component Authentication Protocol allows the host to remotely verify that all executable programs and other fixed data stored within a gaming machine exactly matches the data that has been approved for operation in the local jurisdiction. Microprocessor-based peripheral devices connected to a gaming machine, such as bill validators and printers, may also be verified. The host may interrogate which software, firmware or peripheral components exist on a gaming machine, and request that the gaming machine perform authentication on a specific component. The host may select from any of the authentication methods supported by the component, and provide a seed and offset as appropriate.

A “component” is defined in this protocol to be some unit of logical organization of data. The data may be stored in one or more physical EPROMs, flash memory devices, disk files, etc. This includes fixed data such as executable code, paytables, graphics, sound data, etc. It is up to a gaming machine to organize program memory and data into logical groups.

A component may also be a peripheral device separate from the actual gaming machine, that the gaming machine is able to communicate with, such as a bill validator or a printer. The gaming machine may or may not be able to actually address the data memory within the peripheral. The gaming machine should, at a minimum, be able to determine the type of peripheral, manufacturer and version of firmware within the peripheral, and must be able to instruct the peripheral to perform authentication of its program memory.

Each component must be uniquely identified by an ASCII text string of up to 127 characters. ASCII text should include only printable characters in the range 20 hex through 7E hex. While not required by the protocol, the name of an approved component name should logically correspond to an identifier provided to the jurisdiction as part of the approval process. The name of a peripheral should uniquely identify the peripheral, including type, manufacturer and version. Peripheral manufacturers are encouraged to assign unique identifiers, so peripherals may be identified consistently across different manufacturers’ gaming machine platforms. A gaming machine is also encouraged to assign a unique ASCII name to each unique possible set of component data within the gaming machine. Peripherals should probably not be considered in determining the component set name.

A gaming machine that supports the Component Authentication Protocol will set Features2 bit 4 to one in its long poll A0 response.

### 17.1 LP 6E: Send Authentication Info

Using the type S long poll 6E, Send Authentication Info, the host can monitor and control the Component Authentication Protocol. The variable length command is detailed in Table 17.1a.

**Table 17.1a**  
**Send Authentication Info Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	6E	Send authentication info
Length	1 binary	01-AF	Number of bytes following, not including the CRC
Action	1 binary	00-03	Requested authentication action: 00 = Interrogate number of installed components 01 = Read status of component (address required) 02 = Authenticate component (address required) 03 = Interrogate authentication status

If action requires address specification, the following addressing data is included

Addressing mode	1 binary	00-01	00 = addressing by component index number 01 = addressing by component name
Index/name length	1 binary	01-7F	Length of address data following
Component index/name	x bytes	???	Binary component index if addressing mode = 00, ASCII component name if addressing mode = 01

If action = authenticate, the following authentication data is included

Method	4 binary	nnnnnnnn	Authentication method requested (see Table 17.1c)
Seed length	1 binary	00-14	Length of seed
Seed	x bytes	???	Authentication seed value
Offset length	1 binary	00-10	Length of offset
Offset	x bytes	???	Authentication offset value

CRC always included

CRC	2 binary	0000-FFFF	16-bit CRC
-----	----------	-----------	------------

The variable length gaming machine response is detailed in Table 17.1b.

**Table 17.1b**  
**Send Authentication Info Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Address of gaming machine responding
Command	1 binary	6E	Send authentication info
Length	1 binary	03-B1	Number of bytes following, not including the CRC
Component list CRC	2 binary	0000-FFFF	CRC (see Section 5) across all ASCII component names
Status	1 binary	nn	Status of component list, component, or error code if error (see Table 17.1d)

If status is for a component, the following data is included

Name length	1 binary	00-7F	Length of name data following
Name	x ASCII	???	ASCII list name or component name
Size length	1 binary	00-10	Length of size data following (if component is not byte-addressable, size length will be zero)
Size	x binary	???	Number of components if action = 00, or size of component
Available methods	4 binary	nnnnnnnn	Authentication methods supported by component (see Table 17.1c)

If status = authentication in progress or completed successfully,  
the following authentication data is included

Method	4 binary	nnnnnnnn	Authentication method in use (see Table 17.1c)
Authentication length	1 binary	00-14	00 if authentication in progress
Authentication data	x bytes	???	Authentication data if completed successfully

CRC always included

CRC	2 binary	0000-FFFF	16-bit CRC
-----	----------	-----------	------------

**Table 17.1c**  
**Authentication methods**

<b>Code</b> (binary)	<b>Method</b> (bit set if method supported/active)	<b>Seed size</b> (max bytes)	<b>Result size</b> (max bytes)
00000000	None	n/a	n/a
00000001	CRC 16 (using method from Section 5)	2 binary	2 binary
00000002	CRC 32	4 binary	4 binary
00000004	MD5	16 bytes	16 bytes
00000008	Kobetron I	4 ASCII	4 ASCII
00000010	Kobetron II	4 ASCII	8 ASCII
00000020	SHA1	20 bytes	20 bytes
00000040	SHA-256	32 bytes	32 bytes

**Note:** *If an authentication method does not explicitly include a seed in its published algorithm, any seed provided by the host is included in the authentication process before the actual component data.*

*Seed and result size specified in “bytes” indicates the result is an array of bytes, similar to ASCII, with the byte at array index 0 transmitted first, array index 1 transmitted second, etc. Unlike “printable” ASCII, each byte may be in the value of 00-FF.*

**Table 17.1d  
Authentication Status/Error Codes**

<b>Code (binary)</b>	<b>Status</b>
00	Status request successful
01	Installed component response
40	Authentication currently in progress (not complete)
41	Authentication complete (successful, data included)
<b>Status codes 80 through BF indicate component error status</b>	
80	Component does not exist
81	Component disabled or otherwise unavailable
82	Invalid command
<b>Status codes C0 through FE indicate authentication operation failed</b>	
C0	Authentication failed (reason unknown/unspecified)
C1	Authentication aborted (component list changed)
C2	Component does not support authentication
C3	Requested authentication method not supported
C4	Invalid data for requested authentication method
C5	Component cannot be authenticated at this time
<b>Status code FF indicates no authentication data</b>	
FF	No authentication data available

The send authentication info long poll 6E allows the host to explore the set of installed components and authenticate individual components. The desired operation is selected by the action flag. Some actions require a component address. The host may address a component by index number or ASCII name. Index number 0 is not valid. Index 1 is the first installed component, etc. No two components may have the exact same name. Note that names are case sensitive. The component name "GAME0001" is not the same as "game0001". If the index number is out of range, or the named component does not exist, the response will include error status code 80, component does not exist. All remaining data fields are omitted in this case.

Whenever the list of installed components is altered on a gaming machine, including but not limited to adding, removing, updating or rearranging components, the gaming machine will issue exception 8E, component list changed, to inform the host of the configuration change. If authentication is currently being performed, and the component being authenticated has not been changed, authentication should continue if possible. If an installed peripheral is currently not communicating, this must not, by itself, be considered a change to the installed component list. If authentication is being performed on the peripheral at the time communication is lost, this would likely cause the authentication to fail. However, the peripheral will still be reported as installed, with a status of “unavailable.” This is different from a peripheral that is intentionally removed, for example through operator configuration.

A component list CRC must be included in all status responses, and can also be used by the host to determine if the list of installed components changes in any way. The component list CRC is calculated by concatenating the ASCII text of all component names, in index order, and performing a 16-bit CRC (see Section 5) across the resulting string. The CRC must be recalculated whenever the component list changes (whenever exception 8E is issued).

### 17.1.1 Interrogate Number of Installed Components

The host may interrogate the number of installed components by setting the action flag to 00. The gaming machine response will include the component list CRC, an optional ASCII string identifying the specific collection of installed components, and the number of installed components. The status will be 01 and the available methods field will be zero.

### 17.1.2 Interrogate Status

The host may interrogate the current status of any installed component by setting the action flag to 01, read status of component. The host must specify which component the status is being requested for. The host may address the component by either an index number from 1 to the number of installed components, or by the component’s ASCII name. The gaming machine response will include the component list CRC, the status of the current component (from Table 17.1d), a unique ASCII string identifying the component, the size of the component (in bytes) if known, and a bit mask identifying which authentication methods the component supports. Components that do not support offset specification must report a size of zero.

If the status request is for a component that is currently performing authentication, the response will indicate the current authentication status. See Section 17.1.3 for details. For all other valid status requests, the status will indicate the current status of the component. If the status request is for a component that does not exist, the status response will be 80, component does not exist, and the remaining data fields will be omitted.

### 17.1.3 Authenticate Component

The host may request authentication of any installed component by setting the action flag to 02, authenticate component. The host must provide a component index or name, the desired authentication method, and the relevant seed and offset data. Table 17.1c details the maximum seed size for each authentication method. The offset cannot be greater than the component size. If the addressed component does not exist or does not support the requested authentication method, or if the seed or offset are out of range, the gaming machine will respond with an appropriate error message from Table 17.1d and authentication will not be performed. Otherwise, the gaming machine will respond with status 40, authentication in progress.

Depending on the authentication method used and the size of the component, authentication may take some time to complete. Ideally, the gaming machine should perform authentication as quickly as possible. However, if the gaming machine is playable at the time authentication is being performed, the authentication should be performed in a manner that does not negatively

impact game play. Therefore, it is reasonable to expect authentication to complete quicker if the gaming machine is disabled before authentication is performed.

The host may interrogate status for the component performing authentication at any time, either by setting the action flag to 01 and specifically addressing the component being authenticated, or setting the action flag to 03, interrogate authentication status. If the host requests authentication status using action 03 and there is no authentication status to report, the status flag will be set to FF and the remaining fields omitted. Otherwise the status flag will indicate the authentication status and the remaining fields will identify the component, and its authentication data if any.

Until the authentication is complete, the status response will continue to be 40. When authentication completes successfully, fails or is aborted, if the host has not read the authentication completion status the gaming machine will issue exception 8F, authentication complete. This is a process exception for the exclusive purpose of communicating the current authentication state to the host. It is not inserted in the exception queue, and is never issued to a host not performing authentication. It is only issued if an authentication completion status is available, the status has not been reported and acknowledged, and no priority exceptions are pending. The exception must be reissued every 15 seconds until the final authentication status is reported and acknowledged.

Authentication may only be performed on one component at a time. If the host requests authentication using action code 02 while an authentication is already in progress, the authentication in progress will be aborted, and the new authentication request will be processed. The gaming machine must act as though the prior authentication had never been requested, and not issue exception 8F.

It is important to note that it may not always be possible to authenticate some memory or peripherals while the game is being played without negatively impacting the gaming machine's operation. Many peripherals will stop functioning during the time authentication is in progress. For a touchscreen or bill validator to go "dead" for several seconds during game play for no apparent reason is probably not acceptable in most gaming jurisdictions.

Notwithstanding jurisdictional requirements to the contrary, authentication is not required to be implemented in a way that would substantially detract from normal game play. Gaming machine program memory that may influence game outcomes must be available for authentication at any time. Gaming machine manufacturers are encouraged to provide access to authenticate other program memory and peripherals whenever reasonably possible. However, it must be understood that some memory and/or peripherals may not support authentication, and authenticating some memory and/or peripherals may interfere with normal game play. Therefore, the minimum acceptable implementation is to provide access to those peripherals and non-critical memory areas that support authentication only when the credit meter is zero and the gaming machine is disabled. Jurisdictions or systems may specify other times when authentication must be available.

For components that do not support authentication, such as display memory not readily accessible to the main processor, the response to long poll 0x6E will indicate no available methods. (It may still be advantageous to report that such a component exists, to aid in system verification of correct component sets.)

All components that support authentication will report the methods that they support, even if authentication is not permitted in the current game state. If a supported authentication method is requested and the component is functioning properly, but the component is not currently available for authentication, the correct status response is error code C5, component cannot be authenticated at this time.





## SECTION 18 NEW IN SAS 6.03

The following features are new since SAS 6.02 was published November 11, 2005.

### 18.1 Meter Change Notification Support

Previously published in SAS 6.02 addenda, May 5, 2008

Meter change notification support provides for SAS host notification of pending changes to a gaming machine's meter map or enabled paytables due to activities such as remote game download and remote configuration (Server Based Gaming). Advance notification of a pending change will allow slot accounting systems to make an orderly transition to the new meter map or configuration. Gaming machines that support Meter Change Notification should set Features Codes 3, bit 2, to one in the A0 long poll response.

#### 18.1.1 LP B6: Meter Collect Status

The gaming machine will notify the SAS host of a scheduled activity that may clear meters, change the meter mapping, or change which paytables or denominations are available to the player by issuing exception A0, meter change pending. The gaming machine must not allow game play while it is in this meter change pending state.

The host may use the type S long poll B6 to acknowledge the receipt of exception A0, request that the meters not be changed at this time, or indicate that it is ready for the activity to proceed. This variable length command is detailed in Table 18.1.1a.

Table 18.1.1a Meter Collect Status Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B6	Meter collect status
Length	1 binary	01	Total length of the bytes following, not including the CRC
Status	1 binary	00-FF	Current host status 00 = Acknowledge A0 01 = Ready (meters collected) 80 = Unable to collect meters at this time
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll B6 is detailed in Table 18.1.1b.

**Table 18.1.1b**  
**Meter Collect Status Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B6	Meter collect status
Length	1 binary	01	Total length of the bytes following, not including the CRC
Status	1 binary	00-FF	00 = Meter clear/ add/remove/remap activity 01 = Enable/disable payable/denom only 80 = Not in meter change pending state
CRC	2 binary	0000-FFFF	16-bit CRC

Figure 18.1.1 (below) shows a flow chart of the gaming machine meter change notification process. Exception A0 is issued to notify the host that there is a pending change to meters or enabled paytables or denoms. Exception A0 is a priority exception. Exception A0 is reissued every five seconds until either long poll B6 is received (with any status), or the scheduled activity is cancelled. If the gaming machine does not receive any long poll B6 in 30 seconds, the meter change pending state will be cancelled.

If the meter change pending state is cancelled for any reason after the gaming machine has sent exception A0, the gaming machine will issue exception A1, meter change cancelled.

When the host receives exception A0, it should send long poll B6 with status 00, Acknowledge. When the host has collected all of the gaming machine meters that it tracks or determines it does not need to collect meters, and is ready for the activity to occur, it sends long poll B6 with status 01, Ready. The gaming machine response will always indicate the type of activity that is scheduled.

If the host is in a state where it is unable to collect final meters on the gaming machine, for example due to a scheduled drop, it may send long poll B6 with status 80, Unable. In this case, the meter change pending state will be cancelled, and exception A1 will be issued. Note that the Ready or Unable status may be sent in response to exception A0 without first sending Acknowledge.

If the gaming machine receives long poll B6 when it is not expecting it, i.e. A0 had not been sent or the activity has already been cancelled, it should respond with status 80, not in meter change pending state. In all other cases, including a valid “Unable” status from the host, the correct response is status 00 or 01.

When the gaming machine receives the Acknowledge status, it will allow the host 30 seconds to complete the meter collection process. The host may restart this timer as many times as necessary by resending Acknowledge. If this 30 second timer expires, the gaming machine may restart the process from the very beginning by issuing exception A0 again. In this case, the activity has not been cancelled, so exception A1 is not issued. The host will then again have 30 seconds before the meter change pending state is cancelled. A0 will again be reissued every 5 seconds while this delay is in progress. The gaming machine may also cancel the meter change pending state and issue A1 if the host fails to complete the process in a reasonable amount of time.

After the host has sent the Ready status and the gaming machine has finished the activity, the host may see several exceptions, depending on the nature of the activity. The gaming machine may stop responding for a period of time. The activity may have required a restart of the gaming machine, causing exception 17 to be sent. If the game set was changed, exception 36 or 38 will

be issued. If any lifetime-to-date meters were initialized, cleared, added, removed, or rearranged, exception 7A will be sent.

The gaming machine will issue exception A2, enabled games/denoms changed, whenever the set of paytables available to the player is changed. Exception A2 is sent even if no meters were affected. This exception is sent in addition to any other exceptions that may be appropriate, such as 3C.

Because A0 is only issued if lifetime-to-date meters are to be changed and/or the enabled games/denoms are to be changed, the host can assume it will always get at least exception 7A or A2 when the activity is complete, or A1 if the activity is cancelled.

If exception 36 or 38 is received and some or all game meters were not cleared, the host should verify the correlation between game numbers and payable IDs. Pre-existing paytables may have had their game numbers changed without affecting the meters for those paytables.

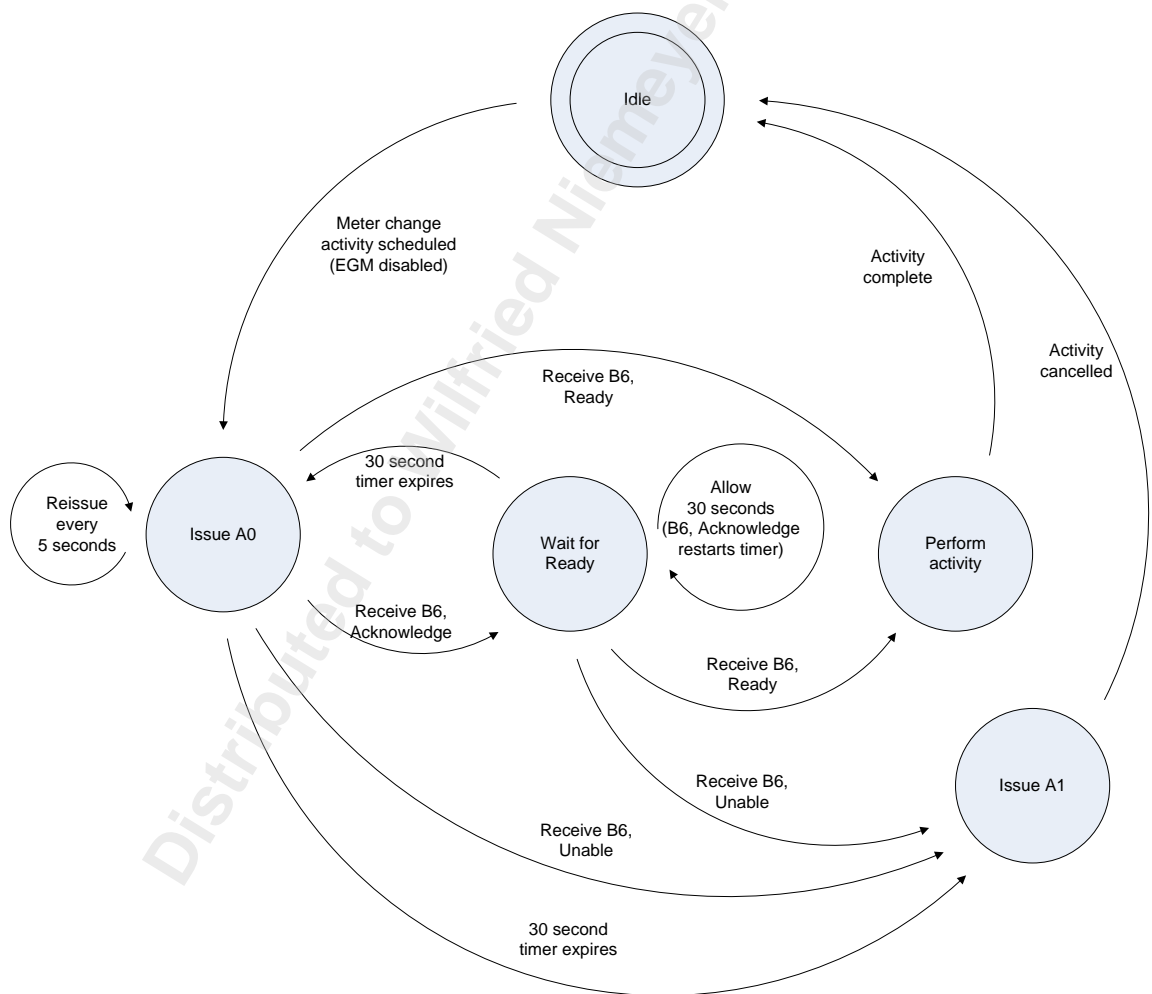


Figure 18.1.1 Meter Change Notification Flow Chart

### 18.1.2 LP B7: Set Machine Numbers

The host may use the type S long poll B7 to set the asset number and floor location in the gaming machine, as well as interrogate the current values on the gaming machine. Long poll B7 should

be supported regardless of whether the host is able to change any values. This variable length command is detailed in Table 18.1.2a.

**Table 18.1.2a**  
**Set Machine Numbers Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	B7	Set machine numbers
Length	1 binary	05-nn	Total length of the bytes following, not including the CRC
Asset number	4 binary	nnnnnnnn	New gaming machine asset number or house ID (use 0 for interrogate only)
Floor location length	1 binary	00-28	Length of gaming machine floor location (use 0 for interrogate only)
Floor location	x ASCII	???	New gaming machine floor location
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response to long poll B7 is detailed in Table 18.1.2b.

**Table 18.1.2a**  
**Set Machine Numbers Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Length	1 binary	06-nn	Total length of the bytes following, not including the CRC
Control flags	1 binary	00-FF	Bit Description
			0 0 = No host control of asset number 1 = Asset number may be changed by host
			1 0 = No host control of floor location 1 = Floor location may be changed by host
			7~2 TBD (leave as 0)
Asset number	4 binary	nnnnnnnn	Current asset number or house ID (0 if no asset number)
Floor location length	1 binary	00-28	Length of gaming machine floor location (0 if no floor location)
Floor location	x ASCII	???	Current floor location
CRC	2 binary	0000-FFFF	16-bit CRC

For security purposes, a gaming machine may limit or prohibit the ability of a host to set some or all machine numbers. Changing the values may require an operator at the machine, or control may be limited to a different host. The control flags in the B7 response indicate which values, if any, may be changed by that host. A zero in either field indicates that the host is not allowed to change the associated value. The response must always include the current values on the gaming machine. If the host sends a new value and the change is successful, the response will include that new value. If the gaming machine indicates the host may change the value but the response still includes the old values, this indicates the gaming machine is in a state where the value cannot

be changed at this time. The host will need to attempt the change at a later time. The host should not attempt to change these values when credits are on the machine.

## 18.2 Session Play Support

Previously published in SAS 6.02 addenda, May 5, 2008

Session Play Support provides for session-based gaming and targeted ticketing. A session is a set of individual games purchased as a group and played in a series. Targeting provides a method to restrict credits to a particular theme, payable, etc. Session Play Support provides a method for the host to determine whether a gaming machine has any session-based paytables, adds a meter to count the number of sessions that have been played on the gaming machine, modifies jackpot reporting for jackpot wins within a session (see Section 7.8), adds Session Start and Session End real time events (see Section 12.5), and provides for tickets that are good only for session play.

Gaming machines that have session-based paytables indicate such by setting Feature Codes 3, bit 4, to one in the A0 long poll response. For game number 0000, bit 4 is set to 1 if any implemented payable supports session play, whether or not that payable is currently enabled for play. For all other game numbers, bit 4 is set to 1 if that payable supports session play.

Meter 0079 counts the number of sessions that have been played on the gaming machine, and the number of sessions that have been played on each session-based payable. If a gaming machine does not have session-based paytables, it can choose to not support this meter, or it can safely return zero as the number of sessions played.

### 18.2.1 Session Play Jackpot Reporting

A jackpot that occurs during a session will likely require special handling, because the actual amount that the player will win for the session has not yet been determined. To facilitate this special handling, long poll 1B includes two new fields, as shown in Table 7.8. When a win over the jackpot limit occurs within session play, the “Amount” field is reported as zero, and the two new fields report the details about the win amount. In this way, a legacy system will not automatically process the jackpot amount as a handpay.

### 18.2.2 Exc 9E, 9F: Session Play Start/End

Game Start and Game End events are reported for each individual game within a session. When a session is initiated, the total amount wagered for the entire session is reported in the first Game Start real time event. (If this amount is greater than 9999 credits, it must be spread across the first two or three games played in the session, as necessary.) Subsequent Game Start events within the session will report a wagered amount of zero. The actual amount added to the Coin In meter for each game includes a percentage of the total session wager plus any positive amount deducted from the session win meter for that game. The Game End real time event for each game played reports only the positive win amount added to the session balance.

To support Session Start and Session End notification, two new exceptions have been added. Exception 9E, Session Start, is reported immediately before the first Game Start exception of the session, and exception 9F is reported following the last Game End exception of the session. These exceptions include additional data when in Real Time Event mode, as detailed in Section 12.

### 18.2.3 Targeted Restricted Promotional Tickets

To support targeted restricted promotional tickets, Target ID length and Target ID have been added to long poll 71, Redeem Ticket (see Table 15.12a). New ticket transfer codes have been added to Table 15.12c and new gaming machine status codes have been added to Table 15.12d.

### 18.3 Tip Money Meter Support

Previously published in SAS 6.02 addenda, May 5, 2008

The Tip Money meter (see Meter Code 007A in Table C-7) is used to indicate the portion of a machine paid win that was removed from a gaming machine specifically to be used as a tip for the dealer. The money must also be metered in the Total Jackpot meter, and either the Total Attendant Paid Paytable Win, Total Attendant Paid Progressive Win, or Total Attendant Paid External Bonus Win meter, as appropriate. If the entire win is required to be paid by an attendant, for example due to the total amount being over the Jackpot Limit, any portion used as a tip must be handled external to the gaming machine. The entire amount will be metered in the Total Jackpot meter, exception 5F is not issued, and the Tip Money meter must not increment in this case. The Tip Money meter is not used to meter funds cashed out from the credit meter for the purpose of tipping an attendant or server. Whenever a win results in funds being added to the Tip Money meter, the gaming machine will issue exception 5F, Tip Awarded.

### 18.4 Keyed Credits Support

Previously published in SAS 6.02 addenda, May 5, 2008

The following provides support for keyed-on and keyed-off credits. This is a feature used in some jurisdictions, particularly in Europe, where an operator may add credits to a gaming machine or remove credits from a gaming machine using an operator menu.

For backwards compatibility with existing accounting systems, keyed-on credits are metered as electronic transfers to the gaming machine, and keyed-off credits are metered as electronic transfers from the gaming machine. While no communication protocol is used to add or remove these credits, there likely will be a ledger or account maintained by the casino, and from the perspective of the SAS host these transfers appear to be no different from transfers that are performed using a communications protocol other than SAS.

Keyed-on credits may be regular cashable, nonrestricted promotional, or restricted promotional funds. The type of funds added to the gaming machine is up to the design of the keyed-on credits functionality, and it is not required or expected by the SAS protocol that any particular type of credits may be keyed-on to a machine. Keyed-off credits must be metered as to the type of funds that are removed. Again, it is up to the implementation as to what types of credits may be keyed-off, whether partial key-offs are allowed, etc. Credits may also be removed through a normal handpay process.

Keyed-on credits are a component of Total Drop, and keyed-off credits are a component of Total Cancelled Credits. Keyed-on and keyed-off credits are added to the appropriate gaming machine electronic transfer meters. In addition, the following new meters are defined in Table C-7 to allow specific tracking of these credit movements.

- Meter 00FA, Regular cashable keyed-on funds
- Meter 00FB, Restricted promotional keyed-on funds
- Meter 00FC, Nonrestricted promotional keyed-on funds
- Meter 00FD, Regular cashable keyed-off funds
- Meter 00FE, Restricted promotional keyed-off funds
- Meter 00FF, Nonrestricted promotional keyed-off funds

All funds added to these meters must also be added to the appropriate regular accounting meters. The following table shows the meters from Table C-7 that increment for each type of keyed-on or keyed-off transaction. This also applies to other functionally equivalent SAS meters, such as long poll 13, Total Drop.

Regular Gaming Machine Accounting Meters affected by Keyed Credit Transfers		
Keyed Meter	EGM Meter	Description
<b>00FA</b>		<b>Regular cashable keyed-on funds (credits)</b>
	0017	Total electronic transfers to gaming machine (credits)
	0024	Total drop (credits)
	002E	Electronic regular cashable transfers to gaming machine (credits)
<b>00FB</b>		<b>Restricted promotional keyed-on funds (credits)</b>
	0017	Total electronic transfers to gaming machine (credits)
	0024	Total drop (credits)
	002F	Electronic restricted promotional transfers to gaming machine (credits)
<b>00FC</b>		<b>Nonrestricted promotional keyed-on funds (credits)</b>
	0017	Total electronic transfers to gaming machine (credits)
	0024	Total drop (credits)
	0030	Electronic nonrestricted promotional transfers to gaming machine (credits)
<b>00FD</b>		<b>Regular cashable keyed-off funds (credits)</b>
	0004	Total cancelled credits
	0018	Total electronic transfers to host (credits)
	0032	Electronic regular cashable transfers to host (credits)
<b>00FE</b>		<b>Restricted promotional keyed-off funds (credits)</b>
	0004	Total cancelled credits
	0018	Total electronic transfers to host (credits)
	0033	Electronic restricted promotional transfers to host (credits)
<b>00FF</b>		<b>Nonrestricted promotional keyed-off funds (credits)</b>
	0004	Total cancelled credits
	0018	Total electronic transfers to host (credits)
	0034	Electronic nonrestricted promotional transfers to host (credits)

## 18.5 Foreign Currency Redemption Support

Previously published as an addendum, June 16, 2011

The following provides support for redeeming bills in multiple currencies on one gaming machine. The legacy SAS bill exceptions and long polls must still be supported for the “native” currency of the gaming machine. The gaming machine’s accounting denom and all player denoms must be based on the native currency, and all gaming machine meters that are reported in cents must be reported using the native currency. Foreign bills are added to the Total Drop meter, the same as native bills. However, all legacy SAS bill meters only include bills in the native currency and the legacy bill exceptions are only used for native currency. New polls provide access to the foreign bill meters, as well as an alternate method to access the native bill meters. Gaming machines that support foreign currency redemption must indicate such by setting Feature Codes 3, bit 5, to 1 in the A0 long poll response.

### 18.5.1 Ticket Redemption Process Enhancements

“Foreign” bills are redeemed using the voucher redemption process. The gaming machine is not responsible for calculating the bill’s local value using the exchange rate. The gaming machine identifies the bill in the long poll 70 response as a foreign bill by using a parsing code of 01, and includes the ISO 4217 standard alphabetic currency code (i.e. USD, CAD, etc.) and the bill denomination (in the minor units of the currency) where the validation number would normally be. The host uses the current conversion rate to determine the value in local currency, and authorizes redemption for that amount using long poll 71. Note that the ISO 4217 standard currency code is used instead of the SAS country code to provide support for a larger range of currencies, and the actual bill denomination is used instead of the SAS bill codes to support more bill denominations. The SAS country codes and bill codes are still used to represent the native currency in the legacy bill support.

Long poll 71 also includes a text string representing the conversion rate, using the Target ID field (see Session Play Support). For example, if the exchange rate is 1.01548, the string “1.01548” is sent to the gaming machine. The gaming machine may use this string to display the current conversion rate to the player and request confirmation of the exchange. The exchange rate is not used by the gaming machine to determine or confirm the actual bill value. If the player accepts the conversion, the gaming machine stacks the bill and adds the local currency value provided by the host to the credit meter.

The gaming machine may show the local currency value and conversion rate to the player. For redemption of a \$20 US bill, for example, the gaming machine may display something like:

**\$20 USD = \$20.30 CAD**  
**\$1 USD = \$1.01548 CAD**

**Accept conversion? [Yes] [No]**

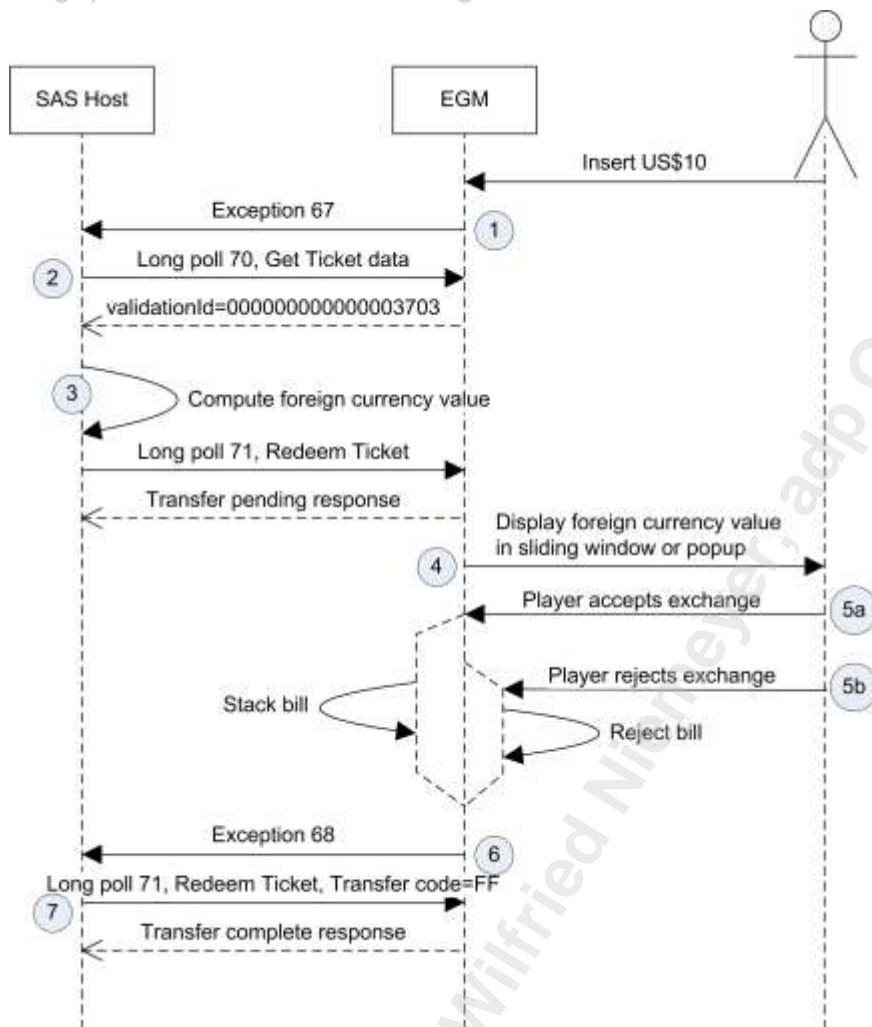
If the player accepts the value, the gaming machine will stack the bill, credit the player, and report a successful redemption to the host.

Table C-7 meter codes 007B and 007C report Total Foreign Bill Converted Amount and Total Foreign Bill Count. The gaming machine also must maintain meters to track the total count and total value for each currency code/bill denomination that it can accept. The host may use long polls 59, 5A and 5B to determine which currencies a gaming machine accepts, which bills a gaming machine supports for each currency, and to read meters for those bills. This includes both the legacy and foreign currencies. For compatibility with legacy casino accounting systems, a gaming machine by default must add foreign bills to the standard Voucher In meters when reporting those meters to the host. Long poll 5C allows a host to disable this legacy reporting mode.

Due to the maximum length of a SAS poll, there is a hard limit to the number of currencies and bill denominations that can be reported to the host. These numbers are sufficiently large that they should never present a problem in a real application. The host is permitted to further limit the currencies and bill denominations that it supports. If the gaming machine reports currencies and/or bill denominations that the host does not support, the host will simply refuse to redeem bills of those currency/denomination combinations. It is the responsibility of the gaming machine and host to support the currencies and bills required for a specific application.

The following flow chart shows the foreign currency redemption process.





1. Gaming machine interprets foreign currency as a special voucher and requests redemption (generate SAS exception 67).
2. Host issues long poll 70 to get voucher information. The foreign currency is identified by using a new parsing code and creating a validation ID which encodes the currency type and denomination (e.g., "US"=37 table C-5, "\$10"=03 table C-6, validation Id = 000000000000003703).
3. Host computes the bill value in the native currency, authorizes the redemption for amount in native currency, and provides the exchange rate using SAS long poll 71.
4. Gaming machine displays the foreign bill value in the native currency and the exchange rate returned by host.
- 5a. Player accepts exchange, bill is stacked and credit meter is incremented.
- 5b. Player refuses exchange and bill is rejected.
6. Gaming machine generates exception 68.
7. Host issues long poll 71 (w/Transfer Code FF) to determine final disposition of bill.

### 18.5.2 LP 59: Send Enabled Currency Codes

The host may use the type R long poll 59 to determine the ISO 4217 alphabetic currency codes for all of the currencies that are currently enabled on a gaming machine. The variable length gaming machine response to long poll 59 is detailed below in Table 18.5.2.

<b>Table 18.5.2</b> <b>Send Enabled Currency Codes Response</b>			
<b>Field</b>	<b>Bytes</b>	<b>Value</b>	<b>Description</b>
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	59	Send enabled currency codes
Length	1 binary	03-nn	Number of bytes following, not including CRC
ISO 4217 currency code	3 ASCII	xxx	ISO 4217 alphabetic currency code of first enabled currency
...	variable	...	Additional currency codes
CRC	2 binary	0000-FFFF	16-bit CRC

Long poll 59 must report all currency codes currently enabled on the gaming machine, including the local currency.

### 18.5.3 LP 5A: Send Supported Bills

The host may use the type S long poll 5A to determine the bills that are supported for a specific currency. This variable length command is detailed in Table 18.5.3a.

<b>Table 18.5.3a</b> <b>Send Supported Bills Command</b>			
<b>Field</b>	<b>Bytes</b>	<b>Value</b>	<b>Description</b>
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5A	Send supported bills
Length	1 binary	03	Number of bytes following, not including the CRC
ISO 4217 currency code	3 ASCII	xxx	ISO 4217 alphabetic currency code
CRC	2 binary	0000-FFFF	16-bit CRC

The variable length gaming machine response to long poll 5A is detailed below in Table 18.5.3b.

**Table 18.5.3b**  
**Send Enabled Currency Codes Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5A	Send supported bills
Length	1 binary	03-nn	Number of bytes following, not including CRC
ISO 4217 currency code	3 ASCII	xxx	ISO 4217 alphabetic currency code
Bill denomination	5 BCD	XXXXX	Denomination of first supported bill (expressed in the currency's minor units)
...	variable	...	Additional denominations
CRC	2 binary	0000-FFFF	16-bit CRC

If the requested currency is currently enabled, long poll 5A must report all denominations supported for the specified currency, regardless of whether the specific denominations are currently enabled for acceptance.

#### 18.5.4 LP 5B: Send Bill Meters

The host may use the type S long poll 5B, Send Bill Meters, to read the count and converted amount meters for a specific currency code/bill denomination, or the count and converted amount meters for the sum of all bills for a specific currency. This poll is particularly useful for obtaining foreign bill redemption information, where the value of each bill can vary based on the current exchange rate, and the meters are not otherwise available to the SAS host. This poll may also be used to obtain bill meters for the local currency. This variable length command is detailed in Table 18.5.4a.

**Table 18.5.4a**  
**Send Bill Meters Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5B	Send bill meters
Length	1 binary	08	Number of bytes following, not including the CRC
ISO 4217 currency code	3 ASCII	xxx	ISO 4217 alphabetic currency code
Bill denomination	5 BCD	XXXXX	Denomination (in the currency's minor units), or 0 for sum of all bills for specified currency
CRC	2 binary	0000-FFFF	16-bit CRC

The variable length gaming machine response is to long poll 5B detailed in Table 18.5.4b.

**Table 18.5.4b**  
**Send Bill Meters Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5B	Send bill meters
Length	1 binary	0A-nn	Number of bytes following, not including the CRC
ISO 4217 currency code	3 ASCII	xxx	ISO 4217 alphabetic currency code
Bill denomination	5 BCD	XXXXX	Denomination (in the currency's minor units) , or 0 for sum of all bills for specified currency
Amount meter size	1 binary	00-09	Amount meter size in number of bytes
Amount meter value (cents)	x BCD	???	Converted amount meter value (0 to 9 bytes) in cents (minor units of local currency)
Count meter size	1 binary	00-09	Count meter size in number of bytes
Count meter value	x BCD	???	Count meter value (0 to 9 bytes)
CRC	2 binary	0000-FFFF	16-bit CRC

If the specified currency is enabled, the meters for the requested bill denomination must be returned, regardless of whether or not that bill is currently enabled. If meters are requested for a currency that is not enabled or a bill denomination that is not supported, the length of both meters should be zero.

### 18.5.5 LP 5C: Foreign Bill Reporting Mode

By default, for compatibility with legacy accounting systems that are not aware of foreign bill redemption meters, foreign bill meters must be added to the gaming machine Voucher In meters when reporting those meters to the host. This includes Table C-7 meters 0015, 0028, 0029, 002A, 002B, 0035, 0036, and 0037, according to the type of funds transferred in long poll 71. Gaming machines must default to this reporting mode on initial power up and when recovering from a power down condition.

Any host that is aware of long polls 59, 5A and 5B may set the gaming machine to report the true Voucher In meters, not including foreign bill amounts. A host that wants to get the true Voucher In meters may issue the variable length long poll 5C to enable foreign bill reporting mode as detailed in Table 18.5.5a. The host should send this poll following communications initialization and following any link down condition, before collecting any voucher meters.

**Table 18.5.5a**  
**Foreign Bill Reporting Mode Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5C	Foreign bill reporting mode
Length	1 binary	01	Number of bytes following, not including the CRC
Command	1 binary	00-01	00 – Add foreign bills to Voucher In meters 01 – Report true Voucher In meters
CRC	2 binary	0000-FFFF	16-bit CRC

The variable length gaming machine response to long poll 5C is detailed below in Table 18.5.5b.

**Table 18.5.5b**  
**Foreign Bill Reporting Mode Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5C	Foreign bill reporting mode
Length	1 binary	01	Number of bytes following, not including the CRC
Mode	1 binary	00-01	00 – Foreign bills added to Voucher In meters 01 – True Voucher In meters reported
CRC	2 binary	0000-FFFF	16-bit CRC

## 18.6 Enhanced Progressive Data Reporting

New in SAS 6.03

Enhanced Progressive Data Reporting provides support for reporting the current progressive broadcast amounts being received by the gaming machine for all progressive levels currently associated with one or more available win categories, as well as progressive accounting data. Gaming machines that support progressive data reporting must indicate such by setting Feature Codes 3, bit 7, to 1 in the A0 long poll response.

### 18.6.1 LP 5E: Send configured progressive controllers

The host may send a type R long poll with a 5E command code to determine which progressive controllers (if any) are currently configured for progressive levels on the gaming machine. The gaming machine response is detailed below in Table 18.6.1.

**Table 18.6.1**  
**Send Configured Progressive Controllers Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5E	Send configured progressive controllers
Length	1 binary	01-nn	Number of bytes following, not including CRC
Number of controllers	1 binary	00-nn	Number of controllers with levels currently associated with available win levels (00 if no progressives configured on gaming machine)
Controller type	1 binary	01-1F	Progressive controller type 01 = SAS protocol 02 = Mikohn protocol 03 = IPP protocol 04 = G2S protocol 05 = WAP protocol 06 = Standalone (gaming machine internal) 07 = Game-based (gaming machine internal) 08 = Other link protocol 09-1F = Reserved
Controller ID	2 BCD	0000-9999	Controller identifier (see below)
...	Variable	...	Additional controller type/controller ID data sets
CRC	2 binary	0000-FFFF	16-bit CRC

Only progressive controllers that are currently configured with levels associated with available win categories are reported. A win category is “available” if the game can be selected by the player. If there are no progressive levels currently linked on the gaming machine, long poll 5E returns with number of controllers set to zero, and does not include any controller type or controller ID data.

Each reported controller must have a unique combination of controller type plus controller ID. For example, if a gaming machine is configured with two Mikohn controllers, the controller type must be configured as Controller type 02 (Mikohn) for both controllers. The controller ID can be 0000 for one and 0001 for the other, or 0001 for one and 0002 for the other, or any other assignment at the discretion of the gaming machine. The assignment must not change as long as the controller is associated with available win categories.

G2S progressive devices should be identified using the G2S device ID. For SAS progressive levels associated with this SAS host, the Controller ID must be the SAS progressive group (converted to BCD). The chosen designations must remain constant for a given gaming machine configuration. The gaming machine must report exception 3C if any configuration is changed that could change the set of currently configured controllers or levels.

Standalone and game-based progressive controllers are both internal to the gaming machine. A Standalone progressive controller is typically a generic controller built into the gaming machine that can typically be used in place of a generic external controller. A game-based progressive controller is typically a controller integrated into a specific game theme. For a game-based controller, the controller ID must be set to the game number of the first (lowest numbered) enabled game that is using that controller.

### 18.6.2 LP 5F: Send progressive broadcast values

To aid in accounting and reconciliation for the various progressive controllers located on a casino floor, it is useful for the accounting system to be able to obtain the current progressive broadcast values for all progressive levels configured on a gaming machine.

The host may send the type S long poll with a 5F command code to request the current progressive broadcast values for the currently configured levels associated with the specified progressive controller. This variable length command is detailed in Table 18.6.2a.

**Table 18.6.2a**  
**Send Progressive Broadcast Values Command**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5F	Send progressive broadcast values
Length	1 binary	03	Number of bytes following, not including CRC
Controller type	1 binary	01-1F	Progressive controller type (from long poll 5E)
Controller ID	2 BCD	0000-9999	Controller identifier (from long poll 5E)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response is detailed below in Table 18.6.2b.

**Table 18.6.2b**  
**Send Progressive Broadcast Values Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5F	Send progressive broadcast values
Length	1 binary	01-nn	Number of bytes following, not including CRC
Controller type	1 binary	01-1F	Progressive controller type (see long poll 5E)
Controller ID	2 BCD	0000-9999	Controller identifier (see long poll 5E)
Number of levels	1 binary	00-nn	Number of progressive levels currently associated with win levels (00 if no configured progressives)
Level number	1 binary	01-20	Progressive level number
Broadcast value	5 BCD	000000000 0-999999999 9	Most recently received level amount in units of cents (0000000000 if no progressive information received in last 5 seconds)
...	Variable	...	Additional level number/broadcast value data sets
CRC	2 binary	0000-FFFF	16-bit CRC

If the requested controller type/controller ID is not valid, or no levels from the controller are currently associated with an available win category on the gaming machine, the gaming machine must respond with the number of levels set to zero and omit all following data fields. Otherwise, the gaming machine must report the number of configured levels, and report the level number and most recent broadcast value for each level. If the progressive broadcast value exceeds \$99,999,999.99 (such as might occur with a WAP progressive), or no data has been received for a level for the last 5 seconds (progressive link down), the gaming machine must set the value to all zeros.

### 18.6.3 LP 77: Send progressive accounting data

To aid in accounting and reconciliation for standalone and game-based progressive controllers, it is useful for the accounting system to be able to obtain the current progressive accounting data for all progressive levels configured on a gaming machine.

The host may send the type S long poll with a 77 command code to request the current progressive accounting data for the currently configured levels associated with the specified progressive controller. This variable length command is detailed in Table 18.6.3a.

Table 18.6.3a Send Progressive Accounting Data Command			
Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	77	Send progressive accounting data
Length	1 binary	03	Number of bytes following, not including CRC
Controller type	1 binary	01-1F	Progressive controller type (from long poll 5E)
Controller ID	2 BCD	0000-9999	Controller identifier (from long poll 5E)
Level number	1 binary	01-20	Progressive level number (from long poll 5F)
CRC	2 binary	0000-FFFF	16-bit CRC

The gaming machine response is detailed below in Table 18.6.3b.



**Table 18.6.3b**  
**Send Progressive Accounting Data Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	77	Send progressive accounting data
Length	1 binary	01-nn	Number of bytes following, not including CRC
Controller type	1 binary	06, 07	Progressive controller type
Controller ID	2 BCD	0000-9999	Controller identifier
Level number	1 binary	01-20	Progressive level number
Total progressive hits	4 BCD	00000000-99999999	Total number of times the progressive level has been hit
Total progressive paid	5 BCD	000000000-999999999	Total progressive paid meter for this level
Total contribution paid	5 BCD	000000000-999999999	Total contribution amounts (from wagers) paid, if available, in units of cents
Total manual adjustments paid	5 BCD	000000000-999999999	Total manual adjustments to progressive value paid, if available, in units of cents
Unpaid manual adjustments	5 BCD	000000000-999999999	Total manual adjustments to progressive value currently on the progressive meter, if available, in units of cents
Current escrow	5 BCD	000000000-999999999	Total outstanding liability not currently on the progressive meter, if available, in units of cents
CRC	2 binary	0000-FFFF	16-bit CRC

If the requested controller type/controller ID is not valid, or no levels from the controller are currently associated with an available win category on the gaming machine, the gaming machine must respond with the number of levels set to zero and omit all following data fields. Otherwise, the gaming machine must report the number of configured levels, and report the level number and most recent broadcast value for each level. For any data that is not available at the gaming machine, the response must be all zeros.

## 18.7 Non-SAS Progressive Hit Reporting

New in SAS 6.03

The following extension provides support for non-SAS progressive hit reporting. “Non-SAS progressives” includes all progressive controllers, internal or external to the gaming machine, including a different SAS host on a separate SAS channel. “Non-SAS” only means not *this* SAS host. For backwards compatibility, the gaming machine must provide the ability to enable/disable this extension. For maximum backwards compatibility, the configuration should initially default to disabled. Gaming

machines that have non-SAS progressive hit reporting enabled must indicate such by setting Feature Codes 3, bit 6, to 1 in the A0 long poll response.

Legacy SAS progressive support includes reporting exception 51 or 54 at the end of any game that includes one or more progressive hits. Long poll 84 reports the total progressive amount paid in the most recent game cycle that included progressive win. In addition, whenever a SAS progressive level controlled by a SAS host, exception 56 is issued to the progressive controller host at the time of the hit. Long poll 85 can be used by that host to report SAS progressive hits using a first-in/first-out queue.

To allow accounting systems to reconcile all progressive levels hit, the gaming machine must also be able to report individual progressive levels when the accounting host is not also the SAS progressive controller. When a non-SAS progressive level is hit, the gaming machine reports exception 59. The host can then use long poll 5D to read the progressive data. The progressive data includes the level hit and the total amount, plus any unpaid escrowed amounts at the time of the hit. For progressives with variable base amounts, the gaming machine can report the base amount portion and increment portion separately.

### 18.7.1 Reporting Non-SAS Progressive Hits

If a gaming machine has non-SAS progressive hit reporting enabled, the gaming machine must maintain an n-entry first-in/first-out queue of non-SAS progressive win data. This queue must be deep enough to hold the maximum number of non-SAS progressive levels that can be hit in any one game cycle. For a gaming machine that supports more than one SAS channel, progressives controlled by a SAS host connected to a different SAS channel are considered non-SAS progressives in this context. When a non-SAS progressive level is hit, the level and amount data, plus the amount of any unpaid escrow known by the gaming machine, is placed in this queue and exception 59 (Non-SAS progressive level hit) is reported. This exception is reported in addition to any exception 51 (handpay is pending) or exception 54 (progressive jackpot cashout device/credit paid). Exception 59 is not reported for SAS progressives, e.g. progressive levels controlled by this SAS host. Exception 59 is not reported if non-SAS progressive hit reporting is not enabled. If non-SAS progressive hit reporting is disabled while progressive data is in the queue, all data must be flushed from the queue.

A progressive award typically consists of a base amount, which is typically the amount defined in the payable for the award, and an increment portion, which is the additional amount provided by the progressive controller based on wagers. The base amount may also include an additional fixed amount provided by the controller that is not defined in the payable. The gaming machine should, whenever possible, report the base amount portion in the Base Amount field and the increment portion in the Amount field. This is particularly important when the base amount varies based on the bet level or other factors. However, with an external controller the gaming machine may not know the base amount. It is also possible for the progressive base amount to be zero. In these cases the gaming machine may report the entire progressive amount in the Amount field and set the Base Amount field to zero. In all cases, the sum of the Amount and Base Amount fields must equal the total progressive win amount added to the progressive payout meters.

### 18.7.2 LP 5D: Send Non-SAS Progressive Win Data

Upon receiving exception 59, the host must request the progressive win data records currently available by sending a type R long poll with a 5D command code. The gaming machine response is detailed below in Table 18.7.2.

**Table 18.7.2**  
**Send Non-SAS Progressive Win Data Response**

Field	Bytes	Value	Description
Address	1 binary	01-7F	Gaming machine address
Command	1 binary	5D	Send non-SAS progressive win data
Length	1 binary	01-B1	Number of bytes following, not including CRC
Number of levels	1 binary	00-08	Number of levels following (00 if queue empty)
Controller type	1 binary	01-1F	Progressive controller type (see long poll 5E)
Controller ID	2 BCD	0000-9999	Controller identifier (see long poll 5E)
Level	1 binary	00, 01-20	Progressive level
Amount (or increment)	5 BCD	000000000 0- 999999999 9	Win amount, or non-base portion of win when base amount field is non-zero, in units of cents
Base amount	5 BCD	000000000 0- 999999999 9	Base portion of this progressive win, if available, in units of cents
Escrow amount	5 BCD	000000000 0- 999999999 9	Any progressive increment that was escrowed at the time of the hit (not paid), if available, in units of cents
...	Variable	...	Additional controller/level/amount/base/escrow data sets
CRC	2 binary	0000-FFFF	16-bit CRC

The response to long poll 5D includes all data in the non-SAS progressive win queue, up to eight records maximum. For any data that is not available at the gaming machine, the response must be all zeros. When the response is acknowledged, the reported records are deleted from the queue. Note that a maximum of eight records can be reported at one time. If game design issues require a queue larger than eight elements, and additional records remain in the queue, the gaming machine reissues exception 59 and the process repeats. If no records are in the queue when the gaming machine receives long poll 5D, it must respond with a length of 01 and the number of levels set 00. No level or amount data will be included.

## 18.8 Gaming Machine Enforced Max Progressive Payback

New in SAS 6.03

The following provides support for jurisdictional requirements that a gaming machine include progressive contribution when enforcing a maximum theoretical payback percentage. To meet these regulations on game themes that use external progressive controllers, the gaming machine must be able to determine the total contribution percentage from all progressive sources. For a progressive game to be playable in a jurisdiction with these regulations, the Extended Progressive Broadcast command **MUST** be supported. The standard progressive broadcast commands should be used in all other jurisdictions. When a gaming machine is enforcing the maximum payback percentage, it must indicate such by setting Feature Codes 4, bit 0, to one in the A0 long poll response.

### 18.8.1 LP 7A: Extended Progressive Broadcast

The host may use the global broadcast long poll 7A to provide the progressive amount, base amount, and contribution rate for up to 16 progressive levels per poll. This variable length command is detailed in Table 18.8.1. Gaming machines do not respond to global broadcasts. Long polls 7A can also be sent to any single gaming machine as a type S poll. When received as a type S poll, the gaming machine ACKs or NACKs the message, as detailed in Table 7.4b.

**Table 18.8.1**  
**Extended Progressive Broadcast Command**

Field	Bytes	Value	Description
Address	1 binary	00-7F	Global broadcast or gaming machine address
Command	1 binary	7A	Extended progressive broadcast
Length	1 binary	0F-E1	Number of bytes following, not including the CRC
Group	1 binary	01-FF	Group ID for this broadcast
Level	1 binary	01-20	Progressive level
Amount	5 BCD	000000000 0- 999999999 9	Progressive amount in units of cents
Base	5 BCD	000000000 0- 999999999 9	Base/reset amount in units of cents
Contribution	3 BCD	000000- 999999	Contribution rate specified expressed as whole numbers with 4 (four) implied decimal places, for example, 015000 represents 1.5000%.
...	Variable	...	Optional additional level/amount/base/contribution data
CRC	2 binary	0000-FFFF	16-bit CRC

It is up to the gaming machine whether to enforce the requirements for maximum theoretical payback percentage. When enforcing these requirements, the gaming machine must not accept progressive data from long polls 80 or 86. If the gaming machine is not receiving the necessary information for any progressive level, it MUST disable the associated game the same as if it was not receiving any progressive information, and issue exception 53, No progressive information has been received for 5 seconds.

When the base amount is set to a value other than the associated value in the game's payable, this will cause variations to the theoretical payback percentage that may be difficult to calculate. A gaming machine may reject any base amount that is not equal to the payable amount, disable the associated progressive game, and issue exception 53 as described above. The gaming machine should display an informative error message to assist an operator in diagnosing the configuration error.

If the gaming machine determines that the maximum theoretical payback percentage for a game is exceeded based on the base maximum payback percentage and the sum of all contributions for all levels associated with a game, the gaming machine must disable the associated progressive game and issue exception 53 as described above. The gaming machine should display an informative error message to assist an operator in diagnosing the configuration error.

## 18.9 Exc 5A: Jackpot Handpay Keyed Off to Machine Pay

## New in SAS 6.03

Exception 5A allows a gaming machine to report that a jackpot handpay has been keyed off to the credit meter, a voucher, or any other gaming machine pay method. Gaming machines that support this feature must indicate such by setting Feature Codes 2, bit 5, to 1 in the A0 long poll response.

SAS, another protocol such as G2S, or an internal gaming machine option may provide a process whereby a jackpot handpay may be keyed off to the credit meter, a voucher, or any other gaming machine pay method. In this case, the win amount is metered as a machine pay (e.g. Coin Out) instead of an attendant pay (e.g. Jackpot). In this case, the gaming machine must still issue exception 52 as with a standard handpay. In addition, the gaming machine may also issue exception 5A, Jackpot handpay keyed off to machine pay. Exception 5A must be issued whether the key-off to machine pay is performed by an attendant at the gaming machine, by a command from a host, or any other process. Exception 5A is not a priority exception, and must be inserted in the exception queue with other non-priority exceptions.



## APPENDIX A GENERAL POLL EXCEPTION CODES

**Note:** Gaming machines must support all exception codes that are applicable to that gaming machine's hardware configuration.

Type P = Priority/Process, Q = Queued (See Section 2.2.1)

**Table A-1**  
**General Exception Codes**

Code	Type	Page	Description
00		2-1, 12-7	No activity
11	Q		Slot door was opened
12	Q		Slot door was closed
13	Q		Drop door was opened
14	Q		Drop door was closed
15	Q		Card cage was opened
16	Q		Card cage was closed
17	Q		AC power was applied to gaming machine
18	Q		AC power was lost from gaming machine
19	Q		Cashbox door was opened
1A	Q		Cashbox door was closed
1B	Q		Cashbox was removed
1C	Q		Cashbox was installed
1D	Q		Belly door was opened
1E	Q		Belly door was closed
1F		12-7, 13-1	No activity and waiting for player input (obsolete)
20	Q		General tilt (Use this tilt when other exception tilt codes do not apply or when the tilt condition cannot be determined.)
21	Q		Coin in tilt
22	Q		Coin out tilt
23	Q		Hopper empty detected
24	Q		Extra coin paid
25	Q		Diverter malfunction (controls coins to drop or hopper)
27	Q		Cashbox full detected
28	Q		Bill jam
29	Q		Bill acceptor hardware failure
2A	Q		Reverse bill detected
2B	Q		Bill rejected
2C	Q		Counterfeit bill detected
2D	Q		Reverse coin in detected
2E	Q		Cashbox near full detected
2F	Q		Bill acceptor version changed
31	Q		Critical data memory error (some data recovered from secondary backup)
32	Q		Critical data memory error (no data recovered from secondary backup)

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH



**Table A-1 (cont.)  
General Exception Codes**

Code	Type	Page	Description
33	Q		Critical data memory error (bad device)
34	Q		Secondary critical data memory error (data error)
35	Q		Secondary critical data memory error (bad device)
36	Q		Program memory error (different checksum – version changed)
37	Q		Program memory error (bad checksum compare)
38	Q		Program memory error (checksum – version changed) (obsolete)
39	Q		Program memory error (bad checksum compare) (obsolete)
3A	Q		Memory error reset by operator
3B	Q		Low backup battery detected
3C	Q		Configuration option changed (This is sent whenever the operator changes any configuration option that could directly or indirectly affect data reported to a SAS host. This includes, but is not limited to, denomination, gaming machine address, or any option that affects the response to long polls 1F, 53, 54, 56, A0, B2, B3, B4, or B5.)
3D	Q/P	15-1	A cash out ticket has been printed
3E	Q/P	15-1	A handpay has been validated
3F	P	15-1	Validation ID not configured
40	Q		Reel Tilt (Which reel is not specified.)
41	Q		Reel 1 tilt
42	Q		Reel 2 tilt
43	Q		Reel 3 tilt
44	Q		Reel 4 tilt
45	Q		Reel 5 tilt
46	Q		Reel mechanism disconnected
47	Q	7-17	\$1.00 bill accepted (non-RTE only)
48	Q	7-17	\$5.00 bill accepted (non-RTE only)
49	Q	7-17	\$10.00 bill accepted (non-RTE only)
4A	Q	7-17	\$20.00 bill accepted (non-RTE only)
4B	Q	7-17	\$50.00 bill accepted (non-RTE only)
4C	Q	7-17	\$100.00 bill accepted (non-RTE only)
4D	Q	7-17	\$2.00 bill accepted (non-RTE only)
4E	Q	7-17	\$500.00 bill accepted (non-RTE only)
4F	Q	7-17, 12-2	Bill accepted (In non-RTE mode, use this exception for all bills without a specific exception. In RTE mode, use for all bill denominations.)
50	Q	7-17	\$200.00 bill accepted (non-RTE only)
51	Q/P	7-14	Handpay is pending (Progressive, non-progressive or cancelled credits)
52	Q/P	7-14	Handpay was reset (Jackpot reset switch activated)
53	Q	10-2	No progressive information has been received for 5 seconds
54	Q	10-2	Progressive win (cashout device/credit paid)
55	Q/P		Player has cancelled the handpay request
56	P	10-3	SAS progressive level hit
57	P	15-11	System validation request
59	P	18-18	Non-SAS progressive level hit
5A	Q	18-20	Jackpot handpay keyed off to machine pay
5F	Q	18-6	Tip awarded

60

Q

Printer communication error

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH

**Table A-1 (cont.)**  
**General Exception Codes**

Code	Type	Page	Description
61	Q		Printer paper out error
66	Q	8-21	Cash out button pressed
67	P	15-17	Ticket has been inserted
68	P	15-18	Ticket transfer complete
69	P	8-8	AFT transfer complete
6A	P	8-20	AFT request for host cashout
6B	P	8-20	AFT request for host to cash out win
6C	P	8-2	AFT request to register
6D	P	8-2	AFT registration acknowledged
6E	Q	8-2	AFT registration cancelled
6F	P	8-5	Game locked
70	P	2-1	Exception buffer overflow
71	Q		Change lamp on
72	Q		Change lamp off
74	Q		Printer paper low
75	Q		Printer power off
76	Q		Printer power on
77	Q		Replace printer ribbon
78	Q		Printer carriage jammed
79	Q		Coin in lockout malfunction (coin accepted while coin mech disabled)
7A	Q		Gaming machine soft (lifetime-to-date) meters reset to zero (all or some)
7B	Q		Bill validator (period) totals have been reset by an attendant/operator
7C	Q	12-2, <b>Error!</b> <b>Bookm</b> <b>ark not</b> <b>define</b> <b>d.</b>	A legacy bonus pay awarded and/or a multiplied jackpot occurred
7E	Q	12-3	Game has started
7F	Q	12-4	Game has ended
80	Q		Hopper full detected
81	Q		Hopper level low detected
82	Q		Display meters or attendant menu has been entered
83	Q		Display meters or attendant menu has been exited
84	Q		Self test or operator menu has been entered
85	Q		Self test or operator menu has been exited
86	Q		Gaming machine is out of service (by attendant)
87	Q		Player has requested draw cards (only send when in RTE mode)
88	Q	12-5	Reel N has stopped (only send when in RTE mode)
89	Q		Coin/credit wagered (only send when in RTE mode, and only send if the configured max bet is 10 or less)
8A	Q	12-5	Game recall entry has been displayed
8B	Q	12-5	Card held/not held (only send when in RTE mode)
8C	Q	12-6	Game selected
8E	Q	17-1	Component list changed

8F	P	17-6	Authentication complete
98	Q		Power off card cage access
99	Q		Power off slot door access

**Table A-1 (cont.)**  
**General Exception Codes**

<b>Code</b>	<b>Type</b>	<b>Page</b>	<b>Description</b>
9A	Q		Power off cashbox door access
9B	Q		Power off drop door access
9E	Q	12-6	Session Start
9F	Q	12-7	Session End
A0	P	18-1	Meter change pending
A1	P	18-1	Meter change cancelled
A2	Q	18-1	Enabled games/denoms changed



## APPENDIX B LONG POLL COMMANDS

**Note:** The Type field specifies the long poll type, as defined in Section 2.

The Page field references where each long poll is described in the document.

Gaming machines must support all long polls that are applicable to that gaming machine's hardware configuration.

**Table B-1**  
**Long Poll Commands**

Poll	Type	Page	Description	Response
01	S	7-5	Shutdown (lock out play)	ACK or NACK
02	S	7-5	Startup (enable play)	ACK or NACK
03	S	7-4	Sound off (all sounds disabled)	ACK or NACK
04	S	7-4	Sound on (all sounds enabled)	ACK or NACK
05	S	7-4	Reel spin or game play sounds disabled	ACK or NACK
06	S	7-4	Enable bill acceptor	ACK or NACK
07	S	7-4	Disable bill acceptor	ACK or NACK
08	S	7-5	Configure bill denominations	ACK or NACK
09	M	7-7	Enable/disable game n	ACK or NACK
0A	S	7-5	Enter maintenance mode	ACK or NACK
0B	S	7-5	Exit maintenance mode	ACK or NACK
0E	S	12-1	Enable/disable real time event reporting	ACK or NACK
0F	R	7-1	Send meters 10 through 15	4-byte BCD total cancelled credits 4-byte BCD total coin in meter 4-byte BCD total coin out meter 4-byte BCD total drop meter 4-byte BCD total jackpot meter. 4-byte BCD games played meter
10	R	7-1	Send total cancelled credits meter	4-byte BCD total cancelled credits meter
11	R	7-1	Send total coin in meter	4-byte BCD total coin in meter
12	R	7-1	Send total coin out meter	4-byte BCD total coin out meter
13	R	7-1	Send total drop meter	4-byte BCD total drop meter
14	R	7-1	Send total jackpot meter	4-byte BCD total jackpot meter
15	R	7-1	Send games played meter	4-byte BCD games played meter
16	R	7-1	Send games won meter	4-byte BCD games won meter
17	R	7-1	Send games lost meter	4-byte BCD games lost meter
18	R	7-11	Send games since last power up and games since last slot door closure meters	2-byte BCD games since power up 2-byte BCD games since last time slot door was closed

**Table B-1 (cont.)  
Long Poll Commands**

Poll	Type	Page	Description	Response
19	R	7-1	Send meters 11 through 15	4-byte BCD total coin in meter 4-byte BCD total coin out meter 4-byte BCD total drop meter 4-byte BCD total jackpot meter 4-byte BCD games played meter
1A	R	7-1	Send current credits	4-byte BCD credit meter
1B	R	7-13	Send handpay information	1-byte binary progressive group 1-byte binary level 5-byte BCD amount 2-byte BCD partial pay amount. 1-byte binary Reset ID 10 unused bytes (zero padded)
1C	R	7-1	Send meters	4-byte BCD total coin in meter 4-byte BCD total coin out meter 4-byte BCD total drop meter 4-byte BCD total jackpot meter 4-byte BCD games played meter 4-byte BCD games won meter 4-byte BCD slot door open meter 4-byte BCD power reset meter
1E	R	7-1	Send total bill meters (# of bills)	4-byte BCD meter for \$1.00 4-byte BCD meter for \$5.00 4-byte BCD meter for \$10.00 4-byte BCD meter for \$20.00 4-byte BCD meter for \$50.00 4-byte BCD meter for \$100.00
1F	R	7-16	Send gaming machine ID & information	2-byte ASCII game ID 3-byte ASCII additional ID 1-byte binary denomination 1-byte binary max bet 1-byte binary progressive group 2-byte binary game options 6-byte ASCII payable ID 4-byte ASCII base percentage
20	R	7-1	Send total dollar value of bills meter	4-byte BCD bill meter in dollars
21	S	6-1	ROM signature verification	2-byte binary ROM signature
2A	R	7-1	Send true coin in	4-byte BCD meter in # of coins/tokens
2B	R	7-1	Send true coin out	4-byte BCD meter in # of coins/tokens
2C	R	7-1	Send current hopper level	4-byte BCD meter in # of coins/tokens
2D	M	7-7	Send total hand paid cancelled credits	2-byte BCD game number 4-byte BCD meter in SAS accounting denom units
2E	S	13-5	Delay game	ACK or NACK



**Table B-1 (cont.)  
Long Poll Commands**

Poll	Type	Page	Description	Response
2F	M	7-3	Send selected meters for game n	1-byte binary length 2-byte BCD game number n-bytes per meter... 1-byte binary meter code n-byte BCD meter value additional code/value pairs as necessary
31	R	7-1	Send \$1.00 bills in meter	4-byte BCD meter in # of bills
32	R	7-1	Send \$2.00 bills in meter	4-byte BCD meter in # of bills
33	R	7-1	Send \$5.00 bills in meter	4-byte BCD meter in # of bills
34	R	7-1	Send \$10.00 bills in meter	4-byte BCD meter in # of bills
35	R	7-1	Send \$20.00 bills in meter	4-byte BCD meter in # of bills
36	R	7-1	Send \$50.00 bills in meter.	4-byte BCD meter in # of bills
37	R	7-1	Send \$100.00 bills in meter	4-byte BCD meter in # of bills
38	R	7-1	Send \$500.00 bills in meter	4-byte BCD meter in # of bills
39	R	7-1	Send \$1,000.00 bills in meter	4-byte BCD meter in # of bills
3A	R	7-1	Send \$200.00 bills in meter	4-byte BCD meter in # of bills
3B	R	7-1	Send \$25.00 bills in meter	4-byte BCD meter in # of bills
3C	R	7-1	Send \$2,000.00 bills in meter	4-byte BCD meter in # of bills
3D	R	15-10	Send cash out ticket information	4-byte BCD ticket number 5-byte BCD amount in cents
3E	R	7-1	Send \$2,500.00 bills in meter.	4-byte BCD meter in # of bills
3F	R	7-1	Send \$5,000.00 bills in meter	4-byte BCD meter in # of bills
40	R	7-1	Send \$10,000.00 bills in meter	4-byte BCD meter in # of bills
41	R	7-1	Send \$20,000.00 bills in meter	4-byte BCD meter in # of bills
42	R	7-1	Send \$25,000.00 bills in meter	4-byte BCD meter in # of bills
43	R	7-1	Send \$50,000.00 bills in meter	4-byte BCD meter in # of bills
44	R	7-1	Send \$100,000.00 bills in meter	4-byte BCD meter in # of bills
45	R	7-1	Send \$250.00 bills in meter	4-byte BCD meter in # of bills
46	R	7-1	Send credit amount of all bills accepted	4-byte BCD meter in SAS accounting denom units
47	R	7-1	Send coin amount accepted from an external coin acceptor	4-byte BCD meter in SAS accounting denom units
48	R	7-17	Send last accepted bill information	1-byte BCD country code 1-byte BCD bill denomination 4-byte BCD meter for accepted bills of this type
49	R	7-1	Send number of bills currently in the stacker	4-byte BCD meter in # of bills
4A	R	7-1	Send total credit amount of all bills currently in the stacker	4-byte BCD meter in SAS accounting denom units
4C	S	15-10	Set secure enhanced validation ID	See Section 15.6
4D	S	15-14	Send enhanced validation information	See Section 15.10

**Table B-1 (cont.)**  
**Long Poll Commands**

Poll	Type	Page	Description	Response
4F	R	7-23	Send current hopper status	1-byte binary length 1-byte binary status 1-byte binary % full 4-byte BCD level
50	S	15-23	Send validation meters	1-byte binary validation type 4-byte BCD total validations 5-byte BCD cumulative amount
51	R	7-8	Send total number of games implemented	2-byte BCD number of games
52	M	7-9	Send game n meters	2-byte BCD game number 4-byte BCD total coin in meter 4-byte BCD total coin out meter 4-byte BCD total jackpot meter 4-byte BCD games played meter
53	M	7-9	Send game n configuration	2-byte BCD game number 2-byte ASCII game ID 3-byte ASCII additional ID 1-byte binary denomination 1-byte binary max bet 1-byte binary progressive group 2-byte binary game options 6-byte ASCII payable ID 4-byte ASCII base percentage
54	R	7-21	Send SAS version ID and gaming machine serial number	1-byte binary data length 3-byte ASCII SAS version number X byte ASCII serial number
55	R	7-11	Send selected game number	2-byte BCD selected game number
56	R	7-11	Send enabled game numbers	1-byte binary data length 1-byte binary number of games X-byte BCD enabled game numbers
57	R	15-11	Send pending cashout information	1-byte binary cashout type 5-byte BCD cashout amount
58	S	15-12	Receive validation number	1-byte binary status
59	R	18-9	Send enabled currency codes	1-byte binary length 3-byte ASCII ISO 4217 country code X-byte ASCII additional country codes
5A	S	18-10	Send supported bills	1-byte binary length 3-byte ASCII ISO 4217 country code 5-byte BCD bill denomination X-byte BCD additional denominations

**Table B-1 (cont.)  
Long Poll Commands**

<b>Poll</b>	<b>Type</b>	<b>Page</b>	<b>Description</b>	<b>Response</b>
5B	S	18-11	Send bill meters	1-byte binary length 3-byte ASCII ISO 4217 country code 5-byte BCD bill denomination 1-byte binary amount meter length X-byte BCD amount meter 1-byte binary count meter length X-byte BCD count meter
5C	S	18-12	Foreign bill reporting mode	1-byte binary length 1-byte binary mode
5D	R	18-18	Send Non-SAS progressive win data	1-byte binary length 1-byte binary number of levels X-byte level data
5E	R	18-13	Send configured progressive controllers	1-byte binary length 1-byte binary number of controllers X-byte controller data
5F	S	18-14	Send progressive broadcast values	1-byte binary length 1-byte binary controller type 1-byte BCD controller ID 1-byte binary number of levels X-byte level data
6E	S	17-1	Send authentication info	See Section 17.1
6F	M	7-25	Send extended meters for game n	1-byte binary length 2-byte BCD game number x bytes per meter, up to 12 meters
70	R	15-17	Send ticket validation data	1-byte binary length 1-byte binary ticket status 5-byte BCD ticket amount 1-byte binary parsing code x-byte validation data
71	S	15-18	Redeem ticket	See Section 15.12
72	S	8-8	AFT transfer funds	See Section 8.3
73	S	8-2	AFT register gaming machine	See Section 8.1
74	S	8-5	AFT game lock and status request	See Section 8.2
75	S/G	8-23	Set AFT receipt data	See Section 8.11
76	S	8-28	Set custom AFT ticket data	See Section 8.12
77	R	18-16	Send progressive accounting data	1-byte binary length 1-byte binary controller type 2-byte BCD controller identifier 1-byte binary level number X-byte controller meter data
7A	S/G	18-19	Extended progressive broadcast	N/A
7B	S/G	15-3	Extended validation status	See Section 15.2
7C	S/G	15-6	Set extended ticket data	See Section 15.3
7D	S/G	15-8	Set ticket data	See Section 15.4
7E	R	7-23	Send current date and time	4-byte BCD date 3-byte BCD time

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH

**Table B-1 (cont.)  
Long Poll Commands**

Poll	Type	Page	Description	Response
7F	S/G	7-22	Receive date and time	N/A
80	S/G	10-1	Receive progressive amount	N/A
83	M	10-5	Send cumulative progressive wins	2-byte BCD game number 4-byte BCD progressive wins in SAS accounting denom units
84	R	10-2	Send progressive win amount	1-byte binary group 1-byte binary level 5-byte BCD amount
85	R	10-3	Send SAS progressive win amount	1-byte binary group 1-byte binary level 5-byte BCD amount
86	S/G	10-1	Receive multiple progressive levels	N/A
87	R	10-4	Send multiple SAS progressive win amounts	1-byte binary length 1-byte binary group 1-byte binary number of levels x-byte level/amount data
8A	S	13-2	Initiate a legacy bonus pay	ACK or NACK
8B	S	13-3	Initiate multiplied jackpot mode (obsolete)	ACK or NACK
8C	M	11-1	Enter/exit tournament mode	ACK or NACK
8E	R	7-18	Send card information	1-byte binary hand type 5-byte binary cards
8F	R	7-19	Send physical reel stop information	9-byte binary physical reel stops
90	R	13-4	Send legacy bonus win amount	1-byte binary jackpot multiplier 4-byte BCD jackpot mult. amount 1-byte bonus tax status 4-byte BCD bonus award amount
94	S	7-15	Remote handpay reset	1-byte binary reset code
95	M	11-1	Send tournament games played	2-byte BCD game number 4-byte BCD meter in # of games
96	M	11-1	Send tournament games won	2-byte BCD game number 4-byte BCD meter in # of games
97	M	11-1	Send tournament credits wagered	2-byte BCD game number 4-byte BCD meter in SAS accounting denom units
98	M	11-1	Send tournament credits won	2-byte BCD game number 4-byte BCD meter in SAS accounting denom units
99	M	11-1	Send meters 95 through 98	2-byte BCD game number 4-byte BCD meter in # of games 4-byte BCD meter in # of games 4-byte BCD meter in # of credits 4-byte BCD meter in SAS accounting denom units

**Table B-1 (cont.)**  
**Long Poll Commands**

Poll	Type	Page	Description	Response
9A	M	13-5	Send legacy bonus meters	2-byte BCD game number 4-byte BCD deductible meter 4-byte BCD non-deductible meter 4-byte BCD wager match
A0	M	7-20	Send enabled features	2-byte BCD game number 2-byte binary features 4 bytes reserved
A4	M	7-22	Send cash out limit	2-byte BCD game number 2-byte BCD cash out limit in SAS accounting denom units
A8	S	14-1	Enable jackpot handpay reset method	1-byte binary ACK code
AA	S	7-25	Enable/disable game auto rebet	ACK or NACK
AF	M	7-25	Send extended meters for game n (alternate)	1-byte binary length 2-byte BCD game number x bytes per meter, up to 12 meters
B0	S	16-1	Multi-denom preamble	See Section 16.1
B1	R	16-6	Send current player denomination	1-byte binary denomination
B2	R	16-6	Send enabled player denominations	1-byte binary length 1-byte binary number of denoms X-byte binary denominations
B3	R	7-27	Send token denomination	1-byte binary token denomination
B4	M	7-30	Send wager category information	See Section 7.24
B5	M	7-27	Send extended game n information	See Section 7.23
B6	S	18-1	Meter collect status	1-byte binary length 1-byte binary status
B7	S	18-3	Set machine numbers	1-byte binary length 1-byte binary control flags 4-byte binary asset number 1-byte binary floor location length X-byte ASCII floor location
EB			Reserved	
FF	S	12-1	Event response to long poll	See Section 12.5

## APPENDIX C DATA TABLES

**Table C-1**  
**Game Identification Codes**

Game ID	Description
A1	Atlas Gaming Technologies
A2	IGT - Ascent
AA	Avantime Amusement Technology
AC	American Coin
AD	Advanced Gaming Solutions
AE	ACE Casino & Games, Inc.
AG	Ainsworth Game Technology Ltd.
AI	All In Technologies
AL	Alfastreet
AM	American Gaming Systems
AN	Advanced Gaming Pty Ltd
AO	Astro Corporation
AP	North American Gaming, Inc.
AR	Arachnid, Inc.
AS	Acres
AT	Aristocrat
AV	IGT - AVP
AX	AC Slots
B7	Bally Video 7000
B9	Bally Gaming
BA	Belatra Co., Ltd
BB	Bluberi Gaming Technologies, Inc.
BC	Barcrest
BD	Borden Technology Corporation
BE	IGT – Player’s Edge-Plus blackjack
BG	Bingo Gaming Technologies, Inc.
BI	Barcrest S2000i
BJ	IGT – Fortune II blackjack
BL	Blue Alpha Management Consultants Ltd
BS	Bally ProSlot
BV	Bally ProVideo
C2	C2 Gaming
CA	Cummins Allison Corp.
CB	Cybercash Technology International Ltd
CC	Cron Corporation
CE	Pace-O-Matic Cutting EDGE
CG	Colombia Games – KIWI
CH	Vista+ LLC
CJ	Cadillac Jack
CM	Coin Master UK
CP	Cory Investments, Ltd.
CR	Carat Technology d.o.o.
CS	CDS Reel Slot

**Table C-1 (cont.)**  
**Game Identification Codes**

Game ID	Description
CT	Cyberview Technology, Ltd.
CV	CDS Video
CY	Technology Exclusive (Cyberdyne)
DA	Data Art, Inc.
DD	DigiDeal Corporation
DE	Deico Electronic Co.
DG	Diamond Game Enterprises
DS	Impact Innovations, Inc.
DZ	Chicago Gaming Company
E!	Eclipse Gaming Systems, LLC
EB	ELBET d.o.o.
EG	Eurogames Technology
EI	Elixer International
EL	Electroncek
EM	Elaut NV
ES	Epic Software LLC
FB	FBM Brasil LTDA
FN	FortuNet, Inc
FP	THT d.o.o. (Fairplay)
FW	Fleetwood
G2	Grover Gaming
GA	Games of Chance
GB	Proizbira doo (Gambree)
GC	Gamecraft, Inc.
GG	Global Gaming Industries Pty. Ltd.
GI	Gamey Industries s.r.o.
GK	IGT – Game King
GO	C and M Technics AG (Golden Games)
GP	Scientific Games Inc.
GR	Gold Club d.o.o.
GS	International Games System
GU	adp Gauselmann GmbH
GV	Grand Vision Gaming, LLC
GW	Gateway Gaming, LLC
HY	HYDAKO
IC	Intui Code Gaming Corporation
ID	ID Interactive, LLC
IG	IGCA
IM	Impera GmbH
IN	Inspired Gaming
IO	Icon Gaming
IP	Igrosoft LLC
IS	Integrated Systems Design
IT	Incredible Technologies, Inc.



**Table C-1 (cont.)  
Game Identification Codes**

<b>Game ID</b>	<b>Description</b>
IY	Infinitygames Production GMBH
JG	JUNO GAMING Co., Ltd
JI	Jicun International Co, Ltd
JP	JPM International
JS	Joint Venture – S+
JT	Jumbo Technology Co, LTD
K+	IGT – Player’s Edge-Plus keno
KE	IGT – Player’s Edge-Plus keno
KG	Konami Gaming
KN	IGT – Fortune II keno
KT	EIC Techno.
LG	LT Game Ltd.
LP	Lightning Poker
LT	Leisure Time
LV	Las Vegas Gaming, Inc.
M+	IGT – M+ Slots
MG	IGT – Player’s Edge-Plus multi-game
MI	Millennium Gaming, Inc.
MJ	MAJSA
MM	Multimedia Games, Inc.
MP	IGT Multi-Player
N1	Novomatic Coolfire 1+
N2	Novomatic Coolfire 2
N5	Novomatic Coolfire 5
NA	NOVOMATIC ASIC
NC	Novomatic Crown Technologies
ND	Nova Desitec
NG	NOVOMATIC Special
NI	Novomatic Indigo
NL	New Gaming Systems
NM	NOVOMATIC Coolfire MasterSlot
NO	Nova Gaming, LLC
NP	Novomatic FireFox Pro
NS	Novomatic SWA
NV	NOVOMATIC Coolfire Video
OG	Orion Gaming
OI	Octavian International
OL	Outer Limits Gaming
P+	IGT – Player’s Edge-Plus poker
P1	Primero Games, LLC
PA	Pacific Gaming
PC	Project Coin Machines
PE	PaokaiElectronic Enterprise Co., Ltd.
PG	Premier Technology
PK	IGT – Fortune II or Player’s-Edge poker
PL	Pace-O-Matic Platinum Plus

**Table C-1 (cont.)  
Game Identification Codes**

<b>Game ID</b>	<b>Description</b>
PM	Mikohn (P&M Coin)
PO	PokerTek, Inc.
PP	IGT – Player’s Edge-Plus poker
PS	IGT – Player’s Edge-Plus slot
PT	Playsystem Technologies
PX	Panter Gaming d.o.o.
RB	Royce & Bach D.O.O.
RD	Regulus Corp. Inc.
RG	Rocket Gaming
RM	RGB Sdn Bhd
RP	Club Royal Poker
RS	IGT – S-Slot
RT	Reel Time Gaming Pty Ltd
RU	Rubicon Technologies d.o.o
S+	IGT – S+ 1988 release S-Slot
S0	SMARTGAMES S/W H/W SYSTEMS HOLDINGS LTD
S1	SMARTGAMES POKER
S2	SMARTGAMES SLOTS
S3	SMARTGAMES KENO
S4	SMARTGAMES ROULETTE
S5	SMARTGAMES BINGO
S6	SMARTGAMES LOTTO
S7	SMARTGAMES BET
S8	Sigma – Super 8 video slot
SA	Sammy Corporation
SB	Subsino America
SC	Sigma Special
SD	Sierra Design Group
SE	Select Electronic Devices
SF	Shuffle Master
SG	IGT – Vision Slot
SH	Sharp Image Gaming
SI	Silicon Gaming
SJ	Southern Gaming Services Pty. Ltd
SL	Sleic
SM	Sigma Mechanical
SO	Spielo
SP	IGT – Spectrum display
SR	Slot Constructor
SS	IGT – S-Plus slot
ST	Stargames
SU	Summit Amusement, LTD
SV	Sigma Video

**Table C-1 (cont.)  
Game Identification Codes**

<b>Game ID</b>	<b>Description</b>
SW	Show Machine México
T+	IGT – Player’s Edge-Plus 21
TB	Tecnobison S.A.
TC	Fazi, TripleCrown Roulette
TE	Techlink Entertainment
TG	IGT Table Game
TM	TableMAX
TP	Town Pump
VT	Video Gaming Technologies, Inc.
TS	Astra Games Ltd. (Top Spot)
U1	U1 Gaming
UD	Unidesa
UL	UNICUM Limited
UN	Aruze Corp.
VD	VDK2 Mexico
VG	Vista Gaming
VM	IGT – Vision Multi-game
WC	IGT – Winner’s Choice
WG	Worldwide Gaming Systems Corporation
WK	Weike (s) Pte Ltd
WO	American Music Company
XB	Playtrix Research Inc.
Z4	Gaming Arts, LLC
ZI	Zitro Factory S.L.

**Please contact IGT for allocation of unique Game Identification Codes**

---

**Table C-2**  
**Game Option Configurations: Vendor Dependent**

---

---

**Note:** *IGT no longer documents the Game Options field in long polls 1F and 53 in the SAS protocol. Gaming machine manufacturers are free to define these bits as desired so long as the definitions are consistent across all gaming machines reporting the same Game ID.*

---

---

**Table C-3**  
**Paytable/Reel Strip IDs: Vendor Dependent**

---

---

**Note:** *IGT no longer documents payable/reel strip IDs in the SAS protocol. Gaming machine manufacturers are free to define Paytable ID and Additional ID fields in long polls 1F and 53 as desired so long as different paytables are uniquely identified across all gaming machines reporting the same Game ID.*

---

**Table C-4**  
**Denomination Table**

<b>Code</b> (binary )	<b>Cents</b> (see note below)	<b>U.S. Denomination</b>
00	none	none
01	1	\$0.01
02	5	\$0.05
03	10	\$0.10
04	25	\$0.25
05	50	\$0.50
06	100	\$1.00
07	500	\$5.00
08	1,000	\$10.00
09	2,000	\$20.00
0A	10,000	\$100.00
0B	20	\$0.20
0C	200	\$2.00
0D	250	\$2.50
0E	2,500	\$25.00
0F	5,000	\$50.00
10	20,000	\$200.00
11	25,000	\$250.00
12	50,000	\$500.00
13	100,000	\$1000.00
14	200,000	\$2000.00
15	250,000	\$2500.00
16	500,000	\$5000.00
17	2	\$0.02
18	3	\$0.03
19	15	\$0.15
1A	40	\$0.40
1B	1/2	\$0.005
1C	1/4	\$0.0025
1D	1/5	\$0.002
1E	1/10	\$0.001
1F	1/20	\$0.0005
20-3F	Reserved	

**Note:** For currencies other than USD, a "cent" is generally equivalent to the minor units of the reporting currency, as defined by the ISO-4217 standard.

**Table C-5**  
**Bill Acceptor Country Code Values**

<b>Code (BCD)</b>	<b>Country</b>	<b>Code (BCD)</b>	<b>Country</b>
00	Unknown country code	21	Italy
01	Argentina	22	Jersey
02	Australia	23	Luxembourg
03	Austria	24	Malta
04	Belgium	25	Mexico
05	Brazil	26	Morocco
06	Bulgaria	27	Norway
07	Canada	28	Poland
08	Columbia	29	Portugal
09	Cyprus	30	Romania
10	Czechoslovakia	31	Russia
11	Denmark	32	Spain
12	Finland	33	South Africa
13	France	34	Sweden
14	Germany	35	Switzerland
15	Great Britain	36	Turkey
16	Gibraltar	37	United States
17	Greece	38	Holland
18	Guernsey	39	Euro
19	Hungary	40-47	Reserved
20	Ireland		

**Table C-6**  
**Bill Denomination Code Values**

<b>Code (BCD)</b>	<b>US Denomination (see note below)</b>
00	\$1
01	\$2
02	\$5
03	\$10
04	\$20
05	\$25
06	\$50
07	\$100
08	\$200
09	\$250
10	\$500
11	\$1,000
12	\$2,000
13	\$2,500
14	\$5,000
15	\$10,000
16	\$20,000
17	\$25,000
18	\$50,000
19	\$100,000
20	\$200,000
21	\$250,000
22	\$500,000
23	\$1,000,000
24-31	Reserved

**Note:** For currencies other than USD, the value of a "\$1 bill" is generally equivalent to 100 of the minor units of the reporting currency, as defined by the ISO-4217 standard. However, legacy bill reporting issues may take precedence over this rule. Please consult system providers operating in the local jurisdiction for guidance.



**Note:** Meter codes 0000 through 000C and 0015 through 007F include all relevant activity on the gaming machine. Meters 000D through 0014 are SAS validation-specific meters and are maintained here for backwards compatibility only.

Min Size is the minimum allowable size for each meter in BCD bytes (one BCD byte equals two digits) for reporting in long polls 6F and AF. Long poll 2F uses Min Size as the number of bytes reported (even if the actual meter is larger).

**Table C-7  
Meter Code Values**

Code (binary)	Meter	Min Size	Recommended Support		
			Gaming Machine	Per Game	Per Denom
0000	Total coin in credits	4 BCD	X	X	X
0001	Total coin out credits	4 BCD	X	X	X
0002	Total jackpot credits	4 BCD	X	X	X
0003	Total hand paid cancelled credits	4 BCD	X		
0004	Total cancelled credits	4 BCD	X		
0005	Games played	4 BCD	X	X	X
0006	Games won	4 BCD	X	X	X
0007	Games lost	4 BCD	X	X	X
0008	Total credits from coin acceptor	4 BCD	X		
0009	Total credits paid from hopper	4 BCD	X		
000A	Total credits from coins to drop	4 BCD	X		
000B	Total credits from bills accepted	4 BCD	X		
000C	Current credits	4 BCD	X		
000D	Total SAS cashable ticket in, including nonrestricted tickets (cents) [same as meters 0080 + 0084]	5 BCD	X		
000E	Total SAS cashable ticket out, including debit tickets (cents) [same as meters 0086 + 008A]	5 BCD	X		
000F	Total SAS restricted ticket in (cents) [same as meter 0082]	5 BCD	X		
0010	Total SAS restricted ticket out (cents) [same as meter 0088]	5 BCD	X		
0011	Total SAS cashable ticket in, including nonrestricted tickets (quantity) [same as meters 0081 + 0085]	4 BCD	X		
0012	Total SAS cashable ticket out, including debit tickets (quantity) [same as meters 0087 + 008B]	4 BCD	X		
0013	Total SAS restricted ticket in (quantity) [same as meter 0083]	4 BCD	X		

Distributed to Wilfried Niemeyer, adp Gauselmann GmbH

**Table C-7 (cont.)  
Meter Code Values**

Code (binary)	Meter	Min Size	Recommended Support		
			Gaming Machine	Per Game	Per Denom
0014	Total SAS restricted ticket out (quantity) [same as meter 0089]	4 BCD	X		
0015	Total ticket in, including cashable, nonrestricted and restricted tickets (credits)	4 BCD	X		
0016	Total ticket out, including cashable, nonrestricted, restricted and debit tickets (credits)	4 BCD	X		
0017	Total electronic transfers to gaming machine, including cashable, nonrestricted, restricted and debit, whether transfer is to credit meter or to ticket (credits) <i>Note: external bonus awards are metered as game win, and not as electronic transfers to gaming machine</i>	4 BCD	X		
0018	Total electronic transfers to host, including cashable, nonrestricted, restricted and win amounts (credits)	4 BCD	X		
0019	Total restricted amount played (credits)	4 BCD	X		
001A	Total nonrestricted amount played (credits)	4 BCD	X		
001B	Current restricted credits	4 BCD	X		
001C	Total machine paid payable win, not including progressive or external bonus amounts (credits)	4 BCD	X	X	X
001D	Total machine paid progressive win (credits)	4 BCD	X	X	X
001E	Total machine paid external bonus win (credits)	4 BCD	X	X	X
001F	Total attendant paid payable win, not including progressive or external bonus amounts (credits)	4 BCD	X	X	X
0020	Total attendant paid progressive win (credits)	4 BCD	X	X	X
0021	Total attendant paid external bonus win (credits)	4 BCD	X	X	X
0022	Total won credits (sum of total coin out and total jackpot)	4 BCD	X	X	X
0023	Total hand paid credits (sum of total hand paid cancelled credits and total jackpot)	4 BCD	X		
0024	Total drop, including but not limited to coins to drop, bills to drop, tickets to drop, and electronic in (credits)	4 BCD	X		
0025	Games since last power reset	4 BCD	X		
0026	Games since slot door closure	4 BCD	X		
0027	Total credits from external coin acceptor	4 BCD	X		
0028	Total cashable ticket in, including nonrestricted promotional tickets (credits)	4 BCD	X		

**Table C-7 (cont.)  
Meter Code Values**

Code (binary)	Meter	Min Size	Recommended Support		
			Gaming Machine	Per Game	Per Denom
0029	Total regular cashable ticket in (credits)	4 BCD	X		
002A	Total restricted promotional ticket in (credits)	4 BCD	X		
002B	Total nonrestricted promotional ticket in (credits)	4 BCD	X		
002C	Total cashable ticket out, including debit tickets (credits)	4 BCD	X		
002D	Total restricted promotional ticket out (credits)	4 BCD	X		
002E	Electronic regular cashable transfers to gaming machine, not including external bonus awards (credits)	4 BCD	X		
002F	Electronic restricted promotional transfers to gaming machine, not including external bonus awards (credits)	4 BCD	X		
0030	Electronic nonrestricted promotional transfers to gaming machine, not including external bonus awards (credits)	4 BCD	X		
0031	Electronic debit transfers to gaming machine (credits)	4 BCD	X		
0032	Electronic regular cashable transfers to host (credits)	4 BCD	X		
0033	Electronic restricted promotional transfers to host (credits)	4 BCD	X		
0034	Electronic nonrestricted promotional transfers to host (credits)	4 BCD	X		
0035	Total regular cashable ticket in (quantity)	4 BCD	X		
0036	Total restricted promotional ticket in (quantity)	4 BCD	X		
0037	Total nonrestricted promotional ticket in (quantity)	4 BCD	X		
0038	Total cashable ticket out, including debit tickets (quantity)	4 BCD	X		
0039	Total restricted promotional ticket out (quantity)	4 BCD	X		
003A- 003D	Reserved for future use				
003E	Number of bills currently in the stacker (Issue exception 7B when this meter is reset)	4 BCD	X		
003F	Total value of bills currently in the stacker (credits) (Issue exception 7B when this meter is reset)	4 BCD	X		
0040	Total number of \$1.00 bills accepted	4 BCD	X		
0041	Total number of \$2.00 bills accepted	4 BCD	X		
0042	Total number of \$5.00 bills accepted	4 BCD	X		
0043	Total number of \$10.00 bills accepted	4 BCD	X		
0044	Total number of \$20.00 bills accepted	4 BCD	X		

---

0045	Total number of \$25.00 bills accepted	4 BCD	X
------	--	-------	---

---

**Table C-7 (cont.)  
Meter Code Values**

Code (binary)	Meter	Min Size	Recommended Support		
			Gaming Machine	Per Game	Per Denom
0046	Total number of \$50.00 bills accepted	4 BCD	X		
0047	Total number of \$100.00 bills accepted	4 BCD	X		
0048	Total number of \$200.00 bills accepted	4 BCD	X		
0049	Total number of \$250.00 bills accepted	4 BCD	X		
004A	Total number of \$500.00 bills accepted	4 BCD	X		
004B	Total number of \$1,000.00 bills accepted	4 BCD	X		
004C	Total number of \$2,000.00 bills accepted	4 BCD	X		
004D	Total number of \$2,500.00 bills accepted	4 BCD	X		
004E	Total number of \$5,000.00 bills accepted	4 BCD	X		
004F	Total number of \$10,000.00 bills accepted	4 BCD	X		
0050	Total number of \$20,000.00 bills accepted	4 BCD	X		
0051	Total number of \$25,000.00 bills accepted	4 BCD	X		
0052	Total number of \$50,000.00 bills accepted	4 BCD	X		
0053	Total number of \$100,000.00 bills accepted	4 BCD	X		
0054	Total number of \$200,000.00 bills accepted	4 BCD	X		
0055	Total number of \$250,000.00 bills accepted	4 BCD	X		
0056	Total number of \$500,000.00 bills accepted	4 BCD	X		
0057	Total number of \$1,000,000.00 bills accepted	4 BCD	X		
0058	Total credits from bills to drop	4 BCD	X		
0059	Total number of \$1.00 bills to drop	4 BCD	X		
005A	Total number of \$2.00 bills to drop	4 BCD	X		
005B	Total number of \$5.00 bills to drop	4 BCD	X		
005C	Total number of \$10.00 bills to drop	4 BCD	X		
005D	Total number of \$20.00 bills to drop	4 BCD	X		
005E	Total number of \$50.00 bills to drop	4 BCD	X		
005F	Total number of \$100.00 bills to drop	4 BCD	X		
0060	Total number of \$200.00 bills to drop	4 BCD	X		
0061	Total number of \$500.00 bills to drop	4 BCD	X		
0062	Total number of \$1000.00 bills to drop	4 BCD	X		
0063	Total credits from bills diverted to hopper	4 BCD	X		

**Table C-7 (cont.)  
Meter Code Values**

Code (binary)	Meter	Min Size	Recommended Support		
			Gaming Machine	Per Game	Per Denom
0064	Total number of \$1.00 bills diverted to hopper	4 BCD	X		
0065	Total number of \$2.00 bills diverted to hopper	4 BCD	X		
0066	Total number of \$5.00 bills diverted to hopper	4 BCD	X		
0067	Total number of \$10.00 bills diverted to hopper	4 BCD	X		
0068	Total number of \$20.00 bills diverted to hopper	4 BCD	X		
0069	Total number of \$50.00 bills diverted to hopper	4 BCD	X		
006A	Total number of \$100.00 bills diverted to hopper	4 BCD	X		
006B	Total number of \$200.00 bills diverted to hopper	4 BCD	X		
006C	Total number of \$500.00 bills diverted to hopper	4 BCD	X		
006D	Total number of \$1000.00 bills diverted to hopper	4 BCD	X		
006E	Total credits from bills dispensed from hopper	4 BCD	X		
006F	Total number of \$1.00 bills dispensed from hopper	4 BCD	X		
0070	Total number of \$2.00 bills dispensed from hopper	4 BCD	X		
0071	Total number of \$5.00 bills dispensed from hopper	4 BCD	X		
0072	Total number of \$10.00 bills dispensed from hopper	4 BCD	X		
0073	Total number of \$20.00 bills dispensed from hopper	4 BCD	X		
0074	Total number of \$50.00 bills dispensed from hopper	4 BCD	X		
0075	Total number of \$100.00 bills dispensed from hopper	4 BCD	X		
0076	Total number of \$200.00 bills dispensed from hopper	4 BCD	X		
0077	Total number of \$500.00 bills dispensed from hopper	4 BCD	X		
0078	Total number of \$1000.00 bills dispensed from hopper	4 BCD	X		
0079	Sessions played (a session is a set of individual games purchased as a group and played in a series)	4 BCD	X	X	
007A	Tip money (credits)	4 BCD	X	X	X
007B	Total foreign bill converted amount (cents)	5 BCD	X		
007C	Total foreign bill count (quantity)	4 BCD	X		
007D- 007E	Reserved for future use				
007F	Weighted average theoretical payback percentage in hundredths of a percent (see note below)	4 BCD	X	X	

**Note:** The exact meaning of the weighted average theoretical payback percentage is based on jurisdictional requirements. For paytables which have a difference between the minimum and maximum theoretical payback, the weighted average theoretical payback percentage is calculated based on actual coin in at each different theoretical base payback percentage for the particular payable. The value is returned as a percentage in hundredths of a percent. See Section 7.24.1 for details.

**Note:** Meter codes 0080 through 0093 and 00A0 through 00BD include only amounts accumulated using the SAS protocol. They do not include amounts from any other process or protocol. These meters are never supported per game or per denom.

**Table C-7 (cont.)**  
**SAS Validation-Specific Meter Code Values**

Code (binary)	Meter	Min Size	Validation Type (Table 15.13c)
0080	Regular cashable ticket in (cents)	5 BCD	80
0081	Regular cashable ticket in (quantity)	4 BCD	80
0082	Restricted ticket in (cents)	5 BCD	81
0083	Restricted ticket in (quantity)	4 BCD	81
0084	Nonrestricted ticket in (cents)	5 BCD	82
0085	Nonrestricted ticket in (quantity)	4 BCD	82
0086	Regular cashable ticket out (cents)	5 BCD	00, 02
0087	Regular cashable ticket out (quantity)	4 BCD	00, 02
0088	Restricted ticket out (cents)	5 BCD	01, 03
0089	Restricted ticket out (quantity)	4 BCD	01, 03
008A	Debit ticket out (cents)	5 BCD	04
008B	Debit ticket out (quantity)	4 BCD	04
008C	Validated cancelled credit handpay, receipt printed (cents)	5 BCD	10
008D	Validated cancelled credit handpay, receipt printed (quantity)	4 BCD	10
008E	Validated jackpot handpay, receipt printed (cents)	5 BCD	20
008F	Validated jackpot handpay, receipt printed (quantity)	4 BCD	20
0090	Validated cancelled credit handpay, no receipt (cents)	5 BCD	40
0091	Validated cancelled credit handpay, no receipt (quantity)	4 BCD	40
0092	Validated jackpot handpay, no receipt (cents)	5 BCD	60
0093	Validated jackpot handpay, no receipt (quantity)	4 BCD	60
0094- 009F	Reserved for future use		



**Table C-7 (cont.)**  
**SAS AFT-Specific Meter Code Values**

<b>Code</b> (binary)	<b>Meter</b>	<b>Min Size</b>	<b>Transfer Type</b> (Table 8.3d)
00A0	In-house cashable transfers to gaming machine (cents)	5 BCD	00
00A1	In-House transfers to gaming machine that included cashable amounts (quantity)	4 BCD	00
00A2	In-house restricted transfers to gaming machine (cents)	5 BCD	00
00A3	In-house transfers to gaming machine that included restricted amounts (quantity)	4 BCD	00
00A4	In-house nonrestricted transfers to gaming machine (cents)	5 BCD	00
00A5	In-house transfers to gaming machine that included nonrestricted amounts (quantity)	4 BCD	00
00A6	Debit transfers to gaming machine (cents)	5 BCD	40
00A7	Debit transfers to gaming machine (quantity)	4 BCD	40
00A8	In-house cashable transfers to ticket (cents)	5 BCD	20
00A9	In-house cashable transfers to ticket (quantity)	4 BCD	20
00AA	In-house restricted transfers to ticket (cents)	5 BCD	20
00AB	In-house restricted transfers to ticket (quantity)	4 BCD	20
00AC	Debit transfers to ticket (cents)	5 BCD	60
00AD	Debit transfers to ticket (quantity)	4 BCD	60
00AE	Bonus cashable transfers to gaming machine (cents)	5 BCD	10, 11
00AF	Bonus transfers to gaming machine that included cashable amounts (quantity)	4 BCD	10, 11
00B0	Bonus nonrestricted transfers to gaming machine (cents)	5 BCD	10, 11
00B1	Bonus transfers to gaming machine that included nonrestricted amounts (quantity)	4 BCD	10, 11
00B8	In-house cashable transfers to host (cents)	5 BCD	80, 90
00B9	In-house transfers to host that included cashable amounts (quantity)	4 BCD	80, 90
00BA	In-house restricted transfers to host (cents)	5 BCD	80, 90
00BB	In-house transfers to host that included restricted amounts (quantity)	4 BCD	80, 90

**Table C-7 (cont.)**  
**SAS AFT-Specific Meter Code Values**

<b>Code</b> (binary)	<b>Meter</b>	<b>Min Size</b>	<b>Transfer Type</b> (Table 8.3d)
00BC	In-house nonrestricted transfers to host (cents)	5 BCD	80, 90
00BD	In-house transfers to host that included nonrestricted amounts (quantity)	4 BCD	80, 90
00BE- FFF9	Reserved for future use		
00FA	Regular cashable keyed-on funds (credits)	4 BCD	X
00FB	Restricted promotional keyed-on funds (credits)	4 BCD	X
00FC	Nonrestricted promotional keyed-on funds (credits)	4 BCD	X
00FD	Regular cashable keyed-off funds (credits)	4 BCD	X
00FE	Restricted promotional keyed-off funds (credits)	4 BCD	X
00FF	Nonrestricted promotional keyed-off funds (credits)	4 BCD	X

**Note:** For forward compatibility reasons, codes must not be added to Table C-7 without the express approval of IGT.

## APPENDIX D FIGURES

**Note:** Schematics are provided for reference only. The user is responsible for creating a working implementation of the reference design.

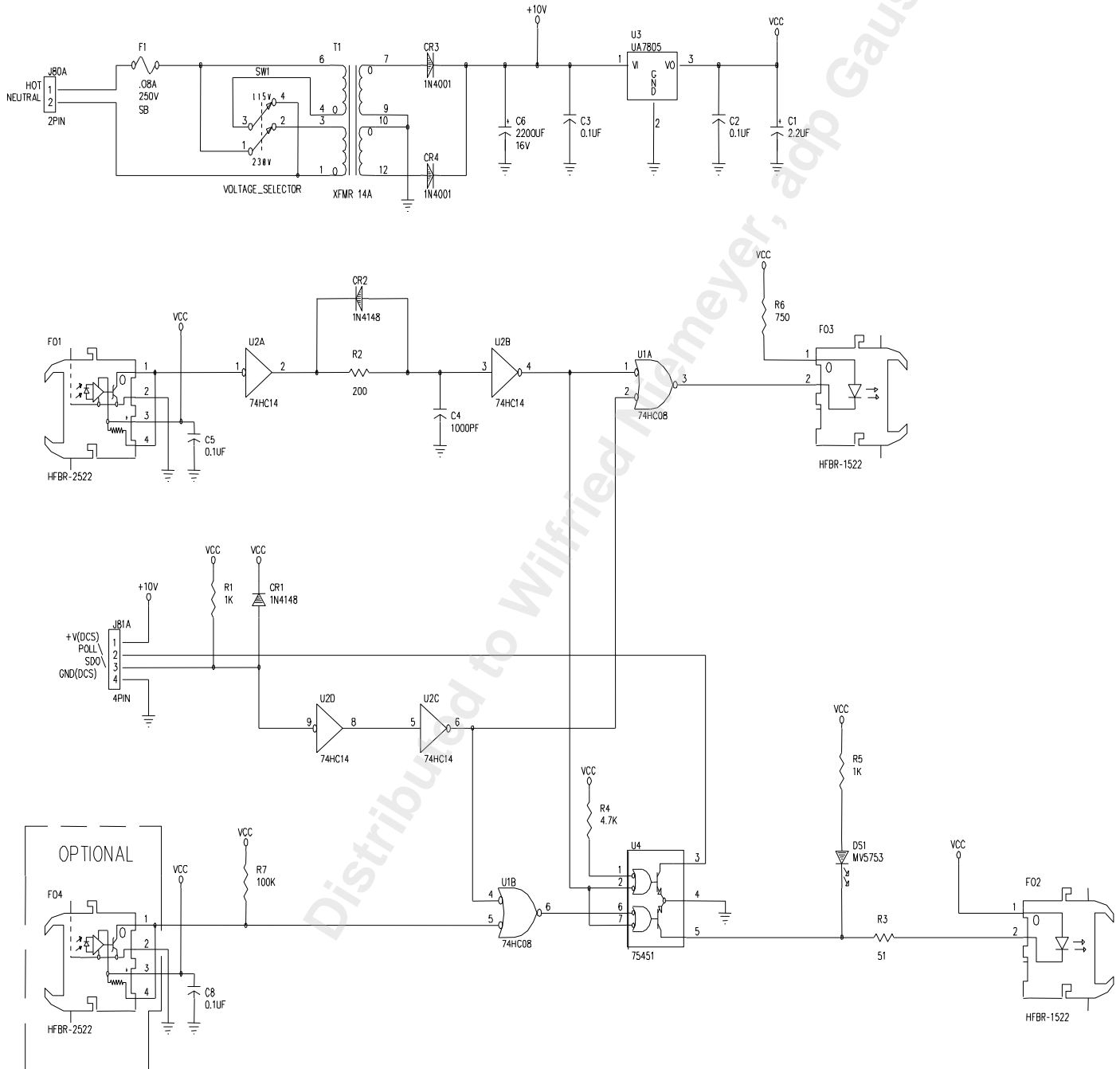


Figure 1. Sample schematic for fiber optic interface board.

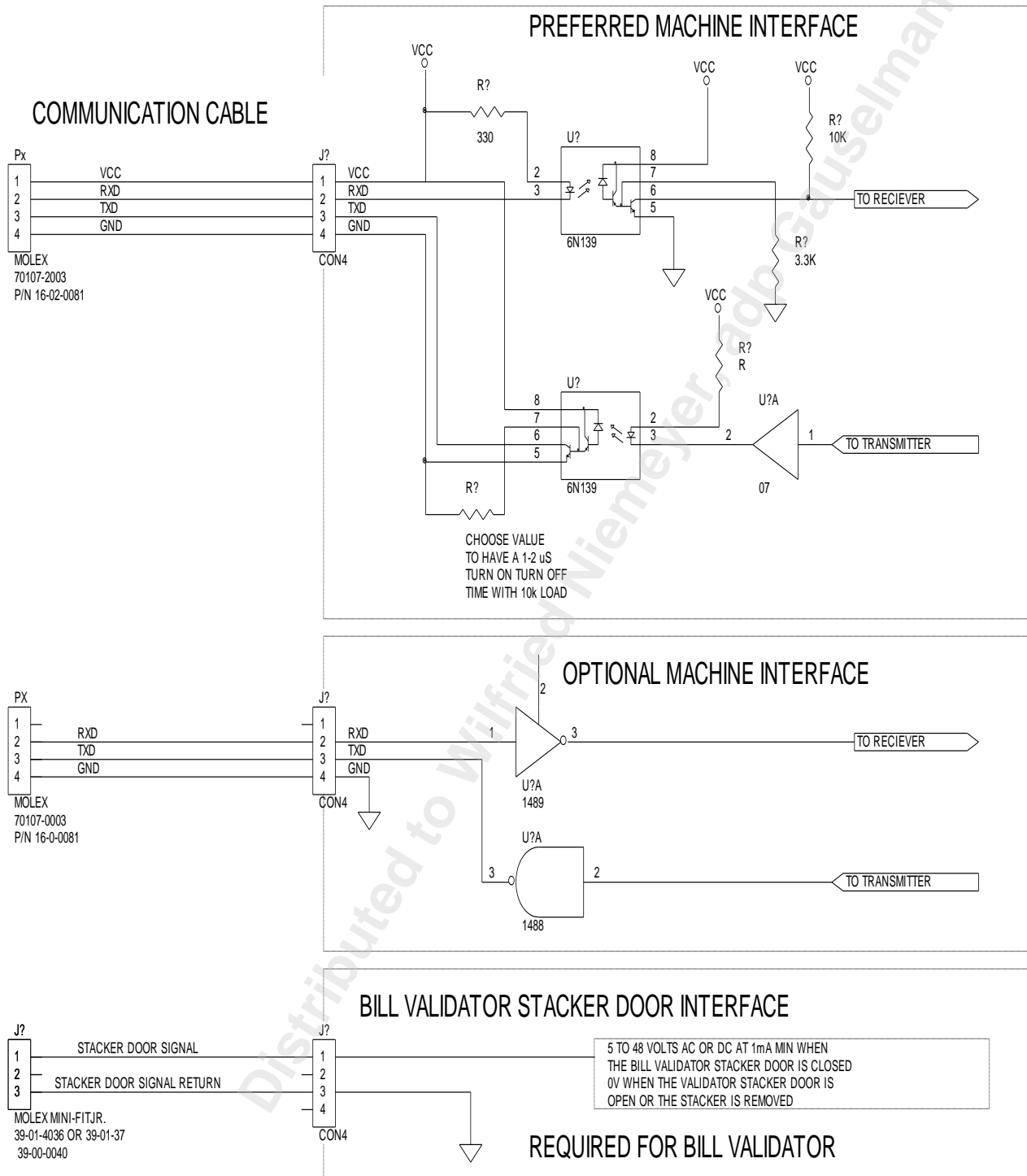


Figure 2. Sample schematic for PT95A-to-machine interface board.

## APPENDIX E LONG POLL 09 GUIDELINES

---

**Note:** *The following are general guidelines only. They have been accepted in several jurisdictions, but manufacturers should consult with the system provider in each specific jurisdiction for any special requirements.*

---

SAS long poll 09 is used primarily in lottery jurisdictions, such as West Virginia Lottery and Illinois Limited Lottery, to control which games are enabled on an EGM. The following guidelines should help in achieving consistent implementation across multiple gaming machine manufacturers. For gaming machines that support more than one SAS channel, long poll 09 is typically only supported on the primary control channel.

### Available Long Poll 09 Responses

The response to long poll 09 is either a simple ACK/NACK response or an optional BUSY response. There is no mechanism available to allow a gaming machine to explain in any detail why the poll succeeds or fails. Therefore it is important to use the available responses consistently.

- ACK means the gaming machine expects to be able to perform the requested enable or disable. Note that a game can effectively be “disabled” even if it is not currently enabled and even if it cannot be enabled.
- NACK (or ignore) means there is an error in the command and the gaming cannot perform the requested enable or disable.
- BUSY means the command is valid, but the EGM cannot perform the requested enable or disable at this time.

### Enable/Disable Game 0000

There has been some controversy as to whether or not it is valid to use game number 0000 to enable or disable all games on a gaming machine. SAS originally specified Game Number as 0000-9999. This was changed in SAS 5.02 to 0001-9999, because the protocol did not define a behavior for game number 0000. While enabling all games on the gaming machine with a single poll is unlikely to be a useful feature, the ability to disable all games with a single poll is very useful.

Therefore, it is strongly recommended that a gaming machine allow long poll 09 with game number = 0000 and enable/disable = 00 to disable all games on the gaming machine, or all games at the specified player denom when a denom is specified using the long poll B0 preamble. For backwards compatibility, it is acceptable that there be an operator configuration for this behavior. Long poll 09 with game number = 0000 and enable/disable = 01 is always an invalid poll.

### Multi-Denom Games

A game may support only a single player denom, or it may be able to be enabled for multiple denoms. There may be an operator option to configure which games and denoms may be enabled using long poll 09. Long poll 09 may be issued for a specific denom using the long poll B0 preamble, or it may be issued without specifying a denom. When a denom is specified, only games that can be enabled at that denom are affected.

### Interaction Between Host and Player/Operator

A gaming machine should not be able to be remotely configured when it is currently being played. This includes enabling and disabling games using long poll 09. Also, remote configuration should not be allowed while an operator is accessing the menu system at the gaming machine.

### Enable/Disable Tournament Games

Long poll 09 cannot be used to enable or disable tournament games. When a tournament game is enabled, real play is effectively disabled on the gaming machine. Tournament play should not affect any accounting information reported by the gaming machine. All non-tournament meters and configurations should be reported

as though tournament play was not enabled. A gaming machine should always report the games that are currently configured for real play. When tournament play is enabled, the gaming machine is assumed to be busy and cannot be remotely configured.

## Example Responses

The following examples should cover most use cases for long poll 09.

The gaming machine should ignore long poll 09 in the following conditions:

- The command is not supported.
- The CRC is not present or incorrect.
- Game number is not valid BCD or not in range (NACK is also permitted).

The gaming machine should reply with NACK in the following conditions:

- The host attempts to enable game 0000.
- The host attempts to disable game 0000 and this behavior is not supported.
- The host attempts to enable a tournament game.
- The game cannot be enabled under any circumstance (for example due to jurisdictional control).
- The game cannot be enabled for the requested denom (either denom not supported for the selected game, or operator has not enabled the requested denom for that game).
- The host attempts to enable game for all denoms when no denom is enabled for the selected game at the gaming machine.

The gaming machine should reply with BUSY in the following conditions:

- Gaming machine is in an operator menu (to prevent conflict between host control and operator control).
- Game play is active (credits in credit meter or game in progress).
- Game is in tournament mode.

The gaming machine should reply with ACK in the following conditions:

- The gaming machine expects to be able to enable or disable the selected game, or the game is already in the requested state. The actual change of game enable status may occur after the gaming machine responds. The gaming machine must issue exception A2, enabled games/denoms changed, once the change is complete. If the game was already enabled or disabled for the requested denom or denoms and no change is necessary, exception A2 is not issued.
- The host requested the gaming machine to disable a game that cannot be enabled. Exception A2 is not issued.

## GLOSSARY

AC	Alternating Current.
ACK	Acknowledge. Sent by the host and the gaming machine during communications to indicate that the transmission was received correctly.
Baud	The number of code elements transmitted per second.
BCD	Packed Binary Coded Decimal.
CRC	Cyclical Redundancy Check. A method for verifying the validity of transmitted data.
Card Cage	The housing that surrounds the processor board. Some gaming machines have a card cage lock, for security purposes, that is capable of detecting when the card cage is unlocked or opened.
Cash Box	For gaming machines equipped with bill acceptors, the cashbox is the storage unit for the accepted bills.
Drop Door	The door that provides access to the drop box.
EEPROM	Electrically Erasable Programmable Read Only Memory. Non-volatile memory used to back up the gaming machine's cumulative meters and game options.
EPROM	Erasable Programmable Read Only Memory. Non-volatile memory where the game program is stored.
LSB	Least Significant Byte.
MSB	Most Significant Byte.
ms	Millisecond. 1 millisecond is equivalent to .001 second (1/1000 sec).
NACK	Negative Acknowledgment. Sent by the host and the gaming machine during communications to indicate that the transmitted data was invalid or not received.
RAM	Random Access Memory. Volatile memory that is used by the gaming machine during normal game play.
ROM	Read Only Memory. Non-volatile memory that is used by the gaming device to store game programs.
Slot Door	The front, or top, of the gaming machine that is opened to gain access to the internal components is called the slot door.
SMIB	Smart Interface Board.
TBD	To Be Defined.

## Credit Meters

### Restricted Promotional Credits

Credits that are not redeemable for cash, and must be wagered or forfeited. Restricted promotional credits are added to the Restricted Promotional Credits Played meter when they are wagered. Restricted promotional amounts may only be removed from a gaming machine by methods that preserve the restricted status of the amounts, such as by electronically transferring restricted amounts to the host or printing restricted tickets. They must never be cashed out on a normal cashout ticket, as cashable coins or tokens, or by an attendant handpay.

### Nonrestricted Promotional Credits

Credits that may be redeemed for cash but have special accounting requirements. Nonrestricted promotional credits are added to the Nonrestricted Promotional Credits Played meter when they are wagered. Whenever nonrestricted promotional amounts are cashed out by a method that cannot preserve the nonrestricted promotional status, they become regular cashable amounts.

### Regular Cashable Credits

Credits that may be redeemed for cash and have no special accounting requirements. This includes funds from coins, bills, regular cashable tickets, and regular player winnings.

**Total Credits** All credits available to the player on the gaming machine; including restricted promotional, nonrestricted promotional, and regular cashable credits. If any combination of restricted promotional credits, nonrestricted promotional credits and regular cashable credits are in a gaming machine's credit meter at the same time, the restricted credits must be played first, then the nonrestricted credits, and finally the regular cashable credits.

## Game Meters

**Total Coin In** Total credits wagered on the gaming machine from all sources. Used to measure the total turnover or gross wagers on a gaming machine.

**Total Coin Out** Total value of all credits directly paid by the gaming machine, as a result of winning wagers and awards from an external bonusing system, whether the payout is made to the credit meter or from a cashout device (hopper, printer, electronic transfer etc.). Used as a component of the estimated win for a gaming machine.

### Machine Paid Paytable Win

The total value of all won credits metered in Total Coin Out, where the award is specifically identified by the manufacturer's par sheet. This meter does not include progressive amounts or amounts awarded as a result of an external bonusing system.

### Machine Paid Progressive Win

The total value of all won credits metered in Total Coin Out, that were for an amount determined by a progressive controller.



## Machine Paid

## External

**Bonus Win** The total value of all won credits metered in Total Coin Out, where the credits won are due to a bonus award or jackpot multiplier from an external bonus controller.

**Total Jackpot** The cumulative sum of all credits paid by an attendant, as a result of winning wagers and awards from an external bonusing system. This includes handpays resulting from progressive jackpots, bonus pays and/or game wins regardless of whether or not the win is one of the top jackpots. Credits added to this meter are NOT added to the Total Coin Out meter. While jurisdictional rules may dictate the criteria by which won credits are metered in either coin out or jackpot, the sum of Total Coin Out and Total Jackpot must always equal the total won credits. Credits accounted for in the Total Hand Paid Cancelled Credits meter are never added to this meter. Used as a component of the estimated win for a gaming machine and to reconcile metered payments with actual payments.

## Attendant Paid

**Paytable Win** The total value of all won credits metered in Total Jackpot, where the award is specifically identified by the manufacturer's par sheet. This meter does not include progressive amounts or amounts awarded as a result of an external bonusing system.

## Attendant Paid

## Progressive

**Win** The total value of all won credits metered in Total Jackpot, that were for an amount determined by a progressive controller.

## Attendant Paid

## External

**Bonus Win** The total value of all won credits metered in Total Jackpot, where the credits won are due to a bonus award or jackpot multiplier from an external bonus controller.

**Games Played** Total count of games played on the gaming machine. Used to calculate the average wager per game and as a gross measure of casino activity.

## Total Hand Paid

## Cancelled

**Credits** The cumulative sum in credits of all handpays that resulted from the player pressing the "cash out" button or otherwise cashed out from the credit meter. These do not include any credits added to the Total Jackpot meter. Total Hand Paid Credits can be calculated as the sum of Total Jackpot and Total Hand Paid Cancelled Credits. Used to reconcile metered payments with actual payments.

**Hopper Level** The current hopper level of the gaming machine in coins/tokens. The net change in hopper level over a period of time is used in the calculation of net win in some jurisdictions, e.g. NSW, Australia. It can also be used to detect hopper fills, to verify hopper empty conditions, and to prevent employee theft.

## Total Cancelled

## Credits

This meter must include all credits removed from a gaming machine except those paid by a hopper and those metered in the Total Jackpot meter. This includes, at a minimum, all credits in the Total Hand Paid Cancelled Credits meter, all credits paid directly to the player by a cashout ticket, and all credits transferred from the game

electronically. The fact that tickets are included in Total Cancelled Credits must be indicated in the long poll A0 response.

Bill Drop	Total credits received by the game from the bill acceptor. Used during soft count reconciliation.
Coin Drop	Total credit value of coins dropped by the game. Used during hard count reconciliation.
Total Drop	This meter must include all credits added to a gaming machine from an external source that do not go to a hopper. This includes, at a minimum, total credits received by the game from the bill acceptor, plus total credits from coins dropped by the game, plus total credits from all tickets redeemed (stacked) by the gaming machine, plus total credits transferred electronically to the gaming machine. The fact that tickets are included in Total Drop must be indicated in the long poll A0 response. Used during hard count and soft count reconciliation.
Total Hand Paid Credits	The cumulative sum of all credits paid by an attendant, the value of which is equal to the sum of the Total Jackpot meter and the Total Hand Paid Cancelled Credits meter.
True Coin In	Total coins/tokens received by the game from coin acceptors. Used to calculate the estimated hopper level.
True Coin Out	Total coins/tokens paid out by the gaming machine. Used to calculate the estimated hopper level.

## Denominations

### Accounting

**Denomination** Reported to the host via long polls 1F and 53, this is the denomination used for all credit values reported to the host, except for those values specifically defined to be in a different unit of measure, such as cents or tokens, or those specifically defined to be in units of player credits. **Please note, when the term “credits” is used in this document without qualifiers, such as “credit meter” or “cancelled credits,” it is not implying any specific method of storage or representation.**

**Game Credits** On a multi-denomination game, it is sometimes convenient to refer to credits in terms independent from any specific denomination. For example, a game's payable, max bet, etc., is generally expressed in terms of credits. Even though max bet can conceivably be different for different player denominations, the max bet amount can be expressed in units of game credits, independent of the denomination used for game play.

### Player/Game

**Denomination** On a multi-denomination game, the denominations available to the player for wagering are called the player denominations or game denominations.

### Token

**Denomination** The denomination of the coin mechanism and/or hopper is called the token denomination.