

Thales e-Security

payShield 9000 v3.0b

Console Reference Manual

1270A544-030

13 December 2015



Contents

CONTENTS	2
END USER LICENSE AGREEMENT.....	6
CHAPTER 1 – INTRODUCTION.....	11
ABOUT THIS MANUAL	11
LIST OF CONSOLE COMMANDS (ALPHABETICAL)	11
LIST OF CONSOLE COMMANDS (FUNCTIONAL)	15
CHAPTER 2 – CONFIGURATION COMMANDS	19
CONFIGURATION COMMANDS	19
Reset to Factory Settings	20
Configure Commands	22
Configure PIN Block Formats	24
Configure Security.....	26
View Security Configuration	36
Configure Console Port	41
View Console Port Configuration	43
Configure Host Port	44
View Host Port Configuration	50
Host Port Access Control list (ACL) Configuration.....	53
Configure Printer Port.....	56
View Printer Port Configuration	59
Configure Management Port.....	60
View Management Port Configuration.....	62
Configure Alarms	63
View Alarm Configuration	64
Add Static TCP/IP Route.....	65
View/Change Instantaneous Utilization Period	68
Suspend/Resume Collection of Utilization Data	69
Suspend/Resume Collection of Health Check Counts.....	70
View SNMP Settings	71
Add a SNMP Community or User	72
Delete a SNMP Community or User.....	73
FRAUD DETECTION COMMANDS	74
Configure Fraud Detection.....	75
Re-enable PIN Verification.....	76
DIAGNOSTIC COMMANDS.....	77
Diagnostic Test	78
View Software Revision Number	81
View Available Commands	85
Show Network Statistics	87
Test TCP/IP Network.....	91
Trace TCP/IP route.....	93
View/Reset Utilization Data	95
View/Reset Health Check Counts.....	97
Check the FICON Host Interface.....	98
CHAPTER 3 – LOCAL MASTER KEYS.....	100
TYPES OF LMKs	100
MULTIPLE LMKs	100
LMK TABLE	101
LMK COMMANDS	102
Generate LMK Component(s)	103
Load LMK	107

Load 'Old' LMKs into Key Change Storage	111
Verify LMK Store	115
Duplicate LMK Component Sets.....	116
Delete LMK	117
Delete 'Old' LMK from Key Change Storage.....	118
View LMK Table	119
CHAPTER 4 – OPERATIONAL COMMANDS	122
AUTHORIZATION COMMANDS	122
Enter the Authorized State	123
Cancel the Authorized State	125
Authorize Activity.....	126
Cancel Authorized Activity.....	135
View Authorized Activities	137
LOGGING COMMANDS.....	138
Display the Error Log	140
Clear the Error Log.....	141
Display the Audit Log.....	142
Clear the Audit Log	144
Audit Options.....	145
Print the Audit Log.....	149
TIME AND DATE COMMANDS	150
Set the Time and Date	151
Query the Time and Date.....	152
Set Time for Automatic Self-Tests.....	153
SETTINGS, STORAGE AND RETRIEVAL COMMANDS.....	154
Save HSM Settings to a Smartcard	155
Retrieve HSM Settings from a Smartcard.....	156
KEY MANAGEMENT COMMANDS	159
Generate Key Component.....	160
Generate Key and Write Components to Smartcard	163
Encrypt Clear Component	167
Form Key from Components	170
Generate Key	176
Import Key	180
Export Key.....	183
Generate a Check Value.....	186
PAYMENT SYSTEM COMMANDS	188
Generate a Card Verification Value	189
Generate a VISA PIN Verification Value.....	191
Load the Diebold Table	193
Encrypt Decimalization Table	195
Translate Decimalization Table	197
Generate a MAC on an IPB	199
SMARTCARD COMMANDS	200
Format an HSM Smartcard.....	201
Create an Authorizing Officer Smartcard	203
Verify the Contents of a Smartcard	204
Change a Smartcard PIN.....	205
Read Unidentifiable Smartcard Details	206
Eject a Smartcard.....	207
DES CALCULATOR COMMANDS	208
Single-Length Key Calculator	209
Double-Length Key Calculator	210
Triple-Length Key Calculator.....	211
LEGACY COMMANDS.....	212
Generate a ZMK Component	213
Generate a ZMK & Write to Smartcards.....	214

<i>Encrypt a Clear ZMK Component</i>	216
<i>Form a ZMK from Encrypted Components</i>	217
<i>Form a Key from Components</i>	219
<i>Import a CVK or PVK</i>	221
<i>Generate a Zone PIN Key</i>	223
<i>Translate a Zone PIN Key</i>	225
<i>Generate a CVK Pair</i>	226
<i>Translate a CVK Pair from LMK to ZMK</i>	227
<i>Generate a Double-Length ZMK Component</i>	228
<i>Form a ZMK from Clear Components</i>	229
<i>Generate a BDK</i>	231
<i>Generate & Export a KML</i>	233
<i>Generate a CCK</i>	234
<i>Export a CCK</i>	235
CHAPTER 5 – PAYSHIELD MANAGER	236
INTRODUCTION	236
<i>Initialize Domain Authority</i>	237
<i>Generate an HSM Certificate</i>	239
<i>Backup Domain Authority Card</i>	241
<i>Add a RACC to the whitelist</i>	242
<i>Decommission the HSM</i>	243
<i>Remove RACC from the whitelist</i>	244
<i>Commission the HSM</i>	245
<i>Generate Customer Trust Anchor</i>	246
<i>Make an RACC left or right key</i>	247
<i>Commission a smartcard</i>	248
<i>Transfer existing LMK to RLMK</i>	249
<i>Decommission a smartcard</i>	251
<i>HSM commissioning status</i>	252
<i>Duplicate CTA share</i>	253
CHAPTER 6 – SECURE HOST COMMS	254
INTRODUCTION	254
<i>Generate Certificate Signing Request</i>	255
<i>Import Certificate</i>	258
<i>Export HSM Certificate's Chain of Trust</i>	260
<i>View Installed Certificate(s)</i>	262
<i>Delete Installed Certificate(s)</i>	264
<i>Generate HRK</i>	265
<i>Change HRK Passphrase</i>	266
<i>Restore HRK</i>	267
APPENDIX A – ERROR CODES	268
APPENDIX B – CORE HSM COMMANDS	269
APPENDIX C – PIN BLOCK FORMATS	270
APPENDIX D – KEY SCHEME TABLE	271
APPENDIX E – VARIANT LMKS	272
APPENDIX F – KEYBLOCK LMKS	273
APPENDIX G – LIST OF AUTHORIZABLE ACTIVITIES	274
APPENDIX H – REDUCED CHARACTER SETS	275
APPENDIX I – CONFIGURE SECURITY SETTINGS	276
APPENDIX J – FRAUD DETECTION FUNCTIONS	277
APPENDIX K – THALES KEYBLOCK / TR-31 KEY USAGE CONVERSION	278

APPENDIX L – UTILIZATION DATA	279
APPENDIX M – HEALTH CHECK DATA.....	280
APPENDIX N – PCI HSM COMPLIANCE	281
APPENDIX O – ERROR RESPONSES EXCLUDED FROM AUDIT LOG	282
GLOSSARY	283
GENERAL ABBREVIATIONS	284

End User License Agreement

Please read this Agreement carefully.

THALES E-SECURITY IS WILLING TO LICENSE SOFTWARE TO THE ENTITY THAT HAS PURCHASED A THALES E-SECURITY HARDWARE DEVICE UPON THE CONDITION THAT ALL OF THE TERMS CONTAINED IN THIS END USER LICENSE AGREEMENT ("AGREEMENT") ARE ACCEPTED AND AGREED. PLEASE READ THIS AGREEMENT CAREFULLY. THE TERMS OF THIS AGREEMENT WILL BE DEEMED TO HAVE BEEN AGREED TO BY THE ENTITY OR END USER CUSTOMER THAT HAS PURCHASED A THALES E-SECURITY HARDWARE DEVICE IF SOFTWARE IS DOWNLOADED OR IF SOFTWARE IS USED OR IF A SECURITY SEAL ON THE MEDIA PACKAGE CONTAINING SOFTWARE IS BROKEN OR IF CONSENT IS MANIFESTED BY CLICKING ON AN ACCEPTANCE KEY.

This document is a legal agreement between Thales e-Security, Inc., ("Thales e-Security"), 900 South Pine Island Road, Suite 710, Plantation, FL 33324 U.S.A. and the end user customer that has purchased a Thales e-Security hardware device, (hereafter referred to as the "End User Customer"). Any person who manifests their agreement to this Agreement represents that they have the requisite and appropriate legal authority to bind the End User Customer.

1. Definitions

- (a) "Affiliates" means Thales Transport & Security (Hong Kong) Ltd. and Thales UK Limited.
- (b) "Software" means machine readable instructions and all modifications and customizations thereof in binary form and any other machine readable materials (including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Thales e-Security or its corporate Affiliates that direct a computer's processor to perform specific operations.

2. Ownership

Software consists of a combination of proprietary components that are owned by or licensed to Thales e-Security or its Affiliates together with free or open source components ("Free Software Components") that are identified in the text files that are provided with the Software. ONLY THOSE TERMS AND CONDITIONS SPECIFIED FOR, OR APPLICABLE TO, EACH SPECIFIC FREE SOFTWARE COMPONENT PURSUANT TO ITS APPLICABLE GOVERNING LICENSE SHALL BE APPLICABLE TO SUCH FREE SOFTWARE COMPONENT. Each Free Software Component is the copyright of its respective copyright owner. Software is licensed to End User Customer and is not sold. End User Customer has no ownership rights in the Software. Rather, End User Customer is hereby granted a license to use the Software. The Software is copyrighted by Thales e-Security or its Affiliates or its suppliers. End User Customer hereby agrees to respect and not to remove or conceal from view any copyright or trademark notice appearing on the Software or documentation, and to reproduce all copyright or trademark notices on any copy of the Software and documentation or any portion thereof and on all portions contained in or merged into other programs and documentation.

3. License to Use

Subject to the terms and conditions of this Agreement Thales e-Security grants to End User Customer a non-exclusive, limited license to use Software unmodified for the sole purpose of running or operating Software on or with a Thales e-Security hardware device and to copy such Software provided that such copies are made in machine readable form for backup purposes.

4. Restrictions

Software is confidential and copyrighted. Unless enforcement is prohibited by applicable law, End User Customer may not modify, decompile, or reverse engineer Software. End User Customer shall not permit any other person to do any of the same. End User Customer may not rent, lease or sublicense the Software. Any rights not expressly granted by Thales e-Security to End User Customer hereunder are reserved by Thales e-Security and its licensors and all implied licenses are disclaimed. Any other use of the Software by any other entity is strictly forbidden and is a violation of this Agreement. The Software and any accompanying written materials are protected by international copyright and patent laws and international trade provisions. No right, title or interest is granted under this Agreement in or to any trademark, service mark, logo or trade name of Thales, S.A., Thales e-Security or its licensors or corporate Affiliates. End User Customer may not disassemble the Thales e-Security owned or licensed components of the Software. End User Customer may not create

derivative works based on the Software except as may be necessary to permit integration with other technology.

5. Limited Warranty

(a) Thales e-Security warrants that a Thales e-Security hardware device and the accompanying Software will function substantially as detailed in their respective and applicable specifications. The warranty period for a Thales e-Security hardware device is one year from the date of delivery and the warranty period for Software is ninety (90) days from the date of delivery. If either a Thales e-Security hardware device or Software fails to materially conform to their applicable specifications, Thales e-Security or its Affiliates will repair or replace the affected hardware device or Software provided that End User Customer provides Thales e-Security with a written notice of a claim or a defect under this warranty within the warranty period herein described. FOR THE AVOIDANCE OF DOUBT, THALES E-SECURITY NEITHER WARRANTS, NOR CAN BE EXPECTED TO WARRANT THAT A THALES E-SECURITY HARDWARE DEVICE OR SOFTWARE IS WHOLLY FREE FROM DEFECT, OR THAT ANY PARTICULAR DEFECT CAN BE REMEDIED, OR THAT A REMEDY CAN BE PROVIDED IN ANY PARTICULAR TIMEFRAME. THE FOREGOING WARRANTY SHALL NOT APPLY IF THE NONCONFORMITY ISSUE IS CAUSED BY ANY MODIFICATION OR REPAIRS TO A THALES E-SECURITY HARDWARE DEVICE OR SOFTWARE PERFORMED BY ANYONE OTHER THAN THALES E-SECURITY OR TO ANY ASSOCIATED OR COMPLEMENTARY EQUIPMENT OR SOFTWARE NOT FURNISHED BY THALES E-SECURITY OR ITS CORPORATE AFFILIATES, OR BY ANY HARDWARE DEVICE OR SOFTWARE MISUSE OR NEGLECT.

(b) NOTWITHSTANDING THE FOREGOING, TO THE MAXIMUM EXTENT PERMITTED BY LAW, THALES E-SECURITY, ON BEHALF OF ITSELF, ITS AFFILIATES AND ITS THIRD PARTY SUPPLIERS, HEREBY DISCLAIMS ANY AND ALL WARRANTIES OF ANY OTHER KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. THALES E-SECURITY DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE HARDWARE DEVICE OR SOFTWARE WILL MEET ANY REQUIREMENTS OR NEEDS THAT END USER CUSTOMER MAY HAVE, OR THAT A THALES E-SECURITY HARDWARE DEVICE OR SOFTWARE WILL OPERATE ERROR FREE, OR IN AN UNINTERRUPTED FASHION, OR THAT ANY DEFECTS OR ERRORS WILL BE CORRECTED, OR THAT THE SOFTWARE OR HARDWARE DEVICE IS COMPATIBLE WITH ANY PARTICULAR PLATFORM. SOME JURISDICTIONS DO NOT ALLOW FOR THE WAIVER OR EXCLUSION OF IMPLIED WARRANTIES SO THEY MAY NOT APPLY. IF THIS EXCLUSION IS HELD TO BE UNENFORCEABLE BY A COURT OF COMPETENT JURISDICTION, THEN ALL EXPRESS AND IMPLIED WARRANTIES SHALL BE LIMITED IN DURATION TO A PERIOD OF THIRTY (30) DAYS FROM THE DATE OF PURCHASE OF THE HARDWARE DEVICE OR SOFTWARE, AND NO WARRANTIES SHALL APPLY AFTER THAT PERIOD.

6. Intellectual Property Indemnification

Thales e-Security shall defend or, at its option, settle, any claim, action or proceeding brought against End User Customer alleging that a Thales e-Security hardware device or Software infringes upon a trademark, patent, copyright or trade secret or other intellectual property right in a country that is a signatory to the Berne Convention, and shall indemnify End User Customer from and against all damages and costs finally awarded against End User Customer in any such action or proceeding, provided that End User Customer (a) promptly notifies Thales e-Security in writing of the claim, (b) gives Thales e-Security full authority, information and assistance to defend such claim and (c) gives Thales e-Security sole control of the defense of such claim and all negotiations for the compromise or settlement thereof. If a Thales e-Security hardware device or Software or any part thereof becomes, or in the opinion of Thales e-Security is likely to become the subject of a valid claim of infringement or the like under any trademark, patent, copyright or trade secret or other intellectual property right law, Thales e-Security shall have the right, at its option and expense, either to obtain for End User Customer a license permitting the continued use of the Thales hardware device or Software or such part, or to replace or modify it so that it becomes non-infringing. Thales e-Security shall have no liability hereunder for any costs incurred or settlement entered into without its prior written consent. Thales e-Security shall have no liability hereunder with respect to any claim based upon (i) the combination of a Thales hardware device or Software with other equipment not furnished by Thales e-Security (except if the infringement occurs due to the use of the Thales e-Security hardware device or Software itself as originally provided by Thales e-Security or its Affiliates); (ii) any addition to or modification of a Thales e-Security hardware device or Software by any person or entity other than Thales e-Security or its Affiliates; or (iii) use of a superseded or altered release of the Software. THE FOREGOING STATES THE SOLE AND EXCLUSIVE LIABILITY OF THALES E-SECURITY AND ITS LICENSORS AND THE SOLE AND EXCLUSIVE REMEDY OF END USER CUSTOMER WITH RESPECT TO ANY CLAIM OF PATENT, COPYRIGHT, TRADEMARK, TRADE SECRET OR OTHER INTELLECTUAL PROPERTY RIGHTS INFRINGEMENT BY A THALES E-SECURITY HARDWARE DEVICE OR SOFTWARE, ANY SERVICE, ANY PART THEREOF OR THE USE THEREOF, AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED OR ARISING BY CUSTOM OR TRADE USAGE, AND INDEMNITIES WITH RESPECT THERETO. NOTWITHSTANDING

THE FOREGOING, ALL OPEN SOURCE SOFTWARE OR FREWARE INCLUDED WITH THE SOFTWARE IS PROVIDED WITHOUT ANY RIGHTS TO INDEMNIFICATION.

7. Limited Liability

TO THE EXTENT ALLOWED BY LAW, IN NO EVENT WILL THALES E-SECURITY OR ITS CORPORATE AFFILIATES OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR ANY SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF THALES E-SECURITY OR ANY OF ITS CORPORATE AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will Thales e-Security or its corporate affiliate's liability, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by End User Customer for the Thales e-Security hardware device or Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose.

8. Export Restrictions.

Thales e-Security hardware devices and Software are subject to the restrictions imposed by the United States Export Administration Regulations and the United Kingdom Export Control Order 2008 and may be subject to export or import regulations in other countries. End User Customer agrees to comply strictly with all such laws and regulations and acknowledges that it has the responsibility to obtain such licenses to export, re-export, or import as may be required.

9. Transfer Rights

End User Customer may transfer a Thales e-Security hardware device together with the Software, and this license to another party if the other party agrees to accept the terms and conditions of this Agreement and notice of such transfer and acceptance is provided to Thales e-Security in writing. FOR THE AVOIDANCE OF DOUBT, IF END USER CUSTOMER TRANSFERS POSSESSION OF ANY COPY OF THE SOFTWARE TO ANOTHER PARTY, EXCEPT AS PROVIDED IN THIS SECTION 9, THE LICENSE GRANT PROVIDED HEREIN IS AUTOMATICALLY REVOKED, CANCELLED AND TERMINATED.

10. Termination

This Agreement is effective until terminated. End User Customer may terminate this Agreement at any time by destroying or erasing all copies of the Software and accompanying written materials in its possession or control. This license will terminate automatically, without notice from Thales e-Security if End User Customer fails to comply with the terms and conditions of this Agreement. Upon such termination, End User Customer shall destroy or erase all copies of the Software (together with all modifications, upgrades and merged portions in any form) and any accompanying written materials in its possession or control.

11. U.S. Government Acquisitions

Software and Documentation acquired by the U.S. Government or on its behalf is furnished with "RESTRICTED RIGHTS," as defined in Federal Acquisition Regulation ("FAR") 52.227-19(b)(2), and DFAR 252.227-7013 to 7019, as applicable. Use, duplication or disclosure of the Software and Documentation by the U.S. Government and parties acting on its behalf is governed by and subject to the restrictions set forth in FAR 52.227-19(b)(1) and (2) or DFAR 252.227-7013 to 7019, as applicable.

12. Governing Law and Dispute Resolution

(a) If an End User Customer is located in Anguilla, Antigua and Barbuda, Argentina, Aruba, Bahamas, Barbados, Belize, Bermuda, Bolivia, Bonaire, Sint Eustatius and Saba, Brazil, British Virgin Islands, Canada, Cayman Islands, Chile, Colombia, Costa Rica, Cuba, Curaçao, Dominica, Dominican Republic, Ecuador, El Salvador, Falkland Islands (Malvinas), French Guiana, Greenland, Grenada, Guadeloupe, Guatemala, Guyana, Haiti, Honduras, Jamaica, Martinique, Mexico, Monserrat, Nicaragua, Panama, Paraguay, Peru, Puerto Rico, Saint-Barthélemy, St. Kitts and Nevis, Saint Lucia, Saint Martin, Saint Vincent and the Grenadines, Saint Pierre and Miquelon, Sint Maarten, Suriname, Trinidad and Tobago, Turks and Caicos Islands, United States, United States Virgin Islands, Uruguay or Venezuela, this End User License Agreement shall be construed, interpreted and governed by the laws of the State of Florida, United States of America without regard to conflicts of laws and all disputes shall be submitted to the State or Federal courts located in Florida.

(c) If an End User Customer is located in Algeria, Angola, Afghanistan, Albania, Andorra, Armenia, Austria, Azerbaijan, Bahrain, Bangladesh, Belarus, Belgium, Benin, Bhutan, Bosnia-Herzegovina, Botswana, Bulgaria, Burkina Faso, Burundi, Cameroon, Cape Verde, Central African Republic, Chad, Comoros, Congo, Cote d'Ivoire,

Croatia, Cyprus, Czech Republic, Denmark, Democratic Republic of the Congo, Djibouti, Egypt, Equatorial Guinea, Eritrea, Estonia, Ethiopia, Finland, Faroe Islands, France, Gabon, Gambia, Germany, Gibraltar, Georgia, Greece, Ghana, Guernsey and Alderney, Guinea, Guinea-Bissau, Hungary, Iceland, India, Iran, Iraq, Ireland, Island of Man, Israel, Italy, Jersey, Jordan, Kazakhstan, Kenya, Kosovo, Kyrgyzstan, Kuwait, Latvia, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Lebanon, Luxembourg, Macedonia, Madagascar, Malawi, Maldives, Mali, Malta, Mauritania, Mauritius, Moldova, Monaco, Montenegro, Morocco, Mozambique, Namibia, Nepal, Netherlands, Niger, Nigeria, Norway, Oman, Palestine, Pakistan, Poland, Portugal, Qatar, Romania, Russia, Rwanda, San Marino, Sao Tome & Principe, Saudi Arabia, Senegal, Serbia, Seychelles, Sierra Leone Slovakia, Slovenia, Somalia, South Africa, South Sudan, Sudan, Spain, Sri Lanka, Swaziland, Sweden, Svalbard and Jan Mayen Islands, Switzerland, Syria, Tajikistan, Tanzania, Togo, Tunisia, Turkey, Turkmenistan, Uganda, Ukraine, United Arab Emirates, United Kingdom, Uzbekistan, Vatican City State (Holy See), Yemen, Zambia, or Zimbabwe, this End User License Agreement will be governed by the laws of England and Wales and all disputes shall be submitted to the courts of England.

(c) If an End User Customer is located in American Samoa, Australia, Brunei Darussalam, Cambodia, China, Cook Islands, Fiji, Guam, Hong Kong, Indonesia, Japan, Kiribati, Laos, Macao, Malaysia, Marshall Islands, Micronesia, Mongolia, Myanmar (ex-Burma), New Caledonia, New Zealand, Papua New Guinea, Philippines, Samoa, Singapore, Solomon Islands, South Korea, Taiwan, Thailand, Timor Leste, Tonga, Tuvalu, Vanuatu, Vietnam or Western Samoa, this End User License Agreement will be governed by the Law of the Hong Kong Special Administrative Region of the People's Republic of China and all disputes shall be submitted to arbitration in Hong Kong.

13. Elliptic Curve Cryptography Activation

If End User Customer elects to purchase a Thales e-Security hardware device containing elliptic curve cryptography software (ECC) it agrees that its use of ECC is limited to storing cryptographic keys and the performance of cryptographic operations in a hardware environment together with the management and issuance of digital certificates by a registration authority or certificate authority provided such certificates are either (i) solely for the internal use of the registration authority or (ii) are solely for the internal use of an enterprise that is hosted by a registration authority or certificate authority. No right or license is provided or granted to use ECC as part of a third party service provider for the purpose of acting as a commercial registration authority or certificate authority as part of a commercial service offered by an enterprise, either as a vendor of digital certificates or in the provisioning of certificates for use in a commercial service.

This page is intentionally left blank.

Chapter 1 – Introduction

About this Manual

This manual is a reference document containing details of all commands that can be used on the HSM console. For other payShield 9000 information, see the following manuals:

- > *payShield 9000 Security Operations Manual*
- > *payShield 9000 Installation Manual*
- > *payShield 9000 Host Programmers Manual*
- > *payShield 9000 Host Command Reference Manual*

List of Console Commands (Alphabetical)

Command	Function	Chapter	Page
A	Enter the Authorized State	4	123
A	Authorize Activity	4	126
A5	Configure Fraud Detection	2	75
A7	Re-enable PIN Verification	2	76
AUDITLOG	Display the Audit Log	4	142
AUDITOPTIONS	Audit Options	4	145
AUDITPRINT	Print the Audit Log	4	149
B	Generate a Zone PIN Key	4	223
BK	Form a Key from Components	4	219
C	Cancel the Authorized State	4	125
C	Cancel Authorized Activity	4	135
CC	Configure Console Port	2	41
CH	Configure Host Port	2	44
CK	Generate a Check Value	4	186
CL	Configure Alarms	2	63
CLEARAUDIT	Clear the Audit Log	4	144
CLEARERR	Clear the Error Log	4	141
CM	Configure Management Port	2	60
CO	Create an Authorizing Officer Smartcard	4	203
CONFIGACL	Host Port Access Control list (ACL) Configuration	2	53
CONFIGCMDS	Configure Commands	2	22
CONFIGPB	Configure PIN Block Formats	2	24
CP	Configure Printer Port	2	56
CS	Configure Security	2	26
CV	Generate a Card Verification Value	4	189
D	Form a ZMK from Encrypted Components	4	217
DA	Generate & Export a KML	4	233

Command	Function	Chapter	Page
DC	Duplicate LMK Component Sets	3	116
DD	Generate a Double-Length ZMK Component	4	228
DE	Form a ZMK from Clear Components	4	229
DG	Generate a BDK	4	231
DM	Delete LMK	3	117
DO	Delete 'Old' LMK from Key Change Storage	3	118
DT	Diagnostic Test	2	78
EC	Encrypt Clear Component	4	167
ED	Encrypt Decimalization Table	4	195
EJECT	Eject a Smartcard	4	207
ERRLOG	Display the Error Log	4	140
F	Generate a ZMK Component	4	213
FC	Format an HSM Smartcard	4	201
FICONTEST	Check the FICON Host Interface	2	98
FK	Form Key from Components	4	170
GC	Generate Key Component	4	160
GETCMDS	View Available Commands	2	85
GETTIME	Query the Time and Date	4	152
GK	Generate LMK Component	3	103
GS	Generate Key and Write Components to Smartcard	4	163
GZ	Generate a ZMK & Write to Smartcards	4	214
HEALTHENABLE	Suspend/Resume Collection of Health Check Counts	2	70
HEALTHSTATS	View/Reset Health Check Counts	2	97
IK	Import Key	4	180
IV	Import a CVK or PVK	4	221
KA	Generate a CVK Pair	4	226
KB	Translate a CVK Pair from LMK to ZMK	4	227
KE	Export Key	4	183
KG	Generate Key	4	176
LK	Load LMK	3	107
LO	Load 'Old' LMKs into Key Change Storage	3	111
MI	Generate a MAC on an IPB	4	199
N	Single-Length Key Calculator	4	209
NETSTAT	Show Network Statistics	2	87
NP	Change a Smartcard PIN	4	205
PING	Test TCP/IP Network	2	91
PV	Generate a VISA PIN Verification Value	4	191
QC	View Console Port Configuration	2	43
QH	View Host Port Configuration	2	50
QL	View Alarm Configuration	2	64

Command	Function	Chapter	Page
QM	View Management Port Configuration	2	62
QP	View Printer Port Configuration	2	59
QS	View Security Configuration	2	36
R	Load the Diebold Table	4	193
RC	Read Unidentifiable Smartcard Details	4	206
RESET	Reset to Factory Settings	2	20
RH	Initialize	5	237
RI	Generate an HSM Certificate	5	239
ROUTE	Add Static TCP/IP Route	5	65
RS	Retrieve HSM Settings from a Smartcard	4	156
RZ	Backup Domain Authority Card	5	241
SD	Delete Installed Certificate(s)	6	264
SE	Export HSM Certificate's Chain of Trust	6	260
SETTIME	Set the Time and Date	4	151
SG	Generate Certificate Signing Request	6	255
SI	Import Certificate	6	258
SK	Generate HRK	6	265
SL	Restore HRK	6	267
SP	Change HRK Passphrase	6	266
SNMP	View SNMP Settings	2	71
SNMPADD	Add a SNMP Community or User	2	72
SNMPDEL	Delete a SNMP Community or User	2	73
SS	Save HSM Settings to a Smartcard	4	154
ST	Set Time for Automatic Self-Tests	4	153
SV	View Installed Certificate(s)	6	262
T	Triple-Length Key Calculator	4	211
TD	Translate Decimalization Table	4	197
TRACERT	Trace TCP/IP route	2	93
UTILCFG	View/Change Instantaneous Utilization Period	2	68
UTILENABLE	Suspend/Resume Collection of Utilization Data	2	69
UTILSTATS	View/Reset Utilization Data	2	95
V	Verify LMK Store	3	115
VA	View Authorized Activities	4	137
VC	Verify the Contents of a Smartcard	4	204
VR	View Software Revision Number	2	81
VT	View LMK Table	3	119
WK	Translate a Zone PIN Key	4	225
XA	Add a RACC to the whitelist	5	242
XD	Decommission the HSM	5	243
XE	Remove RACC from the whitelist	5	244

Command	Function	Chapter	Page
XH	Commission the HSM	5	245
XI	Generate Customer Trust Anchor	5	246
XK	Make an RACC left or right key	5	247
XR	Commission a smartcard	5	248
XT	Transfer existing LMK to RLMK	5	249
XX	Decommission a smartcard	5	251
XY	HSM commissioning status	5	252
XZ	Duplicate CTA share	5	253
YA	Generate a CSCK	4	234
YB	Export a CSCK	4	235
Z	Encrypt a Clear ZMK Component	4	216
\$	Double-Length Key Calculator	4	210

Note: The following Console commands are no longer available and have been superseded by newer commands:

Console Command	Replaced by
DB Import a KML	IK Import Key
DF Import a BDK	IK Import Key
K Encrypt a Key Under LMK Variants 14-15	FK Form Key from Components
YC Import a CSCK	IK Import Key

List of Console Commands (Functional)

Function	Command	Chapter	Page
Configuration Commands			
Reset to Factory Settings	RESET	2	20
Configure Commands	CONFIGCMDS	2	22
Configure PIN Block Formats	CONFIGPB	2	24
Configure Security	CS	2	26
View Security Configuration	QS	2	36
Configure Console Port	CC	2	41
View Console Port Configuration	QC	2	43
Configure Host Port	CH	2	44
View Host Port Configuration	QH	2	50
Host Port Access Control list (ACL) Configuration	CONFIGACL	2	53
Configure Printer Port	CP	2	56
View Printer Port Configuration	QP	2	59
Configure Management Port	CM	2	60
View Management Port Configuration	QM	2	62
Configure Alarms	CL	2	63
View Alarm Configuration	QL	2	64
Add Static TCP/IP Route	ROUTE	2	65
View/Change Instantaneous Utilization Period	UTILCFG	2	68
Suspend/Resume Collection of Utilization Data	UTILENABLE	2	69
Suspend/Resume Collection of Health Check Counts	HEALTHENABLE	2	70
View SNMP Settings	SNMP	2	71
Add a SNMP Community or User	SNMPADD	2	72
Delete a SNMP Community or User	SNMPDEL	2	73
Fraud Detection Commands			
Configure Fraud Detection	A5	2	75
Re-enable PIN Verification	A7	2	76
Diagnostic Commands			
Diagnostic Test	DT	2	78
View Software Revision Number	VR	2	81
Show Network Statistics	NETSTAT	2	85
View Available Commands	GETCMDS	2	87
Test TCP/IP Network	PING	2	91
Trace TCP/IP route	TRACERT	2	93
View/Reset Utilization Data	UTILSTATS	2	95

Function	Command	Chapter	Page
View/Reset Health Check Counts	HEALTHSTATS	2	97
Check the FICON Host Interface	FICONTEST	2	98
LMK Commands			
Generate LMK Component	GK	3	103
Load LMK	LK	3	107
Load 'Old' LMKs into Key Change Storage	LO	3	111
Verify LMK Store	V	3	115
Duplicate LMK Component Sets	DC	3	116
Delete LMK	DM	3	117
Delete 'Old' LMK from Key Change Storage	DO	3	118
View LMK Table	VT	3	119
HSM Authorization			
Enter the Authorized State	A	4	123
Cancel the Authorized State	C	4	125
Authorize Activity	A	4	126
Cancel Authorized Activity	C	4	135
View Authorized Activities	VA	4	137
Logging Commands			
Display the Error Log	ERRLOG	4	140
Clear the Error Log	CLEARERR	4	141
Display the Audit Log	AUDITLOG	4	142
Clear the Audit Log	CLEARAUDIT	4	144
Audit Options	AUDITOPTIONS	4	145
Print the Audit Log	AUDITPRINT	4	149
Time and Date Commands			
Set the Time and Date	SETTIME	4	151
Query the Time and Date	GETTIME	4	152
Set Time for Automatic Self-Tests	ST	4	153
HSM Settings, Storage & Retrieval			
Save HSM Settings to a Smartcard	SS	4	155
Retrieve HSM Settings from a Smartcard	RS	4	156
Key Management Commands			
Generate Key Component	GC	4	160
Generate Key and Write Components to Smartcard	GS	4	163
Encrypt Clear Component	EC	4	167
Form Key from Components	FK	4	170
Generate Key	KG	4	176

Function	Command	Chapter	Page
Import Key	IK	4	180
Export Key	KE	4	183
Generate a Check Value	CK	4	186
Payment System Commands			
Generate a Card Verification Value	CV	4	189
Generate a VISA PIN Verification Value	PV	4	191
Load the Diebold Table	R	4	193
Encrypt Decimalization Table	ED	4	195
Translate Decimalization Table	TD	4	197
Generate a MAC on an IPB	MI	4	199
Smartcard Commands			
Format an HSM Smartcard	FC	4	201
Create an Authorizing Officer Smartcard	CO	4	203
Verify the Contents of a Smartcard	VC	4	204
Change a Smartcard PIN	NP	4	205
Read Unidentifiable Smartcard Details	RC	4	206
Eject a Smartcard	EJECT	4	207
DES Calculator Commands			
Single-Length Key Calculator	N	4	209
Double-Length Key Calculator	\$	4	210
Triple-Length Key Calculator	T	4	211
Legacy Commands			
Generate a ZMK Component	F	4	213
Generate a ZMK & Write to Smartcards	GZ	4	214
Encrypt a Clear ZMK Component	Z	4	216
Form a ZMK from Encrypted Components	D	4	217
Form a Key from Components	BK	4	219
Import a CVK or PVK	IV	4	221
Generate a Zone PIN Key	B	4	223
Translate a Zone PIN Key	WK	4	225
Generate a CVK Pair	KA	4	226
Translate a CVK Pair from LMK to ZMK	KB	4	227
Generate a Double-Length ZMK Component	DD	4	228
Form a ZMK from Clear Components	DE	4	229
Generate a BDK	DG	4	231
Generate & Export a KML	DA	4	233
Generate a CSCK	YA	4	234
Export a CSCK	YB	4	235

Function	Command	Chapter	Page
payShield Manager Commands			
Initialize	RI	5	237
Generate an HSM Certificate	RH	5	239
Backup Domain Authority Card	RZ	5	241
Add a RACC to the whitelist	XA	5	242
Decommission the HSM	XD	5	243
Remove RACC from the whitelist	XE	5	244
Commission the HSM	XH	5	245
Generate Customer Trust Anchor	XI	5	246
Make an RACC left or right key	XK	5	247
Commission a smartcard	XR	5	248
Transfer existing LMK to RLMK	XT	5	249
Decommission a smartcard	XX	5	251
HSM commissioning status	XY	5	252
Duplicate CTA share	XZ	5	253
Secure Host Communications			
Generate Certificate Signing Request	SG	6	255
Import Certificate	SI	6	258
Export HSM Certificate's Chain of Trust	SE	6	260
View Installed Certificate(s)	SV	6	262
Delete Installed Certificate(s)	SD	6	264
Generate HRK	SK	6	265
Change HRK Passphrase	SP	6	266
Restore HRK	SL	6	267

Note: The following Console commands are no longer available and have been superseded by newer commands:

Console Command	Replaced by
DB Import a KML	IK Import Key
DF Import a BDK	IK Import Key
K Encrypt a Key Under LMK Variants 14-15	FK Form Key from Components
YC Import a CSCK	IK Import Key

Chapter 2 – Configuration Commands

This chapter describes the commands used to configure a payShield 9000 HSM to work with the host system. It also includes those commands that provide information to assist with installation and configuration.

Configuration Commands

The payShield 9000 HSM provides the following console commands to support configuration operations:

Command	Page
Reset to Factory Settings (RESET)	20
Configure Commands (CONFIGCMDS)	22
Configure PIN Block Formats (CONFIGPB)	24
Configure Security (CS)	26
View Security Configuration (QS)	36
Configure Console Port (CC)	41
View Console Port Configuration (QC)	43
Configure Host Port (CH)	44
View Host Port Configuration (QH)	50
Configure Printer Port (CP)	56
Host Port Access Control list (ACL) Configuration (ACL)	53
View Printer Port Configuration (QP)	59
Configure Management Port (CM)	60
View Management Port Configuration (QM)	62
Configure Alarms (CL)	63
View Alarm Configuration (QL)	64
Add Static TCP/IP Route (ROUTE)	65
View/Change Instantaneous Utilization Period (UTILCFG)	68
Suspend/Resume Collection of Utilization Data (UTILENABLE)	69
Suspend/Resume Collection of Health Check Counts (HEALTHENABLE)	70
View SNMP Settings (SNMP)	71
Add a SNMP Community or User (SNMPADD)	72
Delete a SNMP Community or User (SNMPDEL)	73

Reset to Factory Settings

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **RESET**

Function: Returns the HSM to the state it was in when it was shipped from the factory, so that it can be securely taken out of service – e.g. for return to Thales for repair. Any configuration changes (including port settings) that the customer has applied will be reversed, and any customer data and logs will be erased. If the HSM is to be returned (e.g. after it has been repaired), a record of all the settings should be made before using this command such that the settings can be re-applied after the HSM's return. This command also reports whether the HSM is currently configured as it left the factory.

Authorization:

- Authorization is not required.
- The HSM must be in the secure state.

Inputs:

- Confirmation that Reset is required.

Outputs:

- Whether HSM is currently in its factory default state.
- Confirmation of Reset.

Notes:

- This utility cannot reset firmware or licenses installed on the HSM. Therefore after use of this facility, the HSM will still have the most recently installed firmware and license – which may be different from the firmware and license when the HSM was shipped from the factory.
- At the end of the reset process, the payShield 9000 will automatically perform a restart. If the console does not display correctly after this, the payShield 9000 should be restarted manually by using the "Restart" button on the front panel.

Example 1: Secure> **RESET** <Return>

Reset HSM to factory settings? [Y/N]: **Y** <Return>

The unit is currently in its factory default state: NO

Resetting the unit will remove all customer data,
including logs, port settings, keys, etc. This may cause
the console to stop functioning.

This operation should only be performed if this unit is being
taken out of normal operation.

Do you want to reset to the factory default settings? [Y/N]: **Y**
<Return>

You selected Yes; please confirm to Proceed with reset? [Y/N]:
Y <Return>

Return to factory default state complete

The HSM will now reboot automatically. Press any key to
continue: <Return>

Secure>

Example 2: Secure> **RESET** <Return>

Reset HSM to factory settings? [Y/N]: **Y** <Return>

The unit is currently in its factory default state: YES

Resetting the unit will remove all customer data,
including logs, port settings, keys, etc. This may cause
the console to stop functioning.

This operation should only be performed if this unit is being
taken out of normal operation.

Do you want to reset to the factory default settings? [Y/N]: **N**
<Return>

Secure>

Configure Commands

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CONFIGCMDS**

Function: To view the list of enabled host and console commands, and (if in secure state) to enable or disable host and console commands. All available commands are enabled by default.

Commands are enabled or disabled using the following syntax:

[+ or -] [C or H] [<Command Code>]

+ indicates that the specified command should be enabled.

- indicates that the specified command should be disabled.

C indicates that <Command Code> is a Console command.

H indicates that <Command Code> is a Host command.

<Command Code> is the command code to be enabled or disabled, and may contain the wildcard character '*'. If the first character is '*', then the second character is absent, and this matches all command codes of the specified type. If the second character is '*', then this matches all command codes of the specified type starting with the given first character.

Authorization: The HSM must be in the secure state to enable/disable host and console commands. The current status of enablement of host and console commands can be viewed in any state.

Inputs:

- List of host commands to enable.
- List of console commands to enable.
- List of host commands to disable.
- List of console commands to disable.

Outputs:

- Complete list of enabled host commands.
- Complete list of enabled console commands.

Errors: Invalid entry

Notes:

- When a disabled host command is invoked, error code 68 is returned.
- When a disabled console command is invoked, the message "Function not defined or not allowed" is displayed.

Example 1: *This example demonstrates the use of the **CONFIGCMDS** console command to view the list of enabled host and console commands.*

```
Online> CONFIGCMDS <Return>
```

```
List of enabled Console commands:
```

```
GC      GS      EC      FK
```

```
List of enabled Host commands:
```

```
A0 A4 GG GY
```

```
Online>
```

Example 2: *This example demonstrates the use of the **CONFIGMDS** console command to enable one console command (DE) and disable one host command (A4).*

```
Secure> CONFIGMDS <Return>

List of enabled Console commands:
GC      GS      EC      FK
List of enabled Host commands:
A0 A4 GG GY
Enter command code (e.g. +CDE) or Q to Quit: +CDE <Return>

List of enabled Console commands:
GC      GS      EC      FK      DE
List of enabled Host commands:
A0 A4 GG GY
Enter command code (e.g. +CDE) or Q to Quit: -HA4 <Return>

List of enabled Console commands:
GC      GS      EC      FK      DE
List of enabled Host commands:
A0 GG GY
Enter command code (e.g. +CDE) or Q to Quit: Q <Return>
Save COMMAND settings to smart card? [Y/N]: N <Return>

Secure>
```

Example 3: *This example demonstrates the use of the **CONFIGMDS** console command using the wildcard character '*' to disable all non-core host commands, and then enable just those host commands beginning with 'A'.*

```
Secure> CONFIGMDS <Return>

List of enabled Console commands:
GC      GS      EC      FK
List of enabled Host commands:
A0 A4 GG GY
Enter command code (e.g. +CDE) or Q to Quit: -H* <Return>

List of enabled Console commands:
GC      GS      EC      FK      DE
List of enabled Host commands:
Enter command code (e.g. +CDE) or Q to Quit: +HA* <Return>
List of enabled Console commands:
GC      GS      EC      FK      DE

List of enabled Host commands:
A0 A2 A4 A6 A8 AA AC AE AG AS AU AW AY
Enter command code (e.g. +CDE) or Q to Quit: Q <Return>
Save COMMAND settings to smart card? [Y/N]: Y <Return>

Insert card and press ENTER: <Return>
COMMAND settings saved to the smartcard.

Secure>
```

Configure PIN Block Formats

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CONFIGPB**

Function: To view the list of enabled PIN block formats, and (if in secure state) to enable or disable individual PIN block formats.
The default settings for the available PIN block formats are listed in Chapter 6 of the *payShield 9000 General Information Manual*.

Authorization: The HSM must be in the secure state to enable/disable PIN block formats. The current status of PIN Block format enablement can be viewed in any state.

Inputs: •PIN block format identifier.

Outputs: •List of enabled PIN block formats.

Errors: •Invalid entry

Example 1: *This example demonstrates the use of the **CONFIGPB** console command to view the list of enabled PIN block formats.*

```
Online> CONFIGPB <Return>

List of enabled PIN Block formats:
01 - ISO 9564-1 & ANSI X9.8 format 0
05 - ISO 9564-1 format 1
35 - MasterCard Pay Now & Pay Later format
41 - Visa/Amex new PIN only format
42 - Visa/Amex new & old PIN format
47 - ISO 9564-1 & ANSI X9.8 format 3

Online>
```

Example 2: *This example demonstrates the use of the **CONFIGPB** console command to enable the use of HSM PIN Block format 03.*

```
Secure> CONFIGPB <Return>

List of enabled PIN Block formats:
01 - ISO 9564-1 & ANSI X9.8 format 0
05 - ISO 9564-1 format 1
35 - MasterCard Pay Now & Pay Later format
41 - Visa/Amex new PIN only format
42 - Visa/Amex new & old PIN format
47 - ISO 9564-1 & ANSI X9.8 format 3

Enter + or - followed by PIN Block format or Q to Quit: +03 <Return>

List of enabled PIN Block formats:
01 - ISO 9564-1 & ANSI X9.8 format 0
03 - Diebold & IBM ATM format
05 - ISO 9564-1 format 1
35 - MasterCard Pay Now & Pay Later format
41 - Visa/Amex new PIN only format
42 - Visa/Amex new & old PIN format
47 - ISO 9564-1 & ANSI X9.8 format 3

Enter + or - followed by PIN Block format or Q to Quit: Q <Return>
Save PIN BLOCK settings to smart card? [Y/N]: Y <Return>
```

```
Insert card and press ENTER: <Return>  
PIN BLOCK settings saved to the smartcard.  
  
Secure>
```

Configure Security

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CS**

Function: To set the security configuration of the HSM and some processing parameters. CS converts all lower-case alpha values to upper case for display purposes, except for the Card issuer Password. Operation is menu-driven, as shown in the examples. The security settings can optionally be saved to a smartcard. The settings are described in Chapter 2 of the Security Operations manual, together with their default values.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- PIN length [4-12]: a one or two-digit number in the range 4 to 12.
 - This value is used by the HSM to define the length of encrypted PINs, symbolized as "L" in the Host Command manuals in the "Length & Type" column. The value of L is one more than the value entered for the PIN length in the CS command. Cleartext PINs (as entered into the BA host command) must have a length of L: shorter PINs can be entered, but must be padded to the right with hexadecimal F digits.
 - For example, if the PIN Length in CS has been set to 6 (i.e. L = 7), and the 4-digit PIN "1234" is to be entered into the BA host command, the value that is included in the command is "1234FFF".
 - All LMK-encrypted PINs will have a length of L.
 - Where a PIN is generated (e.g. JA host command) and the PIN length specified in the command is less than L, the generated PIN will be padded to the right with hexadecimal F characters to a length of L digits.
 - When an LMK-encrypted PIN is decrypted using the NG host command, any F-padding used to expand a shorter PIN is presented in the decrypted PIN and will need to be stripped off to derive the shorter PIN.
 - **Note:** Once the length is set, it cannot be easily altered. If it has to be changed to accommodate longer PINs, all the existing encrypted PINs will have to be translated. This requires two operations: the old PINs are first translated to encryption under, for example, a ZPK; the HSM is then re-configured for the longer PIN length; the PINs are then translated back from the ZPK to the LMK.
 - The above information applies to the following host commands: BA, BC, BE, BG, BQ, CE, CQ, DE, DG, EE, G2, G4, GA, GU, JA, JC, JE, JG, NG, PE, PG, QC, QK, QW, XK, XM, ZM.
- Echo [oN/off]: N or F

- Atalla ZMK variant support [oN/off]: N or F
- Transaction key scheme: Racal, Australian or None? [R/A/N]: R or A or N
 - NOTE: use of this setting may modify the functionality associated with some Host commands. See Chapter 12 of the *payShield 9000 General information Manual* for further information.
- User storage key length [S/D/T/V]: S, D, T, or V
- Display general information on payShield Manager Landing page? [Y/N]: Y or N
- Default LMK identifier [0-x]: Integer between 0 and x
- Management LMK identifier [0-x] : Integer between 0 and x
- Whether to erase the installed LMKs to enable the following settings to be changed.
- Enforce Atalla variant match to Thales key type [Y/N] (*Only displayed if Atalla ZMK variant support is set to ON.*) This relates to the IK Console command and A6 Host command: see the manual entries for these commands to understand the effect of this setting.
- Select clear PINs? [Y/N]: Y or N
- Enable ZMK translate command? [Y/N]: Y or N
- Enable X9.17 for import? [Y/N]: Y or N
- Enable X9.17 for export? [Y/N]: Y or N
- Solicitation batch size [1-1024]: a one to four-digit number, range 1 to 1024. [NOTE: small batch sizes must be avoided to prevent reference numbers and account numbers from being matched.]
- Enable Single-DES [Y/N]: Y or N
- Prevent Single-DES keys masquerading as double or triple-length key? [Y/N]: Y or N
- Single/double length ZMKs [S/D]: S or D (Single or Double)
- Decimalization table Encrypted/Plaintext [E/P]: E
- Enable decimalization table checks? [Y/N]: Y or N
- PIN encryption algorithm: A or B (Visa method or Racal Method)
- Whether to use the default Card Issuer password or to enter a different value (of 8 alphanumeric printable characters). [NOTE: this item should only be changed where customized HSM smartcards are being used. The original value must not be changed if standard Thales smartcards are in use.]
- Authorized State required when importing DES key under RSA key? [Y/N]: Y or N
- Minimum HMAC verification length in bytes [5-64]: number, range 5-64
- Enable PKCS#11 import and export for HMAC keys? [Y/N]: Y or N
- Enable ANSI X9.17 import and export for HMAC keys? [Y/N]: Y or N
- Enable ZEK/TEK encryption of ASCII data or Binary data or None? [A/B/N]: A or B or N
- Restrict Key Check Values to 6 hex chars? [Y/N]: Y or N
- Enable multiple authorized activities? [Y/N]: Y or N

- Enable 2DES LMK encryption of 3DES/2048-bit RSA keys
[Y/N] : Y or N (*If this is set to N, then a double-length Variant LMK cannot be used to protect triple-length keys / 2048-bit RSA keys, and so either a triple-length Variant LMK or a keyblock LMK must be used.*)
- Enable variable length PIN offset? [Y/N]: Y or N
- Enable weak PIN checking? [Y/N]: Y or N
- Enable PIN Block format 34 as output format for PIN translations to ZPK? [Y/N]: Y or N
- Enable translation of account number for LMK encrypted PINs [Y/N]: Y or N. (*This setting enables an account number to be changed whilst still retaining the original PIN. See host command QK.*)
- Use HSM clock for date/time validation? [Y/N]: Y or N
- Additional padding to disguise key length? [Y/N] : Y or N
- Key export and import in trusted format only? [Y/N] : Y or N
- Protect MULTOS Cipher Data Checksums? [Y/N] - *to control whether checksums generated over sensitive data will require encryption. (This is appropriate to MULTOS card data preparation using optional license LIC023.)*
- Card/password authorization [C/P]: C or P (Card or Password). *This setting determines whether security officers/operators authenticate themselves to the HSM and payShield Manager by using smartcards + PINs or passwords. See notes below.*
- Enable use of Tokens in PIN Translation? [Y/N]: Y or N
- Enable use of Tokens in PIN Verification? [Y/N]: Y or N
- Enforce key type 002 separation for PCI HSM compliance [Y/N]: Y or N. *See notes below.*
- Restrict PIN block usage for PCI HSM compliance? [Y/N]: Y or N. (*See notes below and Chapters 6 and 10 of the payShield 9000 General Information Manual.*)
- Enforce Authorization Time Limit? [Y/N] : Y or N. (*See notes below and Chapter 10 of the payShield 9000 General Information Manual.*)
- Enforce Multiple Key Components? [Y/N] : Y or N. (*See notes below and Chapter 10 of the payShield 9000 General Information Manual.*)
- Save SECURITY settings to smartcard? [Y/N]: Y or N

Outputs: • Prompts according to the settings chosen (see examples below).

Errors: Invalid Entry
 Card not formatted to save/retrieve HSM settings.
 Attempt with another card? [Y/N]

Notes: • For software versions which have been PCI HSM certified, in order to be PCI HSM compliant a number of security settings must have specific values as follows:

- Card/password authorization (local) – must be “C”
- Restrict PIN block usage for PCI HSM compliance – must be “Y”

- Enforce key type 002 separation for PCI HSM compliance – must be “Y”
 - Enforce Authorization Time Limit – must be “Y”
 - Enforce Multiple Key Components – must be “Y”
- See Chapters 6 and 10 of the *payShield 9000 General Information Manual* for further information.
- Once all of these settings are at the PCI HSM compliant value, they cannot be changed unless the RESET command is used.
 - If the value of the setting “Enforce key type 002 separation for PCI HSM compliance” is “Y”, then:
 - Key Type Table 2 (see General Information Manual Chapter 4) is in effect. If the setting has a value of “N”, then the HSM is not being operated in a PCI HSM compliant manner and Key Type Table 1 is in effect.
 - The following Host commands are disabled: AA, AE, FC, FE, FG, HC, KA, OE

Example 1: *Erasing LMKs not selected by the user*

```
Secure> CS <Return>
PIN Length [4-12]: 8 <Return>
Echo [oN/ofF]: N <Return>
Atalla ZMK variant support [oN/ofF]: F <Return>
Transaction Key Scheme: Racal, Australian or None [R/A/N]: N <Return>
User storage key length [S/D/T/V] (SINGLE): <Return>
Display general information on payShield Manager Landing page? [Y/N]:
N <Return>
Default LMK identifier [0-4] (0): <Return>
Management LMK identifier [0-4] (0): <Return>

LMKs must be erased before remaining parameters can be set.

Erase LMKs? [Y/N]: N <Return>

Save SECURITY settings to smartcard? [Y/N]: N <Return>
Secure>
```

Example 2: *Settings affecting PCI HSM compliance do not have compliant values. The user wishes to use the default card issuer password.*

```
Secure> CS <Return>

Please make a selection. The current setting is in parentheses.
Press ENTER to keep the current setting.
PIN length [4-12] (4): <Return>
Echo [oN/ofF] (ON): <Return>
Atalla ZMK variant support [oN/ofF] (ON): <Return>
Transaction key scheme: Racal, Australian or None? [R/A/N] (R): <Return>
User storage key length [S/D/T] (SINGLE): <Return>
Display general information on payShield Manager Landing page? [Y/N]: N
<Return>
Default LMK identifier [0-4] (0): <Return>
Management LMK identifier [0-4] (0): <Return>

LMKs must be erased before remaining parameters can be set

Erase LMKs? [Y/N]: Y <Return>

Enforce Atalla variant match to Thales key type? [Y/N] (NO): <Return>
Select clear PINs? [Y/N] (YES): <Return>
Enable ZMK translate command? [Y/N] (YES): <Return>
Enable X9.17 for import? [Y/N] (YES): <Return>
Enable X9.17 for export? [Y/N] (YES): <Return>
Solicitation batch size [1-1024] (5): <Return>

Enable Single-DES? [Y/N] (YES): <Return>
Prevent Single-DES keys masquerading
as double or triple-length key? [Y/N] (YES): <Return>
Single/double length ZMKs [S/D] (DOUBLE): d<Return>
Decimalization table Encrypted/Plaintext [E/P] (P): <Return>
Enable Decimalization Table Checks? [Y/N] (YES): <Return>

PIN encryption algorithm [A/B] (A): <Return>
Use default card issuer password [Y/N] (YES): <Return>
Authorized State required when importing DES key under RSA key?
[Y/N] (YES): <Return>
Minimum HMAC key length in bytes [5-64] (10): <Return>
Enable PKCS#11 import and export for HMAC keys [Y/N] (YES): <Return>
Enable ANSI X9.17 import and export for HMAC keys [Y/N] (YES): <Return>
Enable ZEK/TEK encryption of ASCII data or Binary data or None? [A/B/N]
(NONE): <Return>

Restrict Key Check Values to 6 hex chars [Y/N] (YES): <Return>
Enable multiple authorized activities [Y/N] (NO): <Return>
Allow persistent authorized activities [Y/N] (YES): <Return>
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys [Y/N] (YES): <Return>
Enable support for variable length PIN offset [Y/N] (NO): <Return>
Enable weak PIN checking [Y/N] (YES): <Return>
Enable PIN Block Format 34 as output format
for PIN Translations to ZPK [Y/N] (NO): <Return>
Enable translation of account number for LMK encrypted PINs [Y/N] (YES):
<Return>
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys [Y/N] (YES): <Return>

Use HSM clock for date/time validation? [Y/N] (YES): <Return>
Additional padding to disguise key length? [Y/N] (NO): <Return>
Key export and import in trusted format only? [Y/N] (NO): <Return>
Protect MULTOS Cipher Data Checksums? [Y/N] (YES): <Return>

Enable use of Tokens in PIN Translation? [Y/N] (NO): <Return>
Enable use of Tokens in PIN Verification? [Y/N] (NO): <Return>

The next 5 settings affect PCI HSM compliance - see Console Reference
Manual:

The following setting is not PCI HSM compliant:
```

Card/password authorization (local) [C/P] (P): C <Return>

The following setting is not PCI HSM compliant:

Restrict PIN block usage for PCI HSM compliance? [Y/N] (NO): N <Return>

Note that this setting is not PCI HSM compliant.

Confirm? [Y/N]: Y <Return>

The following setting is not PCI HSM compliant:

Enforce key type 002 separation for PCI HSM compliance? [Y/N] (NO):
<Return>

The following setting is not PCI HSM compliant:

Enforce Authorization Time Limit? [Y/N] (NO): <Return>

The following setting is not PCI HSM compliant:

Enforce Multiple Key Components? [Y/N] (NO): <Return>

Save SECURITY settings to smartcard? [Y/N]: N <Return>

Secure>

Example 3: *Final setting affecting PCI HSM compliance is about to be set to compliant value. The user is specifying a different card issuer software.*

```
Secure> CS <Return>

Please make a selection. The current setting is in parentheses.
Press ENTER to keep the current setting.
PIN length [4-12] (4): <Return>
Echo [oN/ofF] (ON): <Return>
Atalla ZMK variant support [oN/ofF] (ON): <Return>
Transaction key scheme: Racal, Australian or None? [R/A/N] (R): <Return>
User storage key length [S/D/T/V] (SINGLE): <Return>
Display general information on payShield Manager Landing Page? [Y/N] (Y):
<Return>
Default LMK identifier [0-4] (0): <Return>
Management LMK identifier [0-4] (0): <Return>

LMKs must be erased before remaining parameters can be set

Erase LMKs? [Y/N]: Y <Return>

Enforce Atalla variant match to Thales key type? [Y/N] (NO): <Return>
Select clear PINs? [Y/N] (YES): <Return>
Enable ZMK translate command? [Y/N] (YES): <Return>
Enable X9.17 for import? [Y/N] (YES): <Return>
Enable X9.17 for export? [Y/N] (YES): <Return>
Solicitation batch size [1-1024] (5): <Return>

Enable Single-DES? [Y/N] (YES): <Return>
Prevent Single-DES keys masquerading
as double or triple-length key? [Y/N] (YES): <Return>
Single/double length ZMKs [S/D] (DOUBLE): d<Return>
Decimalization table Encrypted/Plaintext [E/P] (P): <Return>
Enable Decimalization Table Checks? [Y/N] (YES): <Return>

PIN encryption algorithm [A/B] (A): <Return>
Use default card issuer password [Y/N] (YES): N<Return>
Enter card issuer password (local):***** <Return>
Password must be 8 characters.
Enter card issuer password (local):***** <Return>
Confirm card issuer password: *****<Return>
Authorized State required when importing DES key under RSA key?
[Y/N] (YES): <Return>
Minimum HMAC key length in bytes [5-64] (10): <Return>
Enable PKCS#11 import and export for HMAC keys [Y/N] (YES): <Return>
Enable ANSI X9.17 import and export for HMAC keys [Y/N] (YES): <Return>
Enable ZEK/TEK encryption of ASCII data or Binary data or None? [A/B/N]
(NONE): <Return>

Restrict Key Check Values to 6 hex chars [Y/N] (YES): <Return>
Enable multiple authorized activities [Y/N] (NO): <Return>
Allow persistent authorized activities [Y/N] (YES): <Return>
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys [Y/N] (YES): <Return>
Enable support for variable length PIN offset [Y/N] (NO): <Return>
Enable weak PIN checking [Y/N] (YES): <Return>
Enable PIN Block Format 34 as output format
for PIN Translations to ZPK [Y/N] (NO): <Return>
Enable translation of account number for LMK encrypted PINs [Y/N] (YES):
<Return>
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys [Y/N] (YES): <Return>

Use HSM clock for date/time validation? [Y/N] (YES): <Return>
Additional padding to disguise key length? [Y/N] (NO): <Return>
Key export and import in trusted format only? [Y/N] (NO): <Return>
Protect MULTOS Cipher Data Checksums? [Y/N] (YES): <Return>

Enable use of Tokens in PIN Translation? [Y/N] (NO): <Return>
Enable use of Tokens in PIN Verification? [Y/N] (NO): <Return>
```


The next 5 settings affect PCI HSM compliance - see Console Reference Manual:

The following setting is not PCI HSM compliant:

Card/password authorization (local) [C/P] (P): C<Return>

The following setting is not PCI HSM compliant:

Restrict PIN block usage for PCI HSM compliance? [Y/N] (NO): Y<Return>

The following setting is not PCI HSM compliant:

Enforce key type 002 separation for PCI HSM compliance? [Y/N] (NO):
Y<Return>

The following setting is not PCI HSM compliant<Return>:

Enforce Authorization Time Limit? [Y/N] (NO): Y

The following setting is not PCI HSM compliant:

Enforce Multiple Key Components? [Y/N] (NO): Y<Return>

These settings will all become PCI HSM compliant.

No further changes will be allowed to these options:

Card/password authorization = 'C'

Restrict PIN block usage = 'Y'

Enforce key type separation = 'Y'

Enforce Authorization Time Limit = 'Y'

Enforce Multiple Key Components = 'Y'

Confirm? [Y/N]: Y <Return>

Save SECURITY settings to smartcard? [Y/N]: Y <Return>

Insert card and press ENTER: <Return>

SECURITY settings saved to the smartcard.

Secure>

Example 4: *All settings affecting PCI HSM compliance have compliant values*

```
Secure> CS <Return>

Please make a selection. The current setting is in parentheses.
Press ENTER to keep the current setting.
PIN length [4-12] (4): <Return>
Echo [oN/off] (ON): <Return>
Atalla ZMK variant support [oN/off] (ON): <Return>
Transaction key scheme: Racal, Australian or None? [R/A/N] (R): <Return>
User storage key length [S/D/T/V] (SINGLE): <Return>
Display general information on payShield Manager Landing Page? [Y/N] (Y):
<Return>
Default LMK identifier [0-4] (0): <Return>
Management LMK identifier [0-4] (0): <Return>

LMKs must be erased before remaining parameters can be set

Erase LMKs? [Y/N]: Y <Return>

Enforce Atalla variant match to Thales key type? [Y/N] (NO): <Return>
Select clear PINs? [Y/N] (YES): <Return>
Enable ZMK translate command? [Y/N] (YES): <Return>
Enable X9.17 for import? [Y/N] (YES): <Return>
Enable X9.17 for export? [Y/N] (YES): <Return>
Solicitation batch size [1-1024] (5): <Return>

Enable Single-DES? [Y/N] (YES): N<Return>
Prevent Single-DES keys masquerading
as double or triple-length key? [Y/N] (NO): <Return>
Making default length for ZMKs: Double
Decimalization table Encrypted/Plaintext [E/P] (P): <Return>
Enable Decimalization Table Checks? [Y/N] (YES): <Return>

PIN encryption algorithm [A/B] (A): <Return>
Use default card issuer password [Y/N] (YES): <Return>
Authorized State required when importing DES key under RSA key?
[Y/N] (YES): <Return>
Minimum HMAC key length in bytes [5-64] (10): <Return>
Enable PKCS#11 import and export for HMAC keys [Y/N] (YES): <Return>
Enable ANSI X9.17 import and export for HMAC keys [Y/N] (YES): <Return>
Enable ZEK/TEK encryption of ASCII data or Binary data or None? [A/B/N]
(NONE): <Return>

Restrict Key Check Values to 6 hex chars [Y/N] (YES): <Return>
Enable multiple authorized activities [Y/N] (NO): <Return>
Allow persistent authorized activities [Y/N] (YES): <Return>
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys [Y/N] (YES): <Return>
Enable support for variable length PIN offset [Y/N] (NO): <Return>
Enable weak PIN checking [Y/N] (YES): <Return>
Enable PIN Block Format 34 as output format
for PIN Translations to ZPK [Y/N] (NO): <Return>
Enable translation of account number for LMK encrypted PINs [Y/N] (YES):
<Return>
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys [Y/N] (YES): <Return>

Use HSM clock for date/time validation? [Y/N] (YES): <Return>
Additional padding to disguise key length? [Y/N] (NO): <Return>
Key export and import in trusted format only? [Y/N] (NO): <Return>
*Protect MULTOS Cipher Data Checksums? [Y/N] (YES): <Return>

Enable use of Tokens in PIN Translation? [Y/N] (NO): <Return>
Enable use of Tokens in PIN Verification? [Y/N] (NO): <Return>

The following settings are all PCI HSM compliant and cannot be changed.
Card/password authorization (local): C
Restrict PIN block usage for PCI HSM Compliance: YES
Enforce key type separation for PCI HSM compliance: YES
Enforce Authorization Time Limit: YES
```

Enforce Multiple Key Components: YES

Save SECURITY settings to USB flash drive? [Y/N]: n
Save SECURITY settings to smartcard? [Y/N]: Y <Return>
Insert card and press ENTER: <Return>
SECURITY settings saved to the smartcard.

Secure>

View Security Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **QS**

Function: Reports the security configuration of the HSM and some processing parameters, plus the LMK check value.

Authorization: This command does not require any authorization.

Inputs: None.

Outputs: • See examples below.

Errors: None.

Notes:

- Where the software has been PCI HSM certified, in order to be PCI HSM compliant a number of security settings must have specific values as follows:
 - Card/password authorization (local) – must be “C”
 - Restrict PIN block usage for PCI compliance – must be “YES” (see Chapters 6 and 10 of the *payShield 9000 General Information Manual*)
 - Enforce key type 002 separation for PCI HSM compliance – must be “YES”

Once all of these settings are at the PCI HSM compliant value, they cannot be changed. See Chapter 10 of the *payShield 9000 General Information Manual* for further information.

Example 1: *Settings affecting PCI HSM compliance do not all have compliant values*

```
Online> qs <Return>

PIN length: 04
Encrypted PIN length: 05
Echo: ON
Atalla ZMK variant support: ON
Transaction key support: RACAL
User storage key length: SINGLE
Display general information on payShield Manager Landing Page:
YES
Default LMK identifier: 0
Management LMK identifier: 0

Enforce Atalla variant match to Thales key type: NO
Select clear PINs: YES
Enable ZMK translate command: YES
Enable X9.17 for import: YES
Enable X9.17 for export: YES
Solicitation batch size: 0005
Single-DES: ENABLED
Prevent Single-DES keys masquerading as double or triple-length
keys: YES
ZMK length: SINGLE
Decimalization tables: PLAINTEXT
Decimalization table checks: ENABLED
PIN encryption algorithm: A

Press "Enter" to view additional security settings... <Return>

Authorized State required when Importing DES key under RSA key:
YES
Minimum HMAC key length in bytes: 10
Enable PKCS#11 import and export for HMAC keys: YES
Enable ANSI X9.17 import and export for HMAC keys: YES
Enable ZEK/TEK encryption of ASCII data or Binary data or None:
NONE

Restrict Key Check Values to 6 hex chars: YES
Enable multiple authorized activities: NO
Allow persistent authorized activities: YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES
Enable variable length PIN offset: NO
Enable weak PIN checking: YES
Enable PIN Block Format 34 as output format for PIN
Translations to ZPK: NO
Enable translation of account number for LMK encrypted PINs:
YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES

Use HSM clock for date/time validation: YES
Additional padding to disguise key length: NO
Key export and import in trusted format only: NO
Protect MULTOS Cipher Data Checksums: YES

Enable use of Tokens in PIN Translation: NO
Enable use of Tokens in PIN Verification: NO

NOTE: The following settings are not all PCI HSM compliant.
Card/password authorization (local): P
Restrict PIN block usage for PCI HSM Compliance: NO
Enforce key type 002 separation for PCI HSM compliance: NO
Enforce Authorization Time Limit: YES
Enforce Multiple Key Components: YES

Online>
```

Example 2: *Settings affecting PCI HSM compliance have compliant values*

```
Online>qs

PIN length: 04
Encrypted PIN length: 05
Echo: ON
Atalla ZMK variant support: ON
Transaction key support: RACAL
User storage key length: SINGLE
Display general information on payShield Manager Landing Page:
YES
Default LMK identifier: 0
Management LMK identifier: 0

Enforce Atalla variant match to Thales key type: NO
Select clear PINs: YES
Enable ZMK translate command: YES
Enable X9.17 for import: YES
Enable X9.17 for export: YES
Solicitation batch size: 0005
Single-DES: ENABLED
Prevent Single-DES keys masquerading as double or triple-length
keys: YES
ZMK length: SINGLE
Decimalization tables: PLAINTEXT
Decimalization table checks: ENABLED
PIN encryption algorithm: A

Press "Enter" to view additional security settings... <Return>

Authorized State required when Importing DES key under RSA key:
YES
Minimum HMAC key length in bytes: 10
Enable PKCS#11 import and export for HMAC keys: YES
Enable ANSI X9.17 import and export for HMAC keys: YES
Enable ZEK/TEK encryption of ASCII data or Binary data or None:
NONE

Restrict Key Check Values to 6 hex chars: YES
Enable multiple authorized activities: NO
Allow persistent authorized activities: YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES
Enable variable length PIN offset: NO
Enable weak PIN checking: YES
Enable PIN Block Format 34 as output format for PIN
Translations to ZPK: NO
Enable translation of account number for LMK encrypted PINs:
YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES

Use HSM clock for date/time validation: YES
Additional padding to disguise key length: NO
Key export and import in trusted format only: NO
Protect MULTOS Cipher Data Checksums: YES

Enable use of Tokens in PIN Translation: NO
Enable use of Tokens in PIN Verification: NO

The following settings are all PCI HSM compliant and cannot be
changed.
Card/password authorization (local): C
Restrict PIN block usage for PCI HSM Compliance: YES
Enforce key type separation for PCI HSM compliance: YES
Enforce Authorization Time Limit: YES
Enforce Multiple Key Components: YES

Online>
```


Example 3: *Software has not been PCI HSM certified*

```
Online> QS <Return>

PIN length: 04
Encrypted PIN length: 05
Echo: ON
Atalla ZMK variant support: ON
Transaction key support: RACAL
User storage key length: SINGLE
Display general information on payShield Manager Landing Page:
YES
Default LMK identifier: 0
Management LMK identifier: 0

Select clear PINs: YES
Enable ZMK translate command: YES
Enable X9.17 for import: YES
Enable X9.17 for export: YES
Solicitation batch size: 0005
Single-DES: ENABLED
Prevent Single-DES keys masquerading as double or triple-length
keys: YES
ZMK length: SINGLE
Decimalization tables: PLAINTEXT
Decimalization table checks: ENABLED
PIN encryption algorithm: A

Press "Enter" to view additional security settings... <Return>

Authorized State required when Importing DES key under RSA key:
YES
Minimum HMAC key length in bytes: 10
Enable PKCS#11 import and export for HMAC keys: YES
Enable ANSI X9.17 import and export for HMAC keys: YES
Enable ZEK/TEK encryption of ASCII data or Binary data or None:
NONE

Restrict Key Check Values to 6 hex chars: YES
Enable multiple authorized activities: NO
Allow persistent authorized activities: YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES
Enable variable length PIN offset: NO
Enable weak PIN checking: YES
Enable PIN Block Format 34 as output format for PIN
Translations to ZPK: NO
Enable translation of account number for LMK encrypted PINs:
YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES

Use HSM clock for date/time validation: YES
Additional padding to disguise key length: NO
Key export and import in trusted format only: NO
Protect MULTOS Cipher Data Checksums: YES

Enable use of Tokens in PIN Translation: NO
Enable use of Tokens in PIN Verification: NO

The following settings are all PCI HSM compliant and cannot be
changed.
Card/password authorization (local): C
Restrict PIN block usage for PCI HSM Compliance: YES
Enforce key type separation for PCI HSM compliance: YES
Enforce Authorization Time Limit: YES
Enforce Multiple Key Components: YES

Online>
```


Configure Console Port

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CC**

Function: To set the baud rate and word format for the console port.
The new settings come into effect immediately after the command has completed.

Authorization: This command does not require any authorization.

Inputs:

- Console baud rate.
- Console word format.
- Console parity.
- Console flow control.

Outputs: None.

Errors: None.

Notes:

- The default settings for the console port are:
 - 9600 baud
 - 8 data bits
 - 1 stop bit
 - No parity
 - No flow control
- A USB port which has been configured for printer connection cannot be used for Console connection.

Example:

```
Offline> cc <Return>

Serial Port:

      BAUD RATES
1. 1200
2. 2400
3. 4800
4. 9600 (current value)
5. 19200
6. 38400
7. 57600
8. 115200
Console baud rate (enter for no change): <Return>

      DATA BITS
1. 5
2. 6
3. 7
4. 8 (current value)
Console data bits (enter for no change): <Return>

      STOP BITS
1. 1 (current value)
2. 2
Console stop bits (enter for no change): <Return>

      PARITY
1. none (current value)
2. odd
3. even
Console parity (enter for no change): <Return>

      FLOW Control
1. none (current value)
2. software
3. hardware
Console flow_ctl (enter for no change): <Return>

Serial Port will be configured as:
Baud: 9600
Word format: 8 bits, none parity, 1 stop
Flow control: none

Offline>
```

View Console Port Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **QC**

Function: To display details of the console port configuration of the HSM.

Authorization: This command does not require any authorization.

Inputs: None.

Outputs:

- The console baud rate.
- The console word format.
- The console flow control.

Errors: None.

Example:

```
Online> QC <Return>

Serial Port:
Baud: 9600
Word format: 8 bits, no parity, 1 stop
Flow control: none

Online>
```

Configure Host Port

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CH**

Function: To configure the Host port to emulate a type of data communications equipment and control equipment, which can be one of the following:

- Standard asynchronous emulation.
- Transparent asynchronous emulation.
- Ethernet.
- FICON

The Host port setting can optionally be saved to a smartcard. The new settings come into effect immediately after the command has completed. It is recommended that the Management Ethernet Port, Host 1 Ethernet Port, and Host 2 Ethernet Port are all on different IP subnets from each other.

Authorization :

- The HSM must be in the offline or secure state to run this command.
- If settings relating to Secure Host Communications (TLS/SSL) or Access Control Lists are to be changed, the payShield 9000 must be in Secure state.

Inputs:

- The options are menu driven and the inputs vary depending on the communication mode selected. See examples below.
- When configuring an asynchronous port to accept non-transparent traffic, you will be prompted for "*Terminating characters (4 hex):*". You can specify a sequence of 1 or 2 bytes. If only a single byte is required, the second byte should be set to 00. For example, if you wish to use the standard ETX character ('03' in hex) then you should enter "0300".
- Inputs specific to the FICON interface have the following definitions:
 - Control Unit Image:
 - Valid Range: 0-255; Default=0
 - This is the actual control unit image defined in the mainframe I/O gens.
 - Unit Address:
 - Valid Range: 0-255; Default=0
 - The unit address for this control unit.
 - Missing Interrupt Handler (mih) Minutes
 - Valid Range: 0-60; Default=0

This specifies the missing interrupt handler value to be used in the read device characteristics CCW for the mainframe. If set to 0, the mainframe setting is used.

Outputs: None.

Notes:

- To achieve maximum throughput on the HSM, the TCP/IP and FICON interfaces need to be driven with multiple connections (or threads). Optimum performance is normally achieved with 4 - 8 connections (depending on the HSM performance model and the commands being processed), although for FICON on the 1500 tps model the performance improves right up to the maximum of 16 connections. Running with only a single thread can significantly reduce the throughput of the HSM, and means that you will not be able to reach the rated throughput for the machine.
- A USB port which has been configured for printer connection cannot be used for Asynchronous Host Communications.
- It is recommended that the Management Ethernet Port, Host 1 Ethernet Port, and Host 2 Ethernet Port are all on different IP subnets from each other.
- Where dual Ethernet host ports are in use, 2 different IP addresses at the Host computer must be used to drive the 2 ports on the HSM.
- The ROUTE Console command can be used to set up static routes from the HSM's Host ports to a Host IP address on a different subnet from the HSM.
- The use of TLS or SSL (v3.0) with TCP is enabled from v2.2a of payShield 9000:
 - TLS/SSL traffic can be supported at the same time as non-TLS/SSL traffic.
 - The specified number of connections are shared between TLS/SSL and non-TLS/SSL traffic.
 - The HSM can be forced to accept only TLS/SSL traffic by setting the *UDP* and *TCP* options to "N".
 - The *TLS Enforced* option is used to ensure that the use of the weaker SSL 3.0 is blocked while still enabling the TLS options to be used. This has the effects of (a) disabling the SSL 3.0 protocol, and (b) of disabling those cipher suites which are designed for use with SSL, even if the TLS protocol is being used. See Chapter 14 of the *payShield 9000 General Information Manual*.

For Ethernet communications (not protected by TLS/SSL), a Well-Known Port Address is defined (default value 1500). See Chapter 1 of the Host Command Reference Manual for information on how port addresses can be used to select the LMK to be used with Host Commands.

If TLS/SSL is enabled, a Well-Known Port Address is also required (default value 2500). This works in the same way as the Well-Known Port Address for non-TLS/SSL traffic.

- The facility to apply ACLs (Access Control Lists) to Host Ports is available in software versions 2.3a and later.
- Host ports can be configured from software v2.3a to get IP addresses from a DHCP server and to support the use of network names.
- When upgrading from a version of payShield 9000 software that does not support Default Gateways (i.e. versions up to

1.3) to a version that does support Default Gateways (i.e. versions 1.4 onwards), a default value for the Default Gateway IP address will be provided by the software. If the IP address for the port that was previously set up was A.B.C.D, then the default value of the Default Gateway IP address will be A.B.C.1.

Errors: None.

Example 1: *In this example, Ethernet communications using TCP/IP and TLS are selected – all types of traffic are allowed. The IP addresses are set up as static, manually-entered addresses. Access Control Lists are to be used, and will be set up using the CONFIGACL console command. Secure state is required to change TLS/SSL or ACL settings.*

```
Secure> CH <Return>

Please make a selection. The current setting is in parentheses.
Message header length [1-255] (4): <Return>
Host interface [[A]sync, [E]thernet, [F]icon] (E): <Return>
Enter Well-Known-Port (1500): <Return>
Enter Well-Known-TLS-Port (2500): <Return>
UDP [Y/N] (Y): <Return>
TCP [Y/N] (Y): <Return>
TLS/SSL [Y/N] (Y): <Return>
TLS Enforced [Y/N] (N): <Return>
ACL Enabled [Y/N] (N): Y<Return>
Number of connections [1-64] (64): 5<Return>
Enter TCP keep alive timeout [1-120 minutes] (120): <Return>
Number of interfaces [1/2] (2): <Return>

Interface Number 1:

IP Configuration Method? [D]HCP or [S]tatic (DHCP): S<Return>
Enter IP Address (192.168.200.36): 192.168.200.100<Return>
Enter subnet mask (255.255.255.0): <Return>
Enter Default Gateway Address (192.168.200.3): <Return>

Enter speed setting for this port:

    SPEED OPTIONS:
    0 Autoselect
    1 10BaseT half-duplex
    2 10BaseT full-duplex
    3 100BaseTX half-duplex
    4 100BaseTX full-duplex
    5 1000BaseT half-duplex
    6 1000BaseT full-duplex

Speed setting (4): 6<Return>

Interface Number 2:

IP Configuration Method? [D]HCP or [S]tatic (DHCP): S<Return>
Enter IP Address (192.168.202.110): <Return>
Enter subnet mask (255.255.255.0): <Return>
Enter Default Gateway Address (192.168.202.3): <Return>

Enter speed setting for this port:

    SPEED OPTIONS:
    0 Autoselect
    1 10BaseT half-duplex
    2 10BaseT full-duplex
    3 100BaseTX half-duplex
    4 100BaseTX full-duplex
```

```
5 1000BaseT half-duplex
6 1000BaseT full-duplex
```

Speed setting (4): 6<Return>

Save HOST settings to smart card? [Y/N]: n<Return>

Secure>

Example 2: *In this example, Ethernet communications using TCP/IP and TLS are selected - but UDP, unprotected TCP, and SSL traffic is not allowed (i.e. all traffic must be TLS-protected). The IP addresses are set up as dynamic addresses to be obtained from a DHCP server. Access Control Lists are not being used. Only one host port is being configured. Secure state is required to change TLS/SSL or ACL settings.*

Secure> CH <Return>

Please make a selection. The current setting is in parentheses.

Message header length [1-255] (4): <Return>

Host interface [[A]sync, [E]thernet, [F]icon] (E): <Return>

Enter Well-Known-Port (1500): <Return>

Enter Well-Known-TLS-Port (2500): <Return>

UDP [Y/N] (Y): n<Return>

TCP [Y/N] (Y): n<Return>

TLS/SSL [Y/N] (Y): <Return>

TLS Enforced [Y/N] (N): y<Return>

ACL Enabled [Y/N] (N): n<Return>

Number of connections [1-64] (64): 5<Return>

Enter TCP keep alive timeout [1-120 minutes] (120): <Return>

Number of interfaces [1/2] (2): 1<Return>

Interface Number 1:

IP Configuration Method? [D]HCP or [S]tatic (static): D<Return>

Network Name (A4665275320Q-host1): HSM1-Host-1 <Return>

Enter speed setting for this port:

SPEED OPTIONS:

```
0 Autoselect
1 10BaseT half-duplex
2 10BaseT full-duplex
3 100BaseTX half-duplex
4 100BaseTX full-duplex
5 1000BaseT half-duplex
6 1000BaseT full-duplex
```

Speed setting (4): 6<Return>

Save HOST settings to smartcard? [Y/N]: n<Return>

Secure>

Example 3: *In this example, transparent asynchronous communications is enabled and the message header length is set to 6 characters. The Host baud is changed to 115200 bps and the word format is set to 8 data bits, no parity and 1 stop bit.*

```
Offline> CH <Return>
Please make a selection. The current setting is in
parentheses.
Message header length (1-255): 6 <Return>
Host interface [[A]sync, [E]thernet, [F]icon] (E): A <Return>
Transparent mode (Y/N): Y <Return>
* No interface device configured *
```

The following possible asynchronous interface devices were found in the system:

1. USB-Serial Controller by Prolific Technology Inc. located at Rear 3

Your selection (enter for no change): 1 <Return>

You must configure the serial parameters for this device:

BAUD RATES

1. 1200
2. 2400
3. 4800
4. 9600 (current value)
5. 19200
6. 38400
7. 57600
8. 115200

Device baud rate (enter for no change): 8 <Return>

DATA BITS

1. 5
2. 6
3. 7
4. 8 (current value)

Device data bits (enter for no change): <Return>

STOP BITS

1. 1 (current value)
2. 2

Device stop bits (enter for no change): <Return>

PARITY

1. none (current value)
2. odd
3. even

Device parity (enter for no change): <Return>

FLOW CONTROL

1. none (current value)
2. hardware
3. software

Host flow control (enter for no change): <Return>

Save HOST settings to smartcard? [Y/N]: Y <Return>

Insert card and press ENTER: <Return>

HOST settings saved to the smartcard.

Offline>

Example 4: *In this example, FICON communications is selected.*

```
Secure> CH <Return>
```

```
Please make a selection. The current setting is in parentheses.
```

```
Message header length [1-255] (4): 4 <Return>
```

```
Host interface [[A]sync, [E]thernet, [F]icon] (E): F <Return>
```

```
Control Unit Image [0-255] (0): <Return>
```

```
Unit address [0-255] (0): <Return>
```

```
Missing Interrupt Handler (mih) Minutes [0-60] (0): <Return>
```

```
Save HOST settings to smart card? [Y/N]: N <Return>
```

```
Secure>
```

View Host Port Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **QH**

Function: To display details of the Host port configuration of the HSM.

Authorization: This command does not require any authorization.

Inputs: None.

Outputs: For all systems:

- The message header length. This is the number of characters at the front of each command from the Host to the HSM (after the STX character). The HSM returns the message header in the response.
- The protocol used.

For an asynchronous system:

- The Host baud rate.
- The Host word format.
- The response delay. This is the delay before the HSM responds to the Host. It allows use of half-duplex Host communications that require a defined delay between the transmission of a command and the response from the HSM.

For an Ethernet system:

- The Well-Known Port. This is the publicized TCP Port address of the HSM.
- Transport method: TCP or UDP.
- Number of TCP connections. Each host interface supports this number of connections.
- The IP address for each host interface, and how they are derived. This is the Internet Protocol address of the HSM in the system.
- Subnet mask for each host interface. This is the subnet mask of the attached TCP/IP network. It is recommended that the Management Ethernet Port, Host 1 Ethernet Port, and Host 2 Ethernet Port are all on different IP subnets from each other.
- The port speed for each host interface.
- Whether Secure Host Communications is being used.
- Whether ACLs are being used.

Errors: None.

Example 1: *In this example, Ethernet communications using TCP/IP and TLS are selected – all types of traffic are allowed. The IP addresses are set up as static, manually-entered addresses. Access Control Lists are to be used, and will be set up using the CONFIGACL console command.*

```
Online> QH <Return>

Message header length: 04
Protocol: Ethernet
Well-Known-Port: 01500
Well-Known-TLS-Port: 02500
Transport: UDP TCP TLS/SSL, 64 connections
TCP Keep_Alive value (minutes): 120 minutes
ACL: Enabled
Number of interfaces : (2)

Interface Number: 1
IP Configuration Method: static
IP address: 192.168.200.036
Subnet mask: 255.255.255.000
Default Gateway: 192.168.200.003
MAC address: 00:d0:fa:04:27:62
Port speed: Ethernet autoselect (1000baseT full-duplex)

Interface Number: 2
IP Configuration Method: static
IP address: 192.168.202.110
Subnet mask: 255.255.255.0
Default Gateway: 192.168.202.3
MAC address: 00:d0:fa:04:27:63
Port speed: Ethernet autoselect (1000baseT full-duplex)

Online>
```

Example 2: *In this example, Ethernet communications using TCP/IP and TLS are selected - but UDP, unprotected TCP, and SSL traffic is not allowed (i.e. all traffic must be TLS-protected). The IP addresses are set up as dynamic addresses to be obtained from a DHCP server. Access Control Lists are not being used. Only one host port has been configured.*

```
Online> QH <Return>

Message header length: 04
Protocol: Ethernet
Well-Known-Port: 01500
Well-Known-TLS-Port: 02500
Transport: TLS, 64 connections
TCP Keep_Alive value (minutes): 120 minutes
ACL: Disabled
Number of interfaces : (1)

Interface Number: 1
IP Configuration Method: DHCP
Network Name: HSM1-Host-1
IP address: 192.168.200.036
Subnet mask: 255.255.255.000
Default Gateway: 192.168.200.003
MAC address: 00:d0:fa:04:3b:4a
Port speed: Ethernet autoselect (1000baseT full-duplex)

Online>
```

Example 3: *In this example, the host interface has been configured for transparent asynchronous communications.*

```
Online> QH <Return>

Message header length: 04
Protocol: Transparent Asynchronous
Terminating Sequence: 03 00
Response delay (ms): 00
Interface device: USB-Serial Controller by Prolific Technology
Inc. located at Rear 2 (ready)
Baud: 19200
Word format: 8 bits, 1 stop bit, no parity
Flow control: none

Online>
```

Example 4: *In this example, the host interface has been configured for FICON communications.*

```
Online> QH <Return>

Message header length: 04
Protocol: FICON
Control Unit Image: 0
Control Unit Address: 0
Missing Interrupt Handler (mih): 0 minutes

Online>
```

Host Port Access Control list (ACL) Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CONFIGACL**

Function: To display and amend the Access Control Lists (ACLs) for the HSM's host ports. When ACL checking is enabled using the CH console command, traffic from hosts is accepted only where the host's IP address is included in one of the ACL entries set up using this command.

Authorization: This command does not require any authorization. The HSM must be in Secure state.

Inputs:

- The user can view/add/delete entries. Entries cannot be amended.
- Each of the 2 host ports has its own ACL set.
- Entries can be of the following types:
 - A single IP address
 - An IP address range
 - An IP address mask
- Multiple types of entry can co-exist.
- Multiple entries of each type are allowed.
- The IP addresses in an entry can overlap with IP addresses in other entries.

Outputs: • Confirmations and errors only.

Errors: • IP address formats are validated.

Notes:

- This command sets up the IP addresses and ranges that will be used when checking traffic against the ACL, but the use of ACLs must be enabled in the CH console command before the ACLs configured in this command are applied.
- If the CH console command enables ACL checking but no ACL entries have been configured using CONFIGACL, then all host traffic will be blocked.
- ACLs apply only to Ethernet (including TLS/SSL) host traffic. They have no effect when asynchronous or FICON host communications are being used.

Example 1: *In this example, only one host interface has been configured in the CH command. There are no existing ACL entries. The user sets up a single address ACL entry, then adds a mask ACL entry, then adds a range ACL entry, and finally deletes the single address ACL entry.*

```
Secure>configacl<Return>

Access control list for Interface 1:
Single:
    None
Range:
    None
Mask:
    None

Add/Delete/Quit [A/D/Q]: A<Return>

Type - Single/Range/Mask [S/R/M]: S<Return>

IP Address: 10.10.41.10<Return>

Access control list for Interface 1:
Single:
    1) 10.10.41.10
Range:
    None
Mask:
    None

Add/Delete/Quit [A/D/Q]: A<Return>

Type - Single/Range/Mask [S/R/M]: M<Return>

Base IP Address: 10.10.40.0<Return>

Mask: 255.255.255.0<Return>

Access control list for Interface 1:
Single:
    1) 10.10.41.10
Range:
    None
Mask:
    2) 10.10.40.0 to 10.10.40.255 (Mask:255.255.255.0)

Add/Delete/Quit [A/D/Q]: A<Return>

Type - Single/Range/Mask [S/R/M]: R<Return>

From IP Address: 192.168.0.0<Return>

To IP Address: 192.168.0.92<Return>

Access control list for Interface 1:
Single:
    1) 10.10.41.10
Range:
    2) 192.168.0.0 to 192.168.0.92
Mask:
```

```
3) 10.10.40.0 to 10.10.40.255 (Mask:255.255.255.0)

Add/Delete/Quit [A/D/Q]: D<Return>
Entry to delete [1/3]: 1<Return>

Access control list for Interface 1:
Single:
    None
Range:
    1) 192.168.0.0 to 192.168.0.92
Mask:
    2) 10.10.40.0 to 10.10.40.255 (Mask:255.255.255.0)

Add/Delete/Quit [A/D/Q]: Q<Return>

Secure>
```

Example 2: *In this example, both host interfaces have been configured in the CH command. The user simply views the existing ACL for host interface 2, and then exits..*

```
Secure>configacl<Return>

Interface 1: 10.10.100.216
Interface 2: 10.10.101.216

Select Interface [1/2]: 2<Return>

Access control list for Interface 2:
Single:
    1) 10.10.40.22
    2) 10.10.40.23
    3) 10.10.40.23

Range:
    4) 10.10.40.200 to 10.10.40.220

Mask:
    None

WARNING: Duplicate - Single: Entries 2 and 3

Add/Delete/Quit [A/D/Q]: Q<Return>

Secure>
```

Configure Printer Port

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CP**

Function: To select and configure a connection to a printer attached to the HSM via a USB port. The HSM is compatible with most printers via its USB interfaces:

- A serial printer may be connected using a USB-to-serial converter cable available from Thales
- A parallel printer may be connected using a USB-to-parallel converter cable available from Thales

The new settings come into effect immediately after the command has completed.

Authorization: This command does not require any authorization.

Inputs:

- CR/LF order (standard or reversed): Y or N
- Selected printer connection.
- Setup Parameters, dependent on printer type.
- Whether to print a test page.

Outputs:

- Test page.

Errors: None.

Notes: A printer must be connected to the HSM before the CP command is invoked.
If a USB port is configured for a printer, it cannot subsequently be used for other purposes such as Console connection or Asynchronous Host Communications.

Example 1: *This example demonstrates the configuration of a printer attached to the HSM via a USB-to-serial cable.*

```
Offline> CP <Return>

Reverse the <LF><CR> order? [Y/N]: N <Return>

The following possible printer devices were found in the
system:
    0.  No printer
    1.  USB-Serial Controller by Prolific Technology Inc. located
        at Rear 4 (current selection)
Your selection (ENTER for no change): 1 <Return>
You must configure the serial parameters for this device:

    BAUD RATES
    1.  1200
    2.  2400
    3.  4800
    4.  9600 (current value)
    5.  19200
    6.  38400
    7.  57600
    8.  115200
Device baud rate (ENTER for no change): 8 <Return>

    DATA BITS
    1.  5
    2.  6
    3.  7
    4.  8 (current value)
Device data bits (ENTER for no change): <Return>

    STOP BITS
    1.  1 (current value)
    2.  2
Device stop bits (ENTER for no change): <Return>

    PARITY
    1.  none (current value)
    2.  odd
    3.  even
Device parity (ENTER for no change): <Return>

    Flow Control
    1.  none
    2.  software (current value)
    3.  hardware
Printer flow_ctl (ENTER for no change): <Return>

    Printer Offline Control
    1.  none (current value)
    2.  RTS
    3.  DTR
Printer offline control (ENTER for no change): <Return>
Timeout [in milliseconds, min=1000, max=86400000] (12000):
<Return>
Delay [in milliseconds, min = 0, max=7200000] (0): <Return>

Print test page? [Y/N]: N <Return>

Offline>
```

Example 2: *This example demonstrates the configuration of a printer attached to the HSM via a USB-to-parallel cable.*

```
Offline> CP <Return>

Reverse the <LF><CR> order? [Y/N]: N <Return>

The following possible printer devices were found in the
system:
    0. No printer
    1. USB2.0-Print by located at Rear 1
Your selection (enter for no change): 1 <Return>
Timeout [in milliseconds, min=1000, max=86400000] (1000):
1000<Return>
Delay [in milliseconds, min = 0, max=7200000] (0): <Return>
Print test page? [Y/N]: Y <Return>

Offline>
```

Example 3: *This example demonstrates the configuration of a native USB printer attached to the HSM.*

```
Offline> CP <Return>

Reverse the <LF><CR> order? [Y/N]: N <Return>

The following possible printer devices were found in the
system:
    0. No printer
    1. USB Printer by EPSON located at Front left (current
selection)
Your selection (ENTER for no change): 1 <Return>
Timeout [in milliseconds, min=1000, max=86400000] (1000):
1000<Return>
Delay [in milliseconds, min = 0, max=7200000] (0): <Return>
Print test page? [Y/N]: n<Return>

Offline>
```

View Printer Port Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **QP**

Function: To display details of the HSM's printer configuration.

Authorization: This command does not require any authorization.

Inputs: • Print test page: Y or N

Outputs: • <LF><CR> order reversed: YES or NO.
• Validation of current printer configuration.
• The serial configuration settings (serial printer only).

Errors: None.

Example 1: *This example demonstrates viewing the configuration of a printer attached to the HSM via a USB-to-serial cable.*

```
Online> QP <Return>
```

```
The configured printer, USB-Serial Controller by Prolific
Technology Inc. located at Rear 1, has been validated
```

```
BAUD RATE:      38400
DATA BITS:      8
STOP BITS:      1
PARITY:         none
Flow Control:   XON/XOFF
Offline Control: none
```

```
<LF><CR> order reversed: NO
```

```
Timeout: 12000 milliseconds
```

```
Delay: 0 milliseconds
```

```
Print test page? [Y/N]: N <Return>
```

```
Online>
```

Example 2: *This example demonstrates viewing the configuration of a printer attached to the HSM via a USB-to-parallel cable.*

```
Online> QP <Return>
```

```
The configured printer, USB2.0-Print by located at Rear 1, has
been validated.
```

```
<LF><CR> order reversed: NO
```

```
Timeout: 12000 milliseconds
```

```
Delay: 0 milliseconds
```

```
Print test page? [Y/N]: N <Return>
```

```
Online>
```

Configure Management Port

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CM**

Function: To configure the Management port, which is an Ethernet port used only for management of the HSM. If connection to the host is via Ethernet then the Ethernet host port is used for that purpose. The Management Ethernet port is used to update the HSM's internal software, updating licensing information, and for enabling management of a HSM via the payShield Manager.

The new settings come into effect immediately after the command has completed.

It is recommended that the Management Ethernet Port and Host Ethernet Ports are all on different IP subnets.

Authorization: The HSM must be in the offline or secure state to run this command.

Inputs:

- Whether IP address is manually or automatically derived.
 - If manually derived, then the address details must be entered.
- Enable (local or remote) payShield Manager connection?

Outputs: None.

Errors: None.

Notes:

- When upgrading from a version of payShield 9000 software that does not support Default Gateways (i.e. versions up to 1.3) to a version that does support Default Gateways (i.e. versions 1.4 onwards), a default value for the Default Gateway IP address will be provided by the software. If the IP address for the port that was previously set up was A.B.C.D, then the default value of the Default Gateway IP address will be A.B.C.1.

Example 1: *In this example, the management port has its IP address set up manually.*

```
Offline> CM <Return>
```

```
Management Ethernet Port:
```

```
IP Configuration Method? [D]HCP or [S]tatic (DHCP): s<Return>
```

```
Enter IP address (192.168.100.200): 192.168.200.90 <Return>
```

```
Enter subnet mask (255.255.255.0): <Return>
```

```
Enter Default Gateway Address (192.168.200.1): <Return>
```

```
Enter speed setting for this port:
```

```
SPEED OPTIONS:
```

- 0 Autoselect
- 1 10BaseT half-duplex
- 2 10BaseT full-duplex
- 3 100BaseTX half-duplex
- 4 100BaseTX full-duplex

```
5 1000BaseT half-duplex
6 1000BaseT full-duplex

Speed setting (4): 6 <Return>
```

```
Enable payShield Manager connection:
  Enable or Disabled? (E): D <Return>

Would you like to apply the changes now? [Y/N]: Y <Return>

Offline>
```

Example 2: *In this example, the management port has its IP address set up automatically by a DHCP server.*

```
Secure> CM <Return>

Management Ethernet Port:
IP Configuration Method? [D]HCP or [S]tatic (DHCP): <Return>
Network Name (B4665271226O-mgmt): HSM-Mngmnt <Return>

Enter speed setting for this port:

    SPEED OPTIONS:
    0 Autoselect
    1 10BaseT half-duplex
    2 10BaseT full-duplex
    3 100BaseTX half-duplex
    4 100BaseTX full-duplex
    5 1000BaseT half-duplex
    6 1000BaseT full-duplex

Speed setting (0): <Return>

Enable payShield Manager connection: <Return>
  Enable or Disabled? (E): <Return>

Would you like to apply the changes now? [Y/N]: Y <Return>

Secure>
```

View Management Port Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **QM**

Function: To display details of the Management port parameters.

Authorization: This command does not require any authorization.

Inputs: None.

Outputs:

- IP address.
- Subnet mask.
- Enable (local or remote) payShield Manager connection?

Errors: None.

Example 1:

```
Online> QM <Return>

Management Ethernet Port:
IP Configuration Method: static
IP address: 192.168.200.90
Subnet mask: 255.255.255.0
Default Gateway: 192.168.200.1
MAC address: 00:d0:fa:04:27:64
Port speed: Ethernet 1000baseT full-duplex

payShield Manager connection: Disabled

Online>
```

Example 2: *In this example, the management port has its IP address set up automatically by a DHCP server.*

```
Online> QM <Return>

Management ethernet port:
IP Configuration Method: DHCP
Network Name: HSM-Mngmnt
IP address: 192.168.1.3
Subnet mask: 255.255.255.0
Default Gateway: 192.168.1.1
MAC address: 00:d0:fa:04:27:64
Port speed: Ethernet autoselect (100baseTX full-duplex)

payShield Manager connection: Enabled

Online>
```

Configure Alarms

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CL**

Function: To enable or disable the motion alarm. The HSM alarm circuitry typically needs to be turned off if the HSM is to be moved. The alarm should be turned on while the HSM is in service or being stored. The alarm setting can optionally be saved to a smartcard. (*In software versions up to v2.1, the temperature alarm could also be turned on or off. From version 2.2 onwards the temperature alarm is permanently enabled.*)

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- Motion alarm status: Low, Medium, High or Off.
- Save settings to smartcard: Yes or No.

Outputs: None.

Errors:

- Card not formatted to save/retrieve HSM settings. Attempt with another card? [Y/N]

Example 1: *In this example, the setting is being made to a **less** secure setting.*

```
Secure> CL <Return>

Please make a selection. The current setting is in
parentheses.
Motion alarm [Low/Med/High/ofF] (MED): F<Return>
LMKs must be erased before proceeding.
Erase LMKs? [Y/N]: Y<Return>
Save ALARM settings to smart card? [Y/N]: N<Return>

Secure>
```

Example 2: *In this example, the setting is being made to a **more** secure setting.*

```
Secure> CL <Return>

Please make a selection. The current setting is in
parentheses.
Motion alarm [Low/Med/High/ofF] (OFF): H<Return>
Save ALARM settings to smart card? [Y/N]: n<Return>

Secure>
```

View Alarm Configuration

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **QL**

Function: To display details of the alarm configuration of the HSM.

Authorization: This command does not require any authorization.

Inputs: None.

Outputs:

- The Temperature alarm status.
- The Motion alarm status.

Errors: None.

Example:

```
Online> QL <Return>
Temperature alarm enabled
Motion alarm enabled high sensitivity
Online>
```


Add Static TCP/IP Route

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **ROUTE**

Function: To configure static routes for routing TCP/IP traffic.

Authorization: The HSM does not require any authorization to run this command.

Inputs: **Syntax:**

`[-f] [-n] [-q] [-v] command { [[modifiers] args] }`

Options:

-I *iface* The interface that ROUTE command is to be applied to. ***This is a required parameter.***

<u>iface Value</u>	<u>HSM Port</u>
h1	Host Port #1
h2	Host Port #2
m	Management Port

-f Remove all routes (as per flush). If used in conjunction with the add, change, delete, or get commands, route removes the routes before performing the command.

-n Don't print host and network names symbolically when reporting actions. (The process of translating between symbolic names and numerical equivalents can be quite time consuming, and may require correct operation of the network; thus it may be expedient to forgo this, especially when attempting to repair networking operations.)

-q Be quiet: suppress all output.

-v Be verbose: print additional details.

Command Options:

Add a route:

`add [-net|-host] destination gateway`

Change aspects of a route (such as its gateway):

`change [-net|-host] destination gateway`

Delete a specific route:

`delete [-net|-host] destination gateway`

(INET and INET6 only) Flush the routing tables of all gateway entries. If you want to delete only routes having destinations with addresses in a specified family, specify INET or INET6 as the *family* variable.

`flush [family]`

Look up and display the route for a destination:

`get [-net|-host] destination gateway`

Report changes to the routing information on a continuing basis:

`monitor`

Display route table (similar to netstat -r):

`show`

Specify the netmask to use when adding a network route:

netmask XXX.XXX.XXX.XXX

destination The destination host or network.

Gateway The next-hop gateway that packets should be addressed to.

If the keyword, default, or the network address, 0.0.0.0, is specified, then all packets sent to a remote network that's not defined in the routing tables, are sent to the specified gateway. Routes to a particular host are distinguished from those to a network by interpreting the Internet address associated with destination. Specifying the optional keywords -net and -host force the destination to be interpreted as a network or a host, respectively.

If the destination has a "local address part" of INADDR_ANY, or if the destination is the symbolic name of a network, then the route is assumed to be to a network; otherwise, the route is assumed to be to a host. For example:

This destination:	Is interpreted as:
128.32	-host 128.0.0.32
128.32.130	-host 128.32.0.130
-net 128.32	128.32.0.0
-net 128.32.130	128.32.130.0.

If the route is via an interface rather than via a gateway, you should specify the -interface modifier; the gateway given is the address of this host on the common network, indicating the interface to be used for transmission.

Option questions.

When certain command options are used, confirmation relating to route persistence is asked for.

Command Option	Requested confirmation
Add	Make route persistent? [Y/N]
Delete	Remove persistent route? [Y/N]
flush	Remove all persistent routes? [Y/N]

Modifiers

You can use the optional -netmask modifier to specify an additional address parameter that's interpreted as a network mask. You can use this like an OSI ESIS redirect with the netmask option, or to manually add subnet routes with netmasks different from that of the implied network interface (as would otherwise be communicated using the OSPF or ISIS routing protocols). After -netmask, enter the address parameter you want interpreted as the network mask.

You can override the implicit network mask generated in the INET case by placing this option after the destination parameter.

Similarly, you can use the -prefixlen modifier for IPv6.

The optional modifiers:

- -expire
- -hopcount
- -mtu
- -recvpipe
- -rtt
- -rttvar
- -sendpipe

- -ssthresh

provide initial values to metrics maintained in the routing entry. To lock any of these modifiers, precede the modifier with the -lock meta-modifier; you can also specify the -lockrest meta-modifier to lock all ensuing metrics.

Diagnostics

add [host | network] %s: gateway %s flags %x

The specified route is being added to the tables. The values printed are from the routing table entry supplied in the ioctl() call. If the gateway address used isn't the primary address of the gateway—the first one returned by gethostname() — the gateway address is printed numerically as well as symbolically.

delete [host &| network] %s: gateway %s flags %x

As above, but when deleting an entry.

%s %s done

A routing table entry is being deleted by the flush command.

Network is unreachable

An attempt to add a route failed because the gateway listed wasn't on a directly connected network. The next-hop gateway must be given.

not in table

A delete operation was attempted for an entry not present in the tables.

routing table overflow

An add operation was attempted, but the system was low on resources and couldn't allocate memory to create the new entry.

Permission denied

The attempted operation is privileged. Only root may modify the routing tables. These privileges are enforced by the kernel.

Outputs: Text messages as appropriate.

Notes: When upgrading from a version of payShield 9000 software that does not support Default Gateways (i.e. versions up to 1.3) to a version that does support Default Gateways (i.e. versions 1.4 onwards), any existing routes previously set up using the ROUTE command will be deleted. If it is required to continue using static routes (despite the availability of Default Gateways), these should be re-entered using the ROUTE command.

Example: Offline> ROUTE -I h1 add 20.20.20.0/24 10.10.10.1 <Return>

```
add net 20.20.20.0: gateway 10.10.10.1
Make route persistent? [Y/N]:
```

Offline>

View/Change Instantaneous Utilization Period

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **UTILCFG**

Function: To display the current setting of the period over which utilization statistics is to be collected when Instantaneous Utilization Data is requested. This command also allows the setting to be amended (in Offline/Secure states only).

Authorization: The HSM does not require any authorization to run this command.

Inputs: Amended value for Instantaneous Utilization Period. (It is suggested that the period should not be set to less than 10 seconds, as data collected over very short periods will not be indicative of actual activity.)

Outputs: Text messages as in example below.
Note that resetting of the value requires the HSM to be in Offline or Secure state.

Example:

```
Online> UTILCFG <Return>

Measurement period for instantaneous statistics is 60 seconds

Online>

...

Offline> UTILCFG <Return>

Measurement period for instantaneous statistics is 60 seconds

Change? [Y/N]: Y <Return>
Enter new value in seconds (1-60): 10 <Return>

Offline>
```

Suspend/Resume Collection of Utilization Data

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **UTILENABLE**

Function: To suspend or resume the collection of Utilization Data and the incrementing of the count of seconds over which the data is being collected. This allows data collection to be suspended if, for example, the HSM is taken out of service or temporarily re-purposed. It ensures that tps rates are not diluted by averaging command volumes over the total elapsed time, but only over the time that data is being collected

Authorization: The HSM does not require any authorization to run this command.

Inputs: Whether to change the current state.

Outputs: Text messages as in example below.

Notes: Following a software load, collection of Utilization Data will be suspended.
Data collection is automatically suspended while the HSM is not online.

Example: Online> UTILENABLE <Return>

```
Utilization statistics gathering is currently turned ON.  
Suspend? [Y/N] Y <Return>
```

Online> UTILENABLE <Return>

```
Utilization statistics gathering is currently turned OFF.  
Resume? [Y/N] Y <Return>
```

Offline>

Suspend/Resume Collection of Health Check Counts

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Not required		

Command: **HEALTHENABLE**

Function: To suspend or resume the collection of Health Check counts. This allows data collection to be suspended if, for example, data is not required.

Authorization: The HSM does not require any authorization to run this command.

Inputs: Whether to change the current state.

Outputs: Text messages as in example below.

Notes: Following a software load, the collection of Health Check counts will be disabled.

Example: Offline> HEALTHENABLE <Return>

```
Health check statistics gathering is currently turned ON.  
Suspend? [Y/N] Y <Return>
```

Offline> HEALTHENABLE <Return>

```
Health check statistics gathering is currently turned OFF.  
Resume? [Y/N] Y <Return>
```

Offline>

View SNMP Settings

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SNMP**

Function: To display the current SNMP settings, and to enable/disable provision of Utilisation and Health Check data via SNMP.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- Whether to Enable/Disable provision of Utilisation and Health Check data via SNMP.
- Which Ethernet port to use for SNMP traffic.

Outputs: Text messages as in example below. Information on Communities relates to SNMP versions 1 and 2; information on Users relates to SNMP version 3.

Notes: The HSM is delivered with no Users or Communities set up.

Example: Online> SNMP <Return>

V1/V2 Communities:

Read=public

Read=public

Read=private

V3 Users:

public: Authentication=none, Privacy=none

shades: Authentication=SHA, Privacy=DES

none: Authentication=none, Privacy=none

md5: Authentication=MD5, Privacy=none

SNMP is currently enabled

Disable? [Y/N]: n <Return>

SNMP is currently enabled on Host Port 2

Change SNMP port? [Y/N]: y <Return>

0. Host Port 1

1. Host Port 2

2. Management Port

SNMP port [0-2] (ENTER for no change): 1 <Return>

Online>

Add a SNMP Community or User

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SNMPADD**

Function: Add an SNMP Community (for SNMP versions 1 or 2) or User (for SNMP version 3).

Authorization:

- The HSM does not require any authorization to run this command.
- The HSM must be in Secure state.

Inputs:

- For an SNMP Community – the community name and security name (i.e. the community read strings).
- For an SNMP User – the user name, authentication algorithm, and privacy algorithm.

Outputs: Text messages as in example below.

Notes: The HSM is delivered with no Users or Communities set up.

Example:

```
Secure> SNMPADD <Return>
```

```
Add Community or User? [C/U]: C <Return>
```

```
Enter read string (Less than 20 characters): PUBLIC <Return>
```

```
The following entry will be added to the table
```

```
'Read=public.
```

```
Confirm? [Y/N]: Y <Return>
```

```
Community added successfully
```

```
Enter additional users or communities? [Y/N]: Y <Return>
```

```
Add Community or User? [C/U]: U <Return>
```

```
Enter user name: SHADES <Return>
```

```
Authentication algorithm [[N]one, [M]D5, [S]HA]: S <Return>
```

```
Enter authentication password: SHA <Return>
```

```
Privacy algorithm [[N]one, [D]ES]: D <Return>
```

```
Enter privacy password: DES <Return>
```

```
The following entry will be added to the table:
```

```
'shades: Authentication=SHA,Privacy=DES'.
```

```
Confirm? [Y/N]: Y <Return>
```

```
User added successfully
```

```
Enter additional users or communities? [Y/N]: N <Return>
```

```
Save and exit? [Y/N]: Y <Return>
```

```
SNMP configuration updated
```

```
Secure>
```


Delete a SNMP Community or User

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SNMPDEL**

Function: Delete an SNMP Community (for SNMP versions 1 or 2) or User (for SNMP version 3).

Authorization:

- The HSM does not require any authorization to run this command.
- The HSM must be in Secure state.

Inputs: The index of the community or user to be deleted.

Outputs: Text messages as in example below.

Notes: The HSM is delivered with no Users or Communities set up.

```
Secure> SNMPDEL <Return>

Remove Community or User? [C/U]: C <Return>
SNMP community table:
    0: Read=public
    1: Read=public
    2: Read=private

Select community to delete [0-2]: 1 <Return>

Community public/private deleted successfully

Remove additional users or communities? [Y/N]: Y <Return>

Remove Community or User? [C/U]: U <Return>
SNMP user table:
    0: User=public, Authentication=none, Privacy=none
    1: User=shades, Authentication=SHA, Privacy=DES
    2: User=none, Authentication=none, Privacy=none
    3: User=md5, Authentication=MD5, Privacy=none

Select user to delete [0-3]: 1 <Return>

User shades deleted successfully

Remove additional users or communities? [Y/N]: N <Return>

Save and exit? [Y/N]: Y <Return>

SNMP configuration updated

Secure>
```

Fraud Detection Commands

The payShield 9000 HSM provides the following console commands to support fraud detection operations:

Command	Page
Configure Fraud Detection (A5)	75
Re-enable PIN Verification (A7)	76

Configure Fraud Detection

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: May be required			
Activity: <u>audit.console</u>			

Command: **A5**

Function: To set the configuration of the HSM fraud detection function.

Authorization: If the Fraud Detection settings are to be edited, the HSM must be:

- in the offline or secure state to run this command, and
- either in the Authorized State, or the activity **audit.console** must be authorized, using the Authorizing Officer cards of the Management LMK.

Inputs:

- Whether and how to respond to Fraud Detection
- Limit on number of PIN verification failures per minute.
- Limit on number of PIN verification failures per hour.
- Limit on number of PIN attacks detected.

Outputs: None.

Errors:

- Not Authorized - the HSM is not authorized to perform this operation.
- Not Offline - the HSM must be offline to run this command.
- Invalid Entry - the value entered is invalid.

Notes:

- See the description of the Fraud Detection facility at Chapter 7 of the *payShield 9000 General Information Manual*.
- If any of the limits set by this command are exceeded, an entry will be made in the Audit Log.
- Setting the HSM reaction to Logging only and the limits to zero will result in Fraud Detection not being recorded in the Health Check data. (*The term "Logging" as used in the screen prompt refers to logging in the Health Check data, not in the Audit Log.*)

Example: Offline-AUTH> **A5** <Return>

HSM reaction to Exceeding Fraud Limits is : ON

The following limits are set:
 PIN verifies per minute: 100
 PIN verifies per hour: 1000
 PIN Attack Limit: 100

HSM reaction to Exceeding Fraud Limits? ([O]n/[L]ogging only): **L**
 <Return>

Note that logging is supported only if enabled via the HEALTHENABLE console command (or its payShield Manager equivalent)

Enter limit on PIN verifies per minute: **200** <Return>
 Enter limit on PIN verifies per hour: **2000** <Return>
 Enter PIN Attack Limit: **200** <Return>

Offline-AUTH>

Re-enable PIN Verification

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>audit.console</u>			

Command: **A7**

Function: To reset the configuration of the HSM fraud detection function.

Authorization: The HSM must be in the offline state to run this command. The HSM must be either in the Authorized State, or the activity **audit.console** must be authorized, using the Authorizing Officer cards of the Management LMK.

Inputs: None.

Outputs: None.

Errors:

- Not Authorized - the HSM is not authorized to perform this operation.
- Not Offline - the HSM must be offline to run this command.
- PIN Verification is not currently disabled

Example:

```
Offline-AUTH> A7 <Return>
PIN verification has been re-enabled
Offline-AUTH>
```

Diagnostic Commands

The payShield 9000 HSM provides the following console commands to support diagnostic operations:

Command	Page
Diagnostic Test (DT)	78
View Software Revision Number (VR)	81
View Available Commands (GETCMDS)	85
Show Network Statistics (NETSTAT)	87
Test TCP/IP Network (PING)	91
Trace TCP/IP route (TRACERT)	93
Add Static TCP/IP Route (ROUTE)	65
View/Reset Utilization Data (UTILSTATS)	95
View/Reset Health Check Counts (HEALTHSTATS)	97
Check the FICON Host Interface (FICONTEST)	98

Diagnostic Test

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **DT**

Function: To perform diagnostic tests.
The DT command tests the following parts of the HSM:

- Battery voltage level
- Various cryptographic algorithms (DES, AES, RSA, SHA-1, etc.)
- Working memory areas
- Power Supplies
- Random Number Generator
- Real-time clock
- Smartcard reader
- Operating temperature
- Operating fan speeds
- Operating voltages

The command also initiates the Health Check Instantaneous Status report as described in Chapter 9 of the *payShield 9000 General Information Manual*.

Authorization: The HSM does not require any authorization for this command.

Inputs: Optional qualifiers to modify scope and detail of output.
Options are:

all	run all the commands (default option)
verbose	be verbose in the output
battery	run the battery diagnostics
des	run the DES diagnostics
aes	run the AES diagnostics
health	run the health check diagnostics
md5	run the MD5 KAT
mem	run the memory diagnostics
psu	run the power supply diagnostics
rng	run the random number generator diagnostics
rsa	run the RSA KAT
rtc	run the real-time clock diagnostics
scr	run the smart card reader diagnostics
sha	run the SHA KAT
temp	run the temperature diagnostics
fans	run the fans diagnostics
volt	run the voltage diagnostics

Note that the multiple options can be combined (e.g. "dt temp verbose"; "dt volt rsa")

Note that whilst the command code ("dt") is not case sensitive, the options listed above are.

Outputs: Status report on each item.

Errors: None.

Notes:

- The diagnostics are run automatically on a daily basis at the time specified using the ST Console command.

Example 1: Offline> DT <Return>

```
Battery:           OK
AES:               OK
DES:               OK
MD5:               OK
Memory:            OK
Power Supply:      OK
RNG:               OK
RSA:               OK
Real-Time Clock:   OK
SHA:               OK
SCR:               OK
Temperature:       OK
Fans:              OK
Voltages:          OK
```

Health Check Status

```
TCP Server:        Up
UDP Server:        Up
Async Server:      Not Enabled
FICON Server:      Not Enabled
Local/Remote Manager Server: Up
Host Ethernet Link 1: Up
Host Ethernet Link 2: Not Enabled
Host Async Link:   Not Enabled
Host FICON Link:   Not Enabled
Unit Tampered?:    No
Fraud limits exceeded?: No
PIN attack limit exceeded?: No
```

Diagnostics complete

Offline>

Example 2: Online>DT verbose <Return>

Battery: OK
Voltage: 3050 mV
HSM will enter tamper state if voltage drops below 2500 mV

AES: OK
DES: OK
MD5: OK
Memory: OK
Power Supply: OK
RNG: OK
RSA: OK
Real-Time Clock: OK
Current Time: Fri Aug 10 09:32:27 2012

SHA: OK
SCR: OK
Temperature: OK
Inlet: 27 C (80 F)
Internal Device 1: 31 C (87 F)
Internal Device 2: 31 C (87 F)
Internal Device 3: 30 C (86 F)
Fans: OK
Fan 1: 3950 RPM
Fan 2: 4047 RPM
Voltages: OK

Expected	Actual	Deviation
3300 mV	3313 mV	0%
12000 mV	11925 mV	0%
5000 mV	5023 mV	0%
2500 mV	2507 mV	0%
1100 mV	1100 mV	0%

Health Check Status

TCP Server:	Up
UDP Server:	Up
Async Server:	Not Enabled
FICON Server:	Not Enabled
Local/Remote Manager Server:	Up
Host Ethernet Link 1:	Up
Host Ethernet Link 2:	Not Enabled
Host Async Link:	Not Enabled
Host FICON Link:	Not Enabled
Unit Tampered?:	No
Fraud limits exceeded?:	No
PIN attack limit exceeded?:	No

Diagnostics complete

Online>

View Software Revision Number

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **VR**

Function: To display details of the software release number, revision number and build number.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs: Software revision numbers.

Errors: None.

Notes: The software revision reported by the VR command will have one of the following forms:

- xxxx-19xx – this indicates that this software has been PCI HSM certified and that the appropriate security settings have been set (e.g. by using the CS Console command) to the required values.
- xxxx-09xx – this indicates that either:
 - this version of software is not PCI HSM certified, or
 - this version of software is PCI HSM certified but one or more of the appropriate security settings have been set (e.g. by using the CS Console command) to the required values.

Example 1: Some security settings are not PCI HSM compliant. FICON host interface is installed.

```
Online> VR <Return>

Base release: X.Xx
Revision:      XXXX-X9XX
Build Number: XXXX

PCI HSM Compliance: Some security settings are not PCI HSM
compliant

HSM Core API Version:      6.0.1

Serial Number:  C4665271228Q
Unit info:      Licenced

Host Configuration: Async,Ethernet,FICON
Licence Issue No:   1
Performance:       1500 TPS
Base Software:      Version 2
Ship Counter:       1
Crypto:             3DES,AES,RSA
LMKs Enabled:       5 LMKs

Press "Enter" to view additional information... <Return>

HSM9-LIC001 Base Software
HSM9-LIC013 5 LMKs
HSM9-LIC024 Mag Stripe Issuers
HSM9-LIC025 Mag Stripe Trans Processing
HSM9-LIC026 EMV Trans Processing
HSM9-LIC027 PIN/Key Mailer
HSM9-LIC028 Visa Cash Processing
HSM9-LIC029 Legacy Functions
HSM9-LIC030 Miscellaneous HSM8000 Base Commands License

Bootstrap Version:      1.10.2
Bootmanager Version:    1.16.12
LBC Version:            1.6
Microcontroller Version: 1.33

AGS Cryptographic Library:      1.10.C644
FIPS Validated DRBG/RNG Algorithm: TSPP-DRBG v1.1
FIPS Validated SHA Algorithm:     TSPP-SHA v1.0
FIPS Validated HMAC Algorithm:    TSPP-HMAC v1.0
FIPS Validated TDES Algorithm:    TSPP-TDES v1.0
FIPS Validated RSA Algorithm:     TSPP-RSA v1.0
FIPS Validated AES Algorithm:     TSPP-AES v1.0
FIPS Validated CMAC Algorithm:    TSPP-CMAC v1.0

Online>
```

Example 2: *All security settings compliant with PCI HSM:*

```
Online> VR <Return>

Base release: X.Xx
Revision:     XXXX-X9XX
Build Number: XXXX

PCI HSM Compliance: Refer to the PCI web site
(https://www.pcisecuritystandards.org/approved\_companies\_providers/approved\_pin\_transaction\_security.php) for current
certification status of this version of payShield 9000
software.

Security settings are consistent with the requirements of PCI
HSM.

HSM Core API Version:      6.0.1

Serial Number:  C4665271228Q
Unit info:      Licenced

Host Configuration: Async,Ethernet,FICON
Licence Issue No:   1
Performance:       1500 TPS
Base Software:      Version 2
Ship Counter:       1
Crypto:             3DES,AES,RSA
LMKs Enabled:       5 LMKs

Press "Enter" to view additional information... <Return>

HSM9-LIC001 Base Software
HSM9-LIC013 5 LMKs
HSM9-LIC024 Mag Stripe Issuers
HSM9-LIC025 Mag Stripe Trans Processing
HSM9-LIC026 EMV Trans Processing
HSM9-LIC027 PIN/Key Mailer
HSM9-LIC028 Visa Cash Processing
HSM9-LIC029 Legacy Functions
HSM9-LIC030 Miscellaneous HSM8000 Base Commands License

Bootstrap Version:      1.10.2
Bootmanager Version:    1.16.12
LBC Version:            1.6
Microcontroller Version: 1.33

AGS Cryptographic Library: 1.10.C644
FIPS Validated DRBG/RNG Algorithm: TSPP-DRBG v1.1
FIPS Validated SHA Algorithm:  TSPP-SHA v1.0
FIPS Validated HMAC Algorithm:  TSPP-HMAC v1.0
FIPS Validated TDES Algorithm:  TSPP-TDES v1.0
FIPS Validated RSA Algorithm:   TSPP-RSA v1.0
FIPS Validated AES Algorithm:   TSPP-AES v1.0
FIPS Validated CMAC Algorithm:  TSPP-CMAC v1.0

Online>
```

Example 3: *Software which has not been PCI HSM certified. TLS/SSL protection of host communications is enabled.*

```
Online> VR <Return>

Base release: X.XX
Revision:      XXXX-09XX
Build Number: XXXX

HSM Core API Version:      6.0.1

Serial Number:  A46652754970
Unit info:      Licenced

Host Configuration: Async,Ethernet,(optional) TLS/SSL
Licence Issue No:  1
Performance:      1500 TPS
Base Software:    Version 2
Ship Counter:     1
Crypto:           3DES,AES,RSA
LMKs Enabled:     5 LMKs

Press "Enter" to view additional information... <Return>

HSM9-LIC001 Base Software
HSM9-LIC013 5 LMKs
HSM9-LIC024 Mag Stripe Issuers
HSM9-LIC025 Mag Stripe Trans Processing
HSM9-LIC026 EMV Trans Processing
HSM9-LIC027 PIN/Key Mailer
HSM9-LIC028 Visa Cash Processing
HSM9-LIC029 Legacy Functions
HSM9-LIC030 Miscellaneous HSM8000 Base Commands License
HSM9-LIC036 Secure Host Comms

Bootstrap Version:      1.10.2
Bootmanager Version:   1.16.12
LBC Version:           1.6
Microcontroller Version: 1.33

AGS Cryptographic Library:      1.10.C644
FIPS Validated DRBG/RNG Algorithm: TSPP-DRBG v1.1
FIPS Validated SHA Algorithm:    TSPP-SHA v1.0
FIPS Validated HMAC Algorithm:   TSPP-HMAC v1.0
FIPS Validated TDES Algorithm:   TSPP-TDES v1.0
FIPS Validated RSA Algorithm:    TSPP-RSA v1.0
FIPS Validated AES Algorithm:    TSPP-AES v1.0
FIPS Validated CMAC Algorithm:   TSPP-CMAC v1.0

Online>
```

View Available Commands

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **GETCMDS**

Function: To display a list of available host & console commands. Commands listed in the output are licensed AND enabled. Commands omitted from the output are either not licensed, or not enabled. Console command CONFIGCMDS can be used to enable/disable individual commands.

GetCmds can optionally generate a hash (message digest) over the set of available commands, thus providing a simple mechanism to verify that two (or more) HSMs have the same set of commands available.

Note: Some of the commands listed may require additional license options enabled. For example the command EI requires the RSA algorithm to be included in the license in order to function correctly.

Authorization: The HSM does not require any authorization to run this command.

Inputs: [-lh]

-l Display available host & console commands.
-h Display hash over available commands.

Outputs: A list of available HSM commands (depending on options).

Errors: None.

Example:

Online> GETCMDS -h -1 <Return>

List of enabled Host commands:

A0	A2	A4	A6	A8	AA	AC	AE	AG	AI	AK	AM	AO	AS	AU
AW	AY	B0	B2	BA	BC	BE	BG	BI	BK	BM	BQ	BS	BU	BW
BY	C0	C2	C4	C6	C8	CA	CC	CE	CG	CI	CK	CM	CO	CQ
CS	CU	CW	CY	D0	D2	D4	D6	D8	DA	DC	DE	DG	DI	DK
DM	DO	DQ	DS	DU	DW	DY	E0	E2	E4	E6	E8	EA	EC	EE
EG	EI	EK	EM	EO	EQ	ES	EU	EW	EY	F0	F2	F4	F6	F8
FA	FC	FE	FG	FI	FK	FM	FO	FQ	FS	FU	FW	G0	G2	G4
G6	G8	GA	GC	GE	GG	GI	GK	GM	GO	GQ	GS	GU	GW	GY
H0	H2	H4	H6	H8	HA	HC	HE	HG	HI	HK	HM	HO	HQ	HS
HU	HW	I0	I2	I4	I6	I8	IA	IC	IE	J0	J2	J4	J6	J8
JA	JC	JE	JG	JI	JK	JO	JS	JU	K0	K2	K8	KA	KC	KE
KG	KI	KK	KM	KO	KQ	KS	KU	KW	KY	L0	LA	LC	LE	LG
LI	LK	LM	LO	LQ	LS	LU	LW	LY	M0	M2	M4	M6	M8	MA
MC	ME	MG	MI	MK	MM	MO	MQ	MS	MU	MW	MY	N0	NC	NE
NG	NI	NK	NO	NY	OA	OC	OE	OI	OK	OU	OW	P0	P2	P4
PA	PC	PE	PG	PI	PK	PM	PO	PQ	PS	PU	PW	PY	Q0	Q2
Q4	Q6	Q8	QA	QC	QI	QK	QM	QO	QQ	QS	QU	QW	QY	R2
R4	R6	R8	RA	RC	RE	RG	RI	RK	RM	RO	RQ	RS	RU	RW
RY	T0	T2	T4	T6	TA	U0	U2	U4	U6	U8	V0	V2	V4	V6
V8	W0	W2	W4	W6	W8	X0	X2	X4	X6	X8	XK	XM	XO	XQ
XS	XU	XW	Y0	Y2	Y4	Y6	Y8	Z0	ZA	ZE	ZK	ZM	ZU	

List of enabled Console commands:

A	A5	A6	A7	AUDITLOG	AUDITOPTIONS	
AUDITPRINT		B	BK	C	CC	CH
CK	CL	CLEARERR		CLEARAUDIT	CM	CO
CONFIGACL		CONFIGCMDS		CONFIGPB	CP	CS CV
D	DA	DC	DD	DE	DG	
DM	DO	DT	EA	EC	ED	
EJECT	ERRLOG	F	FC	FICONTEST	FK	
GC	GETCMDS	GETTIME	GK	GS	GZ	
HEALTHENABLE		HEALTHSTATS		IK	IV	KA KB
KE	KG	LK	LO	MI	N	
NETSTAT	NP	PING	PV	QC	QH	
QL	QM	QP	QS	R	RC	
RESET	RH	RI	ROUTE	RS	SD	
SE	SETTIME	SG	SI	SK	SL	
SP	SNMP	SNMPADD	SNMPDEL	SS	ST	
SV	SYSLOG	T	TD	TRACERT	UTILCFG	
UTILENABLE		UTILSTATS		V	VA	VC VR
VT	WK	XA	XD	XE	XH	
XI	XK	XR	XT	XX	XY	
XZ	YA	YB	Z	ZZ	\$	

Host/Console Command Hash Value: 3aee4c

Online>

Show Network Statistics

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **NETSTAT**

Function: The HSM records details about network activity on both its Management and Host Ethernet ports for diagnostic and security purposes. As a diagnostic aid, it can provide useful information when configuring the unit. If reviewed periodically, it can also provide evidence of unexpected network activity, which may require further investigation.

The HSM collects information about each 'endpoint' that communicates with it. The information recorded will depend on the particular protocol that was used to send the packet.

Authorization: The HSM does not require any authorization to run this command.

Inputs: **Syntax:**

Display a list of active sockets for each protocol:
netstat [-AanT] [-f address_family]

Display the contents of one of the other network data structures:
netstat [-dgiLmnrSTv] [-f address_family]

Continuously display (as per the wait interval) the information regarding packet traffic on the configured network interfaces:
netstat [-dnT] [-I interface] [-w wait]

Display statistics about the named protocol:
netstat [-T] [-p protocol]

Display per-interface statistics for the specified protocol:
netstat [-p protocol] [-iT] [-I interface]

Display per-interface statistics for the specified address family:
netstat [-sT] [-f address_family] [-i] [-I interface]

Options:

- A Show the addresses of any protocol control blocks associated with sockets.
- a Show the state of all sockets. Without -a, sockets used by server processes aren't shown.
- d Show the number of dropped packets.
- f *address_family* Limit the statistics or address control block reports to those of the specified address family.

Address family	address_family value
AF_INET	inet
AF_INET6	inet6
AF_LOCAL	local or unix
AF_ARP	arp
- g Show information related to multicast (group address) routing. By default, show the IP Multicast virtual-interface and routing tables. If -s is also specified, show the multicast routing statistics.
- I *interface* If used with -w, show information about the specified

interface only.

If used with `-f address_family` and `-s`, or with `-p protocol`, show per-interface statistics on the interface for *address_family* or protocol, respectively.

Interface	HSM Port
h1	Host Port #1
h2	Host Port #2
m	Management Port

If the `-I` option is not specified, netstat will report on all the interfaces.

- i Show the state of interfaces that have been auto-configured. Interfaces statically configured into a system but not located at boot time aren't shown. If you also specify `-a`, show multicast addresses currently in use for each Ethernet interface and for each IP interface address. Multicast addresses are shown on separate lines following the interface address with which they're associated. If used with `-f address_family` and `-s`, or with `-p protocol`, show per-interface statistics on the interface for *address_family* or protocol, respectively
- L Don't show link-level routes (e.g., IPv4 ARP or IPv6 neighbour cache).
- m Show statistics recorded by the memory-management routines (the network manages a private pool of memory buffers).
- n Show network addresses as numbers (normally netstat interprets addresses and attempts to display them symbolically).
- p *protocol* Show statistics about *protocol*, which is either a well-known name for a protocol or an alias for it. A null response typically means that there are no interesting numbers to report. The utility complains if *protocol* is unknown or if there's no statistics routine for it.
- r Show the routing tables. If `-s` is also specified, show the routing statistics instead.
- s Show per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.
- T Use TCP for name lookups (the default is UDP).
- v Show extra (verbose) detail for the routing tables (`-r`), or avoid truncating long addresses.
- w *wait* Specify the time interval for displaying network interface statistics.

Outputs: Text messages as appropriate.

The reported state can have the following values:

ESTABLISHED

The socket has an established connection.

SYN_SENT

The socket is actively attempting to establish a connection.

SYN_RECV

A connection request has been received from the network.

FIN_WAIT1

The socket is closed, and the connection is shutting down.

FIN_WAIT2

Connection is closed, and the socket is waiting for a shutdown from the remote end.

TIME_WAIT

The socket is waiting after close to handle packets still in the network.

CLOSED

The socket is not being used.

CLOSE_WAIT

The remote end has shut down, waiting for the socket to close.

LAST_ACK

The remote end has shut down, and the socket is closed. Waiting for acknowledgement.

LISTEN

The socket is listening for incoming connections.

CLOSING

Both sockets are shut down but we still don't have all our data sent.

UNKNOWN

The state of the socket is unknown

Example:

```
Offline> NETSTAT <Return>

Active Internet connections
Proto Recv-Q Send-Q Local Address           Foreign
Address      State
tcp          0      0 192.168.200.100.xserve  *.*
LISTEN
tcp          0      0 192.168.200.100.ftp    *.*
LISTEN
udp          0      0 *.*                     *.*
udp          0      0 *.syslog                *.*
udp          0      0 *.5002                   *.*
Offline>
```

Test TCP/IP Network

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **PING**

Function: To test the specified network node, and the route to it.

Authorization: The HSM does not require any authorization to run this command.

Inputs: **Syntax:**

```
ping [-adDfLnoPqQrRv] [-c count] [-E policy] [-g gateway]
    [-h host] [-I interface] [-i wait] [-l preload]
    [-p pattern] [-s packetsize] [-t tos] [-T ttl]
    [-w maxwait] host
```

Options:

-a	Emit an audible beep (by sending an ASCII BEL character to the terminal) after receiving each non-duplicate response.								
-c <i>count</i>	Stop after sending (and receiving) this many ECHO_RESPONSE packets.								
-D	Set the Don't Fragment bit in the IP header. This is meant to determine the path MTU.								
-d	Set the SO_DEBUG option on the socket being used.								
-E <i>policy</i>	Specify the IPsec policy for packets.								
-g <i>gateway</i>	Use Loose Source Routing to send the ECHO_REQUEST packets via gateway. The default is to use the routing table.								
-h <i>host</i>	Alternate way of specifying the target host instead of as the last argument.								
-I <i>interface</i>	The interface that PING is to be sent from. <table border="1"> <thead> <tr> <th>interface Value</th><th>HSM Port</th></tr> </thead> <tbody> <tr> <td>h1</td><td>Host Port #1</td></tr> <tr> <td>h2</td><td>Host Port #2</td></tr> <tr> <td>m</td><td>Management Port (default)</td></tr> </tbody> </table>	interface Value	HSM Port	h1	Host Port #1	h2	Host Port #2	m	Management Port (default)
interface Value	HSM Port								
h1	Host Port #1								
h2	Host Port #2								
m	Management Port (default)								
-i <i>interval</i>	Wait interval seconds between sending each packet (default is one second). For the -f option, the interval is 0.01 seconds.								
-l <i>preload</i>	Send this many packets as fast as possible before returning to normal behaviour.								
-L	Disable loopback when sending to multicast destinations, so the transmitting host doesn't see the ICMP requests.								
-n	Print numeric output only. No attempt is made to look up symbolic names for host addresses.								
-o	Exit successfully after receiving one reply packet.								
-P	Use a pseudo-random sequence for the data instead of the default, fixed sequence of incrementing 8-bit integers. This is useful to foil compression on PPP and other links.								
-p <i>pattern</i>	Fill out the packet with this many "padding" bytes (maximum is 16). You should find this useful for diagnosing data-dependent problems in a network. For example, -p ff causes the sent packet to be filled with ones.								
-Q	Don't display responses such as Network Unreachable ICMP messages concerning the ECHO_REQUESTs sent.								
-q	Be quiet: display nothing except for the summary lines at startup time and when finished.								
-R	Record the route.								
-r	Bypass the normal routing tables and send directly to a host on an attached network. If the host isn't on a directly attached network, an error is returned. You can use this option to ping a local host through an interface that has no route through it.								

<code>-s packetsize</code>	Send this many data bytes. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.
<code>-T ttl</code>	Use the specified time-to-live. It represents how many hops the packet can go through before being discarded (when it reaches 0). The default is 255.
<code>-t tos</code>	Use the specified hexadecimal type of service.
<code>-v</code>	Verbosity (default none).
<code>-w maxwait</code>	Specify a timeout, in seconds, before ping exits regardless of how many packets have been sent or received.

Outputs: Text messages as appropriate.

Example: Offline> PING -I h1 192.168.100.123 <Return>

```
PING 192.168.100.123 (192.168.100.123): 56 data bytes
64 bytes from 192.168.100.123: icmp_seq=0 ttl=32 time=16 ms
64 bytes from 192.168.100.123: icmp_seq=1 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=2 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=3 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=4 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=5 ttl=32 time=101 ms
64 bytes from 192.168.100.123: icmp_seq=6 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=7 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=8 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=9 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=10 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=11 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=12 ttl=32 time=4 ms
64 bytes from 192.168.100.123: icmp_seq=13 ttl=32 time=4 ms
```

Offline>

Trace TCP/IP route

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **TRACERT**

Function: To view the path taken from the HSM to the specified address.

Authorization: The HSM does not require any authorization to run this command.

Inputs: **Syntax:**

```
tracert [-DdFIInPrvx] [-a | -A as_server] [-f first_ttl]
        [-g gateway] [-i interface] [-m max_ttl] [-p port]
        [-q nqueries] [-s src_addr] [-t tos] [-w wait_time]
        host [packetsize]
```

Options:

-A <i>as_server</i>	Turn on AS lookups and use the given server instead of the default.												
-a	Turn on AS lookups for each hop encountered.												
-D	Dump the packet data to standard error before transmitting it.												
-d	Turn on socket-level debugging.												
-F	Set the "don't fragment" bit.												
-f <i>first_ttl</i>	Set the initial time-to-live used in the first outgoing probe packet.												
-g <i>gateway</i>	Specify a loose source route gateway (8 maximum).												
-I	Use ICMP ECHO instead of UDP datagrams.												
-i <i>interface</i>	<table> <thead> <tr> <th><u>interface</u></th> <th><u>Value</u></th> <th><u>HSM Port</u></th> </tr> </thead> <tbody> <tr> <td>h1</td> <td></td> <td>Host Port #1</td> </tr> <tr> <td>h2</td> <td></td> <td>Host Port #2</td> </tr> <tr> <td>m</td> <td></td> <td>Management Port (default)</td> </tr> </tbody> </table>	<u>interface</u>	<u>Value</u>	<u>HSM Port</u>	h1		Host Port #1	h2		Host Port #2	m		Management Port (default)
<u>interface</u>	<u>Value</u>	<u>HSM Port</u>											
h1		Host Port #1											
h2		Host Port #2											
m		Management Port (default)											
-l ("el")	Display the TTL (time-to-live) value of the returned packet. This is useful for checking for asymmetric routing.												
-m <i>max_ttl</i>	Set the maximum TTL (maximum number of hops) used in outgoing probe packets. The default is 30 hops (the same default as is used for TCP connections).												
-n	Print hop addresses numerically only. By default, addresses are printed both symbolically and numerically. This option saves a nameserver address-to-name lookup for each gateway found on the path.												
-P	Set the "don't fragment" bit and use the next hop MTU each time a "need fragmentation" error is received, thus probing the path MTU.												
-p <i>port</i>	The base UDP port number to be used in probes (default is 33434). The tracert utility hopes that nothing is listening on UDP ports base to base + nhops -1 at the destination host (so an ICMP PORT_UNREACHABLE message is returned to terminate the route tracing). If something is listening on a port in the default range, you can use this option to pick an unused port range.												
-q <i>nqueries</i>	The number of probes per ttl to <i>nqueries</i> (default is three probes).												
-r	Bypass the normal routing tables and send directly to a host on an attached network. If the host isn't on a directly attached network, an error is returned. You can use this option to "ping" a local host through an interface that has no route through it (for example, after the interface was												

	dropped by routed).
<code>-s <i>src_addr</i></code>	The IP address (must be given as an IP number, not a hostname) to be used as the source address in outgoing probe packets. If the host has more than one IP address, you can use this option to force the source address to be something other than the IP address of the interface that the probe packet is sent on. If the IP address you specify isn't one of this machine's interface addresses, an error is returned and nothing is sent.
<code>-t <i>tos</i></code>	The type-of-service (TOS) to be used in probe packets (default is zero). The value must be a decimal integer in the range 0 to 255. You can use this option to see if different TOSs result in different paths. Not all TOS values are legal or meaningful. You should find the values <code>-t 16</code> (low delay) and <code>-t 8</code> (high throughput) useful.
<code>-v</code>	Be verbose. Received ICMP packets other than TIME_EXCEEDED and UNREACHABLEs are listed.
<code>-w <i>wait_time</i></code>	The time (in seconds) to wait for a response to a probe (default is 5).
<code>-x</code>	Toggle checksums. Normally, this prevents traceroute from calculating checksums. In some cases, the operating system can overwrite parts of the outgoing packet but not recalculate the checksum (so in some cases the default is to not calculate checksums and using <code>-x</code> causes them to be calculated). Note that checksums are usually required for the last hop when using ICMP ECHO probes (<code>-I</code>).
<code>host</code>	The destination hostname or IP number.
<code>packetsize</code>	The probe datagram length (default is 40 bytes).

Outputs: Text messages as appropriate.

Example:

```
Offline> TRACERT -I h1 -g 10.10.10.1 10.10.11.2 <Return>

traceroute to 10.10.11.2 (10.10.11.2), 64 hops max, 40 byte packets
 1  10.10.10.1 (10.10.10.1)  5.000 ms  7.000 ms  5.000 ms
 2  10.10.11.2 (10.10.11.2)  5.000 ms  6.000 ms  6.000 ms

Offline>
```

View/Reset Utilization Data

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **UTILSTATS**

Function: To display Utilization Data at the Console. Options to print the data to an HSM-attached printer and to reset accumulated data to zero.

Authorization: The HSM does not require any authorization to run this command.

Notes:

- Utilization statistics are also reset when new software is installed on the HSM.
- The precise meaning of an HSM loading range identified below as, for example, "10-20%" is "from exactly 10% to just under 20%".
- Statistics are provided irrespective of which host interface the commands are received over.

Inputs:

- Whether to print output to HSM-attached printer
- Whether to Reset data

Outputs: Text messages as in example below.

Note that the number of seconds displayed is not necessarily the number of seconds between the start and end times: rather, it is the number of seconds during this period when data collection was enabled using the UTILENABLE command and the HSM was online.

Example: Online> UTILSTATS <Return>

```
HSM Serial Number:      A4665271570Q

Report Generation Time:  21-Mar-2011 23:23.05
Report Start Time:      01-JAN-2011 14:25.01
Report End Time:        05-MAR-2011 23:19.59
Total number of secs:   123,456
```

```
HSM Loading:
  0-10%:    56,789
 10-20%:    24,109
 20-30%:    21,445
 30-40%:    12,382
 40-50%:     3,288
 50-60%:     2,917
 60-70%:     2,123
 70-80%:     403
 80-90%:      0
 90-100%:    0
100%:       0
```

```
Press "Enter" to continue... <Return>
```

Host Command Volumes:

Cmd Code	Total Transactions	Average TPS
A0	225	4.79
A4	99	2.11
A6	342	7.28
A8	408	8.68
AA	141	3.00
AC	135	2.87
AE	84	1.79
AG	66	1.40
AS	18	0.38
AU	94	2.00
AW	94	2.00
AY	94	2.00
B0	50	1.06
BA	14	0.30
BC	34	0.72
BE	42	0.89
BG	5	0.11
BI	11	0.23
BK	128	2.72

Press "Enter" to continue... <Return>

Cmd Code	Total Transactions	Average TPS
BM	10	0.21
LA	2	0.04

Instantaneous HSM Load: 17%

Instantaneous Host Command Volumes:

Cmd Code	Total Transactions	Average TPS
BM	10	0.21
LA	2	0.04

Send output to printer? [Y/N]: Y <Return>

Reset All Stats? [Y/N]: Y <Return>

All Utilization statistics will be reset to 0. Confirm? [Y/N]: Y
<Return>

Online>

View/Reset Health Check Counts

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: May be required			
Activity: <u>diagnostics</u>			

Command: **HEALTHSTATS**

Function: To display Health Check counts at the Console. Options to print the data to an HSM-attached printer and to reset accumulated data to zero.

Authorization: The HSM does not require any authorization to run this command to view the data.
The HSM must be in Offline/Secure Authorized state (or the activity **diagnostics** must be authorized) for the Management LMK to reset the Health Check Counts

Notes:

- Accumulated health check counts are also reset when new software is installed on the HSM.
- If collection of health check data has been suspended at any time, the counts relating to Fraud Detection (i.e. failed PIN verifications and PIN Attacks) will not represent the values of those counts which will be used by the HSM to trigger return of Error 39 or deletion of LMKs.

Inputs:

- Whether to print output to HSM-attached printer
- Whether to Reset data (requires Offline/Secure Authorized state).

Outputs: Text messages as in example below.

Example: Offline-AUTH> **HEALTHSTATS** <Return>

HSM Serial Number: A4665271570Q

```
Report Generation Time:    21-Dec-2010 23:22.28
Report Start Time:        01-Dec-2010 01:11.21
Report End Time:          21-Dec-2010 23:22.28
Number of reboots:        3
Number of tamperers:      1
Failed PIN verifies/minute limit exceeded: 57
Failed PIN verifies/hour limit exceeded:    4
PIN Attack Limit exceeded: 0
```

Send output to printer? [Y/N]: **Y** <Return>

Reset All Stats? [Y/N]: **Y** <Return>

All Utilization statistics will be reset to 0. Confirm? [Y/N]: **Y** <Return>

Offline-AUTH>

Check the FICON Host Interface

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **FICONTEST**

Function: To check the operation of the FICON Host interface board (if fitted) and optical transceivers.

Authorization: The HSM does not require any authorization to run this command.

Notes:

- This test is appropriate only to payShield 9000 units fitted with the FICON option.
- The test can be run between 2 transceivers or on a single transceiver.
- A suitable FICON optical cable must be used to connect the two transceivers. Where 2 transceivers are being used, a standard FICON cable pair should be used to connect the transceivers.
- Where a single transceiver is being used, the loopback cable provided with the payShield 9000 should be used. Alternatively, one connection out of a standard FICON cable pair can be used.
- The test will send 10 packets and report success/failure on each.
- The test will check that the following components are installed and operational:
 - HSM main board
 - FICON board and connectors
 - Transceivers and connectors
 - Optical cable
 - FICON software

Inputs: • None

Outputs: Text messages as in example below.

Example:

Offline> **FICONTEST** <Return>

Please connect FICON Port 1 to FICON Port 2 or insert a loopback cable in FICON port 1 and press enter to continue: <Return>

Packet 1 success
Packet 2 success
Packet 3 success
Packet 4 success
Packet 5 success
Packet 6 success
Packet 7 success
Packet 8 success
Packet 9 success
Packet 10 success

terminating...
10 packets sent, 10 packets received, 0% loss

Offline>

Chapter 3 – Local Master Keys

Types of LMKs

A **Variant LMK** is a set of 20 double- or triple-length TDES keys, with different “pairs” and variants of those “pairs” being used to encrypt different types of keys. Note that the term “*pair*” is used regardless of whether the LMK consists of double-length keys, or triple-length keys. The standard LMK format supported in all previous versions of Thales (Racal) HSM firmware consists of 20 double-length TDES keys.

Note: The term “Variant LMK” refers to the fact that variants are applied to the LMK prior to using the LMK; a Variant LMK is not itself a variant of any other key.

A **Keyblock LMK** is either a triple-length TDES key, or a 256-bit AES key, and is used to encrypt keys in a keyblock format. A Keyblock LMK is not compatible with a Variant LMK, and it can only be used to encrypt keys in the keyblock format.

Note: The term “Keyblock LMK” refers to the ‘keyblock’ method of encrypting keys; a Keyblock LMK is not itself stored in the keyblock format.

Multiple LMKs

It is possible to install multiple LMKs within a single HSM. The precise details of the number and type of installed LMKs are controlled via the HSM’s license file:

License	Description
Default – no specific multi-LMK license	Two concurrent LMKs can be installed; however, one must be a Variant LMK, and the other a Keyblock LMK.
HSM9-LIC012 LMK x 2 (optional license)	Two concurrent LMKs can be installed; they can be any combination of Variant and Keyblock LMKs.
HSM9-LIC013 LMK x 5 (optional license)	Five concurrent LMKs can be installed; they can be any combination of Variant and Keyblock LMKs.
HSM9-LIC021 LMK x 10 (optional license)	Ten concurrent LMKs can be installed; they can be any combination of Variant and Keyblock LMKs.

See Chapter 1 of the Host Command Reference Manual for information on how the required LMK can be identified in Host commands.

LMK Table

LMKs are stored in a table within the secure memory of the HSM, with each LMK occupying a different 'slot' within the table. Each slot has the following attributes:

Attribute	Description
LMK ID	A 2-digit number which uniquely indicates the location of each LMK within the table. All references to LMKs are made by specifying the LMK Identifier.
Key Scheme	<ul style="list-style-type: none"> • "Variant" for traditional Racal/Thales LMK – key encryption performed using the <i>variant</i> method. • "Keyblock" for enhanced security – key encryption performed using the <i>keyblock</i> method.
Algorithm	<ul style="list-style-type: none"> • "3DES (2key)" or "3DES (3key)" is used by Variant LMKs. • "3DES (3key)" or "AES (256-bit)" is used by Keyblock LMKs. <p>Other algorithm types may be supported in future software releases.</p>
Status	<ul style="list-style-type: none"> • "Test" indicates that the LMK is used for testing purposes. • "Live" indicates that the LMK is used for live production purposes. <p>When installing LMKs, the HSM will prevent any mixing of Test and Live LMKs within the same slot (i.e. LMK Value and Old LMK Value must have the same status).</p>
Comments	User-entered text, which can be used to help identify LMKs.
Authorization	Indicates the authorization status of the HSM for this particular LMK – either a flag (for Authorized State) or a list of authorized activities.
LMK Check Value	The check value of the LMK.
Old LMK Check Value	The check value of the 'old' LMK (in Key Change Storage).

Use the console command VT (View LMK Table) to view the contents of the HSM's LMK table (but not the actual LMK values).

LMK Commands

The HSM provides the following console commands to support LMK operations:

Command	Page
Generate LMK Component (GK)	103
Load LMK (LK)	107
Load 'Old' LMKs into Key Change Storage (LO)	111
Verify LMK Store (V)	115
Duplicate LMK Component Sets (DC)	116
Delete LMK (DM)	117
Delete 'Old' LMK from Key Change Storage (DO)	118
View LMK Table (VT)	119

Generate LMK Component(s)

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **GK**

Function: To generate component(s) of an LMK, and store the component(s) on smartcards.
This command may be used to generate components for the following types of LMKs:

- Double-length (2DES) Variant LMK
- Triple-length (3DES) Variant LMK
- Triple-length (3DES) Keyblock LMK
- 256-bit AES Keyblock LMK.

When creating a Variant LMK or a 3DES Keyblock LMK, this command generates the data for a single LMK component card.

When creating an AES Keyblock LMK, this command generates the data for all the required number of LMK component cards.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- LMK Scheme (Variant or Keyblock).
- LMK Algorithm:
 - Double-length (2DES) or triple-length (3DES) if Variant scheme is selected
 - Triple-length (3DES) or AES if Keyblock scheme is selected.
- LMK Status (Test or Live).
- For TDES LMKs (Variant or Keyblock):
 - Component set number.
 - Three or four values (A, B, C, D).
 - For a double-length (2DES) Variant LMK, there are 3 secret values: A & B each consist of 16 hex digits, and C is 8 hex digits.
 - For a triple-length (3DES) Variant LMK, there are 4 secret values: A, B & C each consist of 16 hex digits, and D is 8 hex digits.
 - For a triple-length (3DES) Keyblock LMK, there are 3 secret values: A, B & C each consist of 16 hex digits.
 - The HSM generates random values if no values are input.
 - In the prompts for the secret values, a 16 hex digit values is referred to as "Secret Value" and an 8 hex digit value is referred to simply as "Value".
- For an AES Keyblock LMK:
 - Number of components.
 - Number of components required to reconstitute the LMK.

- Outputs:
- LMK components written to smartcards.
 - LMK component check value.
- Errors:
- Card not formatted – use the FC command to format the card.
 - Not a LMK card – the card is formatted for HSM settings or is a license card.
 - Warning – card not blank. Proceed? [Y/N] – LMK card is not blank.
 - Overwrite LMK set? [Y/N] – card already contains an LMK component.
 - Smartcard error; command/return: 0003 – invalid PIN is entered.
 - Invalid PIN; re-enter – a PIN of less than 4 or greater than 8 is entered.
- Notes:
- PINs must be entered within 60 seconds of being requested.
 - If the CS setting “Card/Password authorization” is set to “Card”, then the HSM will write a random password to the 1st and 2nd LMK component cards. These passwords will be required in order to put the HSM into the Authorized State.

Example 1:
(Triple-length
Variant LMK)

This example generates a triple-length Variant LMK component set, and writes the components to a smartcard.

```
Secure> GK <Return>
Variant scheme or key block scheme? [V/K]: V <Return>
Enter algorithm type [2=2DES, 3=3DES]: 3 <Return>

Key status? [L/T]: L <Return>
LMK component set [1-9]: 1 <Return>
Enter secret value A: AAAA AAAA AAAA AAAA <Return>
Enter secret value B: BBBB BBBB BBBB BBBB <Return>
Enter secret value C: CCCC CCCC CCCC CCCC <Return>
Enter value D: DDDD DDDD <Return>
Insert blank card and enter PIN: ***** <Return>
Writing keys...
Checking keys...
Device write complete, check: ZZZZZZ

Remove the smartcard and store it securely.

Make another copy? [Y/N]: N <Return>
1 copies made.

Repeat the procedure to generate further component sets.

Secure>
```


Example 2: (Double-length Variant LMK)

This example generates a double-length variant LMK component set, and writes the components to a smartcard.

```
Secure> GK <Return>
Variant scheme or key block scheme? [V/K]: V <Return>
Enter algorithm type [2=2DES, 3=3DES]: 2 <Return>

Key status? [L/T]: L <Return>
LMK component set [1-9]: 1 <Return>
Enter secret value A: AAAA AAAA AAAA AAAA <Return>
Enter secret value B: BBBB BBBB BBBB BBBB <Return>
Enter value C: CCCC CCCC <Return>
Insert blank card and enter PIN: ***** <Return>
Writing keys...
Checking keys...
Device write complete, check: ZZZZZZ
```

Remove the smartcard and store it securely.

```
Make another copy? [Y/N]: N <Return>
1 copies made.
```

Repeat the procedure to generate further component sets.

Secure>

Example 3: (Triple-length 3DES Keyblock LMK)

This example generates a 3DES keyblock LMK component, and writes the component to a smartcard.

```
Secure> GK <Return>
Variant scheme or key block scheme? [V/K]: K <Return>
Enter algorithm type [D=DES,A=AES]: D <Return>
Key status? [L/T]: L <Return>
LMK component set [1-9]: 1 <Return>
Enter secret value A: <Return>
Enter secret value B: <Return>
Enter secret value C: <Return>
Insert blank card and enter PIN: ***** <Return>
Writing keys...
Checking keys...
Device write complete, check: ZZZZZZ
```

Remove the smartcard and store it securely.

```
Make another copy? [Y/N]: N <Return>
1 copies made.
```

Repeat the procedure to generate further components.

Secure>

Example 4:
(AES Keyblock
LMK)

This example generates a set of AES keyblock LMK components, and writes each component to a smartcard.

```
Secure> GK <Return>
Variant scheme or key block scheme? [V/K]: K <Return>
Enter algorithm type [D=DES,A=AES]: A <Return>
Enter the number of components to generate: [2-9]: 5 <Return>
Enter the number of components required to reconstitute the
LMK: [2-5]: 2 <Return>
Key status? [L/T]: L <Return>
```

Check value for the LMK: ZZZZZZ

```
Insert blank card and enter PIN: ***** <Return>
Writing keys...
Checking keys...
Device write complete, check: ZZZZZZ
```

Remove the smartcard and store it securely.

```
Insert blank card and enter PIN: ***** <Return>
Writing keys...
Checking keys...
Device write complete, check: ZZZZZZ
```

Remove the smartcard and store it securely.

The above sequence is repeated to generate each component card.

Secure>

Load LMK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **LK**

Function: To load LMK components from smartcards.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- LMK Identifier: 2 numeric digits.
- Smartcards (RLMKs are supported) with LMK components.
- PINs for the Smartcards or passwords. The PIN must be entered within 60 seconds.
- Whether to make this LMK the Default/Management LMK - see Notes below.

Outputs:

- Individual LMK component check value(s).
- Final LMK check value.

Notes:

- For PCI HSM compliance, PINs and smartcards must be used to authenticate the Security Officers.
- Use of this command will always create an entry in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.
- If there is not already a Default and/or Management LMK installed (i.e. the LMK IDs identified in the security settings as being the default and management LMKs are empty), you will be asked if you wish to make this new LMK the Default/Management LMK.
- An error is returned if an attempt is made to load an LMK with a single component where:
 - The LMK is not a test LMK, and
 - The security setting to enforce multiple components has been set to YES.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Load failed check comparison - card is blank.
- Not a LMK card - card formatted for HSM settings or is a license card.
- Card not formatted - card is not formatted.
- Smartcard error; command/return: 0003 - invalid PIN is entered.
- Invalid PIN; re-enter - a PIN of less than 5 or greater than 8 digits is entered.
- Invalid key – a standard Thales test key cannot be given live status.
- Incompatible key status – the components have different status ("live" or "test").
- Invalid key - Multiple key components required – an attempt has been made to load an LMK (other than a test LMK) using a single component when the security setting to enforce multiple components has been set to YES.

**Example 1:
(Double-length
Variant LMK)**

This example loads a double-length Variant LMK from smartcards and installs it in the HSM. There is already Default and Management LMKs installed.

```
Secure> LK <Return>
Enter LMK id: 00 <Return>
Enter comments: Live LMK for ABC Bank <Return>
LMK in selected location must be erased before proceeding
Erase LMK? Y <Return>
Load LMK from components
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: Y <Return>
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure. When all components have been loaded and the HSM displays the LMK Check value, record the check value.

```
LMK Check: ZZZZZZ
LMK id: 00
LMK key scheme: Variant
LMK algorithm: 3DES (2key)
LMK status: Live
Comments: Live LMK for ABC Bank
Confirm details? [Y/N]: Y <Return>
Use the LO command to load LMKs into key change storage.
Secure>
```

**Example 2:
(Triple-length
Variant LMK)**

This example loads a triple-length variant LMK from smartcards and installs it in the HSM. There are already Default and Management LMKs installed.

```
Secure> LK <Return>
Enter LMK id: 01 <Return>
Enter comments: Process System One <Return>
LMK in selected location must be erased before proceeding
Erase LMK? Y <Return>
Load LMK from components
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: Y <Return>
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure. When all components have been loaded and the HSM displays the LMK Check value, record the check value.

```
LMK Check: ZZZZZZ
LMK id: 01
LMK key scheme: Variant
LMK algorithm: 3DES (3key)
LMK status: Live
Comments: Process System One
Confirm details? [Y/N]: Y <Return>
Use the LO command to load LMKs into key change storage.
Secure>
```

Example 3:
(Any LMK type)

In this example, the PIN is not entered within 60 seconds.

```
Secure> LK <Return>
Enter LMK id [0-9]: 0 <Return>
Enter comments: <Return>
Load LMK from components
Insert card and enter PIN:
Terminated
Secure>
```

Example 4:
(Double- or
triple-length
Variant LMK)

In this example, the security setting requiring use of multiple components has been set to YES, but the user has attempted to load a non-Test LMK using only one component.

```
Secure> LK <Return>
Enter LMK id [0-4]: 0 <Return>
Enter comments: <Return>
Load LMK from components
Insert card and enter PIN: *****<Return>
Check: 562342
Load more components? [Y/N]: n<Return>
LMK Check: 562342
Invalid key - Multiple key components required
Secure>
```

Example 5:
(3DES Keyblock
LMK)

This example loads a 3DES keyblock LMK from smartcards and installs it in the HSM. There is already Default and Management LMKs installed.

```
Secure> LK <Return>
Enter LMK id: 01 <Return>
Enter comments: Live LMK for XYZ Bank <Return>
LMK in selected location must be erased before proceeding
Erase LMK? Y <Return>
Load LMK from components
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: Y <Return>
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure. When all components have been loaded and the HSM displays the LMK Check value, record the check value.

```
LMK Check: ZZZZZZ
LMK id: 01
LMK key scheme: KeyBlock
LMK algorithm: 3DES (3key)
LMK status: Live
Comments: Live LMK for XYZ Bank
Confirm details? [Y/N]: Y <Return>
Use the LO command to load LMKs into key change storage.
Secure>
```

**Example 6:
(AES Keyblock
LMK)**

This example loads an AES keyblock LMK from smartcards and installs it in the HSM. There is already Default and Management LMKs installed.

```
Secure> LK <Return>
Enter LMK id: 02 <Return>
Enter comments: Live LMK for XYZ Bank <Return>
LMK in selected location must be erased before proceeding
Erase LMK? Y <Return>
Load LMK from components
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure. When all components have been loaded and the HSM displays the LMK Check value, record the check value.

```
LMK Check: ZZZZZZ
LMK id: 02
LMK key scheme: KeyBlock
LMK algorithm: AES-256
LMK status: Live
Comments: Live LMK for XYZ Bank
Confirm details? [Y/N]: Y <Return>
Use the LO command to load LMKs into key change storage.
Secure>
```

**Example 7:
(AES Keyblock
LMK - no
Default or
Management
LMK already
installed.)**

This example loads an AES keyblock LMK from smartcards and installs it in the HSM. There is no Default or Management LMK already installed.

```
Secure> LK <Return>
Enter LMK id: 02 <Return>
Enter comments: Live LMK for XYZ Bank <Return>
LMK in selected location must be erased before proceeding
Erase LMK? Y <Return>
Load LMK from components
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure. When all components have been loaded and the HSM displays the LMK Check value, record the check value.

```
LMK Check: ZZZZZZ
LMK id: 02
LMK key scheme: KeyBlock
LMK algorithm: AES-256
LMK status: Live
Comments: Live LMK for XYZ Bank
Confirm details? [Y/N]: Y <Return>
Use the LO command to load LMKs into key change storage.
Do you want to make this LMK the default LMK? [Y/N]: Y <Return>
Do you want to make this LMK the management LMK? [Y/N]: Y
<Return>
Secure>
```

Load 'Old' LMKs into Key Change Storage

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>admin.console</u>			

Command: **LO**

Function: To load an old LMK component set into Key Change Storage for use in translations from old to new keys. Note that the current LMK must be installed before an "old" LMK can be installed. Also note that it is possible to install a Variant LMK as the "old" LMK, and with a Keyblock LMK as the "new" LMK.

Authorization: The HSM must be in the secure state to run this command. Additionally, the HSM must be either in the Authorized State, or the activity **admin.console** must be authorized, using the Authorizing Officer cards of the specified LMK.

Inputs:

- LMK identifier: 2 numeric digits.
- Smartcards (RLMKs are supported) with old LMK components.
- PINs for the Smartcards or passwords. PINs must be entered within 60 seconds of being requested.

Outputs:

- Individual LMK Component check value(s).
- Final LMK key check value.

Errors:

- No LMK loaded – there is no LMK loaded in main memory.
- Invalid LMK identifier – entered identifier out of range
- Key block LMK not permitted – it is not permitted to load a key block LMK into key change storage if a variant LMK is loaded in main memory.
- Load failed check comparison – card is blank.
- Not a LMK card – card formatted for HSM settings or is a license card.
- Card not formatted – card is not formatted.
- Smartcard error; command/return: 0003 – invalid PIN is entered.
- Invalid PIN; re-enter – a PIN of less than 4 or greater than 8 is entered.
- Command only allowed from Secure-Authorized – the HSM is not in Secure State, or the HSM is not authorized to perform this operation, or both.
- Invalid key – a standard Thales test key cannot be given live status.
- Incompatible cards – the component cards have different formats.
- Incompatible key status – the components have different status ("live" or "test").
- Invalid key - Multiple key components required – an attempt has been made to load an LMK (other than a Test LMK) using a single component when the security setting to enforce multiple components has been set to YES.

Notes:

- For PCI HSM compliance, PINs and smartcards must be used to authenticate the Security Officers.
- Use of this command will always create an entry in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.
- It is not permitted to load a Keyblock LMK into the “old” LMK slot of a Variant LMK.
- It is not permitted to load an AES Keyblock LMK into the “old” LMK slot of a 3DES Keyblock LMK.
- If multiple LMKs are loaded on the HSM, each can have a corresponding old LMK. The ID of the LMK being processed is defined in the command input.

Example 1:
(Double-length
Variant LMK)

This example loads a double-length Variant LMK from smartcards and installs it as 'old' LMK 00.

```
Secure-AUTH> LO <Return>
Enter LMK id: 00 <Return>
Enter comments: Old LMK for ABC Bank <Return>
Load old LMK from components.
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: Y <Return>
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure until all old component sets have been loaded. When all components have been loaded and the HSM displays the LMK Check value, ensure that this is the expected value.

```
LMK Check: ZZZZZZ
LMK id: 00
LMK key scheme: Variant
LMK algorithm: 3DES (2key)
LMK status: Live
Comments: Old LMK for ABC Bank
Confirm details? [Y/N]: Y <Return>
Secure-AUTH>
```


Example 2: (Triple-length Variant LMK)

This example loads a triple-length Variant LMK from smartcards and installs it as 'old' LMK 00.

```
Secure-AUTH> LO <Return>
Enter LMK id: 01 <Return>
Enter comments: Old LMK for Process System One <Return>
Load old LMK from components.
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: Y <Return>
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure until all old component sets have been loaded. When all components have been loaded and the HSM displays the LMK Check value, ensure that this is the expected value.

```
LMK Check: ZZZZZZ
LMK id: 00
LMK key scheme: Variant
LMK algorithm: 3DES (3key)
LMK status: Live
Comments: Old LMK for Process System One
Confirm details? [Y/N]: Y <Return>
Secure-AUTH>
```

Example 3: (Double- or triple-length Variant LMK)

This example attempts to load a non-Test LMK using a single component when the security setting to enforce multiple components has been set to YES.

```
Secure-AUTH> LO <Return>
Enter LMK id: 00 <Return>
Enter comments: Old LMK for ABC Bank <Return>
Load old LMK from components.
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: n <Return>
Check: AAAAAA
Invalid key - Multiple key components required
Secure-AUTH>
```

Example 4: (3DES Keyblock LMK)

This example loads a 3DES keyblock LMK from smartcards and installs it as 'old' LMK 01.

```
Secure-AUTH> LO <Return>
Enter LMK id: 01 <Return>
Enter comments: Old LMK for XYZ Bank <Return>
Load old LMK from components.
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
Load more components? [Y/N]: Y <Return>
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure until all old component sets have been loaded. When all components have been loaded and the HSM displays the LMK Check value, ensure that this is the expected value.

```
LMK Check: ZZZZZZ
LMK id: 01
LMK key scheme: Key block
LMK algorithm: 3DES (3key)
LMK status: Live
Comments: Old LMK for XYZ Bank
Confirm details? [Y/N]: Y <Return>
Secure-AUTH>
```

Example 5:
(AES Keyblock
LMK)

This example loads an AES keyblock LMK from smartcards and installs it as 'old' LMK 02.

```
Secure-AUTH> LO <Return>
Enter LMK id: 02 <Return>
Enter comments: Old LMK for XYZ Bank <Return>
Load old LMK from components.
```

```
Insert card and enter PIN: ***** <Return>
Check: AAAAAA
```

Remove the smartcard. Insert the second and subsequent smartcards and repeat the loading procedure until all old component sets have been loaded. When all components have been loaded and the HSM displays the LMK Check value, ensure that this is the expected value.

```
LMK Check: ZZZZZZ
LMK id: 02
LMK key scheme: Key block
LMK algorithm: AES-256
LMK status: Live
Comments: Old LMK for XYZ Bank
Confirm details? [Y/N]: Y <Return>
Secure-AUTH>
```

Verify LMK Store

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **V**

Function: To confirm that the check value is identical to the value that was recorded when the LMK set was installed.
For Variant LMKs, the length of the displayed check value is determined by the CS (Configure Security) setting "Restrict Key Check Value to 6 hex chars".
For Keyblock LMKs, the length of the displayed check value is always 6 hex digits.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- LMK Identifier: 2 numeric digits.

Outputs:

- Master key check value.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.

Example:

```
Online> V <Return>
Enter LMK id: 03 <Return>
Check: ZZZZZZ
Online>
```

Duplicate LMK Component Sets

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **DC**

Function: To copy an LMK component onto another smartcard.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- Smartcard (RLMKs are supported) with LMK component.
- PIN for the smartcard. PINs must be entered within 60 seconds of being requested.

Outputs:

- LMK check value.

Errors:

- Load failed check comparison - card is blank
- Not a LMK card - card formatted for HSM settings or is a license card
- Card not formatted - card is not formatted
- Smartcard error; command/return: 0003 - invalid PIN is entered
- Invalid PIN; re-enter - a PIN of less than 4 or greater than 8 is entered.
- Warning - card not blank. Proceed? [Y/N] - LMK card is not blank
- Overwrite LMK set? [Y/N] - the smartcard already contains an LMK component. It can be overwritten if desired.

Example:

```
Secure> DC <Return>
Insert card to be duplicated and enter PIN: ***** <Return>
Insert blank card and enter PIN: ***** <Return>
Writing keys...
Checking keys...
Device write complete, check: ZZZZZZ
Make another copy? [Y/N]: N <Return>
Secure>
```

Delete LMK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>admin.console</u>			

Command: **DM**

Function: To delete a selected LMK and (if loaded) the LMK in the corresponding location in key change storage.

Authorization: The HSM must be in the secure state to run this command. Additionally, the HSM must be either in the Authorized State, or the activity **admin.console** must be authorized, using the Authorizing Officer cards of the specified LMK.

Inputs: • LMK Identifier: 2 numeric digits.

Outputs: • Display of relevant entry from LMK table and the key change storage table.

Errors: • Invalid LMK identifier - no LMK loaded or entered identifier out of range.
• Command only allowed from Secure-Authorized - the HSM is not in Secure State, or the HSM is not authorized to perform this operation, or both.
• LMK id xx is the Default and Management LMK ID – the default and Management LMKs cannot be deleted.

Notes: • LMKs which are the Default or Management LMK cannot be deleted. They Default and Management LMK must be re-assigned to a new LMK before the desired LMK can be deleted. (The LMK ID of the Management and default LMKs can be viewed by running the QS command.)

Example: Secure-AUTH> **DM** <Return>
Enter LMK id: **01** <Return>

LMK table entry:

LMK table:

ID	Auth	Scheme	Algorithm	Status	Check	Comments
01	No	Key Block	3DES(3key)	Test	999999	Test LMK for XYZ Bank

Key change storage table:

ID	Scheme	Algorithm	Status	Check	Comments
01	Variant	3DES(2key)	Test	876543	Old test LMK for XYZ Bank

Confirm LMK deletion [Y/N]: **y** <Return>

LMK deleted from main memory and key change storage

Secure>

Delete 'Old' LMK from Key Change Storage

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **DO**

Function: To delete a selected LMK from key change storage. This command may only be used if an LMK is loaded in the corresponding location in main LMK memory.

Authorization: The HSM must be in the secure state to run this command.

Inputs: • LMK Identifier: 2 numeric digits.

Outputs: • Display of relevant entry from the key change storage table.

Errors: • Invalid LMK identifier - no LMK loaded or entered identifier out of range.

Example:

```
Secure> DO <Return>
Enter LMK id: 01 <Return>

Key change storage table entry:
ID Scheme      Algorithm  Status Check  Comments
01 Variant    3DES(2key) Test   876543 Old test LMK for XYZ
Bank

Confirm LMK deletion [Y/N]: Y <Return>
LMK deleted from key change storage

Secure>
```

View LMK Table

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **VT**

Function: To display the LMK table and the corresponding table for key change storage.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- Displayed LMK table and key change storage table.
- For each LMK currently installed, the following information is displayed:
 - ID – identifier selected during installation of this LMK.
 - Auth – current authorized status:
 - No – not authorized state/activities not active;
 - Yes – authorized state is active;
 - Yes (nX) – ‘n’ authorized activities are active (if HSM is configured for multiple authorized activities), with X identifying whether **Host** or **Console** commands.
 - (Note that LMKs in key change storage cannot be authorized.)
 - Scheme – The LMK scheme:
 - Variant – indicating a Variant LMK
 - Keyblock – indicating a Keyblock LMK
 - Algorithm – the LMK algorithm:
 - 3DES (2key) – indicating a double-length TDES Variant LMK
 - 3DES (3key) – indicating a triple-length TDES Variant or triple-length (3DES) Keyblock LMK
 - AES-256 – indicating an AES Keyblock LMK.
 - Status – the LMK status, selected during generation of the LMK.
 - Live – LMK is a ‘live’ LMK.
 - Test – LMK is a ‘test’ LMK.
 - Check – the check value of the LMK.
 - Comments – the comments entered during installation of this LMK.

Errors: None.

Example 1: *The HSM is configured for single authorized state, but has not been authorized:*

Secure> **VT** <Return>

LMK table:

ID	Authorized	Scheme	Algorithm	Status	Check	Comments
00	No	Variant	3DES(2key)	Test	268604	test variant

Key change storage table:
No keys loaded in key change storage

Secure>

Example 2: *The HSM is configured for single authorized state, and both host and console commands are authorized for LMK 01:*

Secure> **VT** <Return>

LMK table:

ID	Authorized	Scheme	Algorithm	Status	Check	Comments
00	No	Variant	3DES(2key)	Test	268604	test variant
01	Yes(H,C)	Variant	3DES(2key)	Test	268604	test variant
02	Yes(1H,1C)	Variant	3DES(3key)	Live	554279	Production 1

Key change storage table:
No keys loaded in key change storage

Secure>

Example 3: *The HSM is configured for single authorized state, and only host and commands are authorized for LMK 01 (console command authorization has automatically expired after 12 hours):*

Secure> **VT** <Return>

LMK table:

ID	Authorized	Scheme	Algorithm	Status	Check	Comments
00	No	Variant	3DES(2key)	Test	268604	test variant
01	Yes(H)	KeyBlock	AES-256	Live	963272	Mngmnt LMK

Key change storage table:
No keys loaded in key change storage

Secure>

Example 4: *The HSM is configured for multiple authorized activities. Output shows how many host and console commands are authorized for each LMK:*

Online-AUTH> **VT** <Return>

LMK table:

ID	Authorized	Scheme	Algorithm	Status	Check	Comments
00	Yes(1H,1C)	Variant	3DES(2key)	Test	268604	For RST Bank
01	No	KeyBlock	3DES(3key)	Test	999999	For XYZ Bank
02	Yes(1H,1C)	Variant	3DES(3key)	Live	554279	Production 1
03	Yes(0H,1C)	KeyBlock	AES-256	Live	963272	Mngmnt LMK

Key change storage table:

ID	Scheme	Algorithm	Status	Check	Comments
01	Variant	3DES(2key)	Test	876543	For XYZ Bank
02	Variant	3DES(2key)	Live	448796	Old LMK for Production
1					

Online-AUTH>

Chapter 4 – Operational Commands

Authorization Commands

The payShield 9000 HSM needs to be authorized for certain commands to be executed - usually those involving clear text data.

There are two methods of authorizing the HSM – using:

- > a single Authorized State;
- > multiple Authorized Activities.

Note: The console command CS (Configure Security) setting "Enable multiple authorized activities" determines which method is to be used; by default, multiple Authorized Activities are used.

If the HSM needs to be placed in Authorized State using the Authorizing Officer cards (or passwords) corresponding to a particular LMK, then the command will only be authorized for that particular LMK identifier. For example, if the "FK" console command ("Form Key from Components") is authorized using the passwords corresponding to the LMK with identifier "00", then only keys encrypted using LMK "00" may be formed using the command.

It is possible to authorize the HSM using multiple Authorizing Officer cards (or passwords), so that the HSM may be simultaneously authorized for different LMKs.

Note: For PCI HSM compliance, PINs and smartcards must be used to authenticate the Security Officers: passwords must not be used.

The payShield 9000 HSM provides the following console commands to support the authorization of the HSM:

Command	Page
Enter the Authorized State (A)	123
Cancel the Authorized State (C)	125
Enter the Authorized State Multi-Auth (A)	126
Cancel Authorized Activity Multi-Auth (C)	135
View Authorized Activities (VA)	137

Enter the Authorized State

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **A**

Function: To set the HSM into the Authorized State.
The HSM prompts for either Smartcards or Passwords, as applicable, which must correspond to the LMK being authorized.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- LMK Identifier: 1 or 2 numeric digits.
- PIN (if applicable): 5 to 8 alphanumeric characters. The PIN must be entered within 60 seconds. (4-digit PINs on legacy cards will also be accepted.)
- Either:
 - Smartcards (RLMKs are supported) with authorizing both passwords.
 - Password: 16 alphanumeric characters.

Outputs: • Text messages as shown in examples.

Notes:

- If the CS setting "Card/Password authorization" is set to "Card", then the passwords required to put the HSM into the Authorized State will be read from smartcards. Note that only the first 2 LMK component cards contain passwords.
- This command is only available when the console command CS (Configure Security) setting "Enable multiple authorized activities [Y/N]" is set to "N".
- For PCI HSM compliance (see Chapter 10 of the *payShield 9000 General Information Manual*), authentication must use smartcards and PINs, not passwords.
- Use of this command will always cause an entry to be made in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.
- Console commands remain authorized for 12 hours (720 minutes) – see Chapter 10 of the *payShield 9000 General Information Manual*.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Card not formatted - card is not formatted.
- Not a LMK card - card formatted for HSM settings or is a license card.
- Smartcard error; command/return: 0003 - invalid PIN is entered.
- Invalid PIN; re-enter - a PIN of less than 5 or greater than 8 digits is entered.
- Data invalid; please re-enter - the password is an invalid length.

Example 1: *This example authorizes the HSM using smartcards.*

```
Online> A <Return>
Enter LMK id [0-9]: 00 <Return>
First Officer:
Insert card and enter PIN: ***** <Return>
Second Officer:
Insert card and enter PIN: ***** <Return>
AUTHORIZED
Console authorizations will expire in 720 minutes (12 hours).

Online-AUTH>
```

Example 2: *This example authorizes the HSM using passwords.*

```
Online> A <Return>
Enter LMK id [0-4]: 1 <Return>
First Officer:
Password: ***** <Return>
Second Officer:
Password: ***** <Return>                      ← Password
too long
Data invalid; please re-enter: ***** <Return>
AUTHORIZED
Console authorizations will expire in 720 minutes (12 hours).

Online-AUTH>
```

Cancel the Authorized State

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **C**

Function: To cancel the Authorized State.
There is an equivalent command available to the host (Host command 'RA')

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- LMK Identifier: 2 numeric digits.

Outputs:

- Text messages as shown in example.

Notes:

- This command is only available when the console command CS (Configure Security) setting "Enable multiple authorized activities [Y/N]" is set to "N".
- Use of this command will always cause an entry to be made in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.

Example 1:

```
Online-AUTH> C <Return>
Enter LMK id [0-9]: 00 <Return>
NOT AUTHORIZED for LMK id 00
Online>
```

Authorize Activity

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **A**

Function: To authorize the HSM to perform certain specified activities.
In command line mode, the operator specifies which activities are to be authorized.

In menu mode, the operator is prompted to enter the activities.

In both cases, the specified activities are authorized by submitting two Security Officer cards or passwords, which must correspond to the LMK being authorized.

Authorized activities can be made persistent, in which case they are retained even if the power to the HSM is cycled.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- LMK Identifier: 2 numeric digits
- Activities to be authorized.
- Timeout value: Number of minutes before HSM will revoke chosen authorized activity. Where the security setting *Enforce Authorization Time Limit* has been set to "YES" (i.e. to the PCI HSM compliant value) then console commands can be authorized for a maximum period of 12 hours (720 minutes).
- PIN (if applicable): 5 to 8 alphanumeric characters. The PIN must be entered within 60 seconds of being requested. (4-digit PINs on legacy cards will also be accepted.)
- Either:
 - Smartcards (RLMKs are supported) with authorizing both passwords.
 - Password: 16 alphanumeric characters.
- Use "-h" to display help.

Outputs: • Text messages as shown in examples.

Syntax: **A** [*<Activity>*] [*<Activity>*] ...

Activity: *<Category>*. [*<Sub-category>*]. [*<Interface>*] [:*<Timeout>*]

Category =
generate|component|genprint|import|export|pin|audit|admin|diag|misc| command

Sub-category (for 'generate|import|export') = key type code, e.g. 001 for ZPK.

Sub-category (for 'pin') = mailer|clear

Interface = host|console

Timeout = value in minutes or 'p' for persistent. (A maximum of 12 hours (720 minutes) is applied to Console commands.)

Names may be shortened but must remain unique.

- Errors:
- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
 - Card not formatted - card is not formatted.
 - Not a LMK card - card formatted for HSM settings or is a license card.
 - Smartcard error; command/return: 0003 - invalid PIN is entered.
 - Invalid PIN; re-enter - a PIN of less than 4 or greater than 8 is entered.
 - Data invalid; please re-enter: the password is an invalid length.

- Notes:
- If the CS setting "Card/Password authorization" is set to "Card", then the passwords required to put the HSM into the Authorized State will be read from smartcards. Note that only the first 2 LMK component cards contain passwords.
 - This command is only available when the console command CS (Configure Security) setting "Enable multiple authorized activities [Y/N]" is set to "Y".
 - For PCI HSM compliance (see Chapter 10 of the *payShield 9000 General Information Manual*), the following security settings must be set:
 - user authentication must be by smartcard and PIN, and not by using passwords.
 - Authorization time limit for Console commands must be enforced.
 - Where the security setting *Enforce Authorization Time Limit* has been set to "YES" (i.e. to the PCI HSM compliant value) then console commands can be authorized for a maximum period of 12 hours (720 minutes).
 - Use of this command will always cause an entry to be made in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.
 - Activities are described in terms of four fields: Category, Sub-Category, Interface and Timeout. If the Timeout field is omitted, the activity remains authorized until cancelled either by the console command "C" or the host command "RA".
 - Omitting either the Sub-Category and/or the Interface field is equivalent to authorizing multiple activities consisting of all possible combinations of valid values for the missing fields. For clarification:

pin.mailer

is equivalent to:

pin.mailer.host
pin.mailer.console

and:

pin

is equivalent to:

pin.clear.console
pin.clear.host
pin.mailer.console
pin.mailer.host

- When authorizing activities, two (or more) activities may overlap, for example:

pin
pin.mailer

- There is no requirement to attempt to reduce activities to the minimum set. The list of authorized activities simply consists of all those entered (and authorized) by the user.
- There is one case when it will be necessary to overwrite an existing activity: when only the Timeout field changes. For example, suppose that the following activity is authorized:

export.001.console:11

and the user uses the 'A' command to authorize the following activity:

export.001.console:60

then this should overwrite the first one (even if the newer activity has a shorter *Timeout* value).

- Note: When omitting the sub-category, but including the interface, there should be two delimiters "." between them:
Example: export..host allows export of any (valid) key using a host command.
- The option to make an authorization persistent (i.e. to survive across a re-boot of the HSM) is only available for Host commands and where the authorization is also permanent.

Example 1:
(Variant or
Keyblock LMK)

This example authorizes a single activity via the menu.

```
Online> A <Return>
Enter LMK id [0-9]: 0 <Return>
No activities are authorized for LMK id 00.
List of authorizable activities:
generate      genprint      component      import
export        pin           audit          admin
diagnostic    misc          command
Select category: pin <Return>
clear         mailer
Select sub-category, or <RETURN> for all: mailer <Return>
host          console
```



```
Select interface, or <RETURN> for all: <Return>
Enter time limit for pin.mailer, or <RETURN> for permanent:
<Return>
Make activity persistent? [Y/N]: N <Return>
Enter additional activities to authorize? [y/N]: N <Return>

The following activities are pending authorization for LMK id
00:
pin.mailer

First Officer:
Insert Card for Security Officer and enter the PIN: *****
<Return>
Second Officer:
Insert Card for Security Officer and enter the PIN: *****
<Return>

The following activities are authorized for LMK id 00:
pin.mailer

Online-AUTH>
```

Example 2:
(Variant or
Keyblock LMK)

This example authorizes activities via the command line, with no time limits specified.

```
Online> A gene comp genp i e p au ad di m comm<Return>

Enter LMK id [0-4]: 0 <Return>

Console authorizations will expire in 720 minutes (12 hours).

The following activities are pending authorization for LMK id
00:

admin..console:720
admin..host
audit..console:720
audit..host
command..console:720
command..host
component..console:720
component..host
diagnostic..console:720
diagnostic..host
export..console:720
export..host
generate..console:720
generate..host
genprint..console:720
genprint..host
import..console:720
import..host
misc..console:720
misc..host
pin..console:720
pin..host

First officer:
Insert card and enter PIN: *****<Return>

Second officer:
Insert card and enter PIN: *****<Return>

The following activities are authorized for LMK id 00:
```

```
admin..console:720  (720 mins remaining)
admin..host
audit..console:720  (720 mins remaining)
audit..host
command..console:720  (720 mins remaining)
command..host
component..console:720  (720 mins remaining)
component..host
diagnostic..console:720  (720 mins remaining)
diagnostic..host
export..console:720  (720 mins remaining)
export..host
generate..console:720  (720 mins remaining)
generate..host
genprint..console:720  (720 mins remaining)
genprint..host
import..console:720  (720 mins remaining)
import..host
misc..console:720  (720 mins remaining)
misc..host
pin..console:720  (720 mins remaining)
pin..host
```

```
Online-AUTH>
```

Example 3:
(Variant LMK)

This example authorizes three activities additional Example 1 via the menu.

```
Online-AUTH> A <Return>
Enter LMK id [0-9]: 00 <Return>
The following activities are authorized for LMK id 00:
pin.mailer
List of authorizable activities:
generate      genprint      component      import
export        pin          audit          admin
diagnostic    misc          command
Select category: generate <Return>
000           100           200           001
002           400           003           006
008           009           109           209
309           409           509           709
00a           00b           rsa
Select sub-category, or <RETURN> for all: 000 <Return>
host          console
Select interface, or <RETURN> for all: C <Return>
Enter time limit for generate.000.console, or <RETURN> for
permanent: 60 <Return>

Enter additional activities to authorize? [y/N]: y <Return>
List of authorizable activities:
generate      genprint      component      import
export        pin          audit          admin
diagnostic    misc          command
Select category: export <Return>
000           100           200           001
002           400           003           006
008           009           109           209
309           409           509           709
00a           00b           rsa
Select sub-category, or <RETURN> for all: 001 <Return>
host          console
Select interface, or <RETURN> for all: H <Return>
Enter time limit for export.001.host, or <RETURN> for
permanent: <Return>
Make activity persistent? [Y/N]: n <Return>

Enter additional activities to authorize? [y/N]: y <Return>
List of authorizable activities:
generate      genprint      component      import
export        pin          audit          admin
diagnostic    misc          command
Select category: admin <Return>
host          console
Select interface, or <RETURN> for all: c <Return>
Enter time limit for admin, or <RETURN> for permanent: 240
<Return>

Enter additional activities to authorize? [y/N]: n <Return>
The following activities are pending authorization for LMK id
00:
admin..console:240
export.001.host
generate.000.console:60

First Officer
Insert Card for Security Officer and enter the PIN: ****
<Return>
Second Officer
Insert Card for Security Officer and enter the PIN: ****
<Return>

The following activities are authorized for LMK id 00:
admin:240 (240 mins remaining)
export.001.host
generate.000.console:60 (60 mins remaining)
```

Example 4:
(Variant LMK)

This example authorizes three activities additional to Example 1 via the command line, including time limits.

```
pin.mailer
Online-AUTH>

Online-AUTH> A gene.000.con:60 exp.001.host:p admin:240
<Return>
Enter LMK id [0-19]: 00 <Return>

The following activities are pending authorization for LMK id
00:

admin:240
export.001.host:persistent
generate.000.console:60

First Officer:
Insert Card for Security Officer and enter the PIN: ****
<Return>
Second Officer:
Insert Card for Security Officer and enter the PIN: ****
<Return>

The following activities are authorized for LMK id 01:

admin:240 (240 mins remaining)
export.001.host:persistent
generate.000.console:60 (60 mins remaining)

Online-AUTH>
```

Example 5:
(Variant or
Keyblock LMK)

This example authorizes a single activity via the command line.

```
Online> A pin.clear <Return>
Enter LMK id [0-9]: 01 <Return>

Console authorizations will expire in 720 minutes (12 hours).

The following activities are pending authorization for LMK id
01:

pin.clear.console:720
pin.clear.host

First Officer:
Insert Card for Security Officer and enter the PIN: ****
<Return>
Second Officer:
Insert Card for Security Officer and enter the PIN: ****
<Return>

The following activities are authorized for LMK id 01:

pin.clear.console:720 (720 mins remaining)
pin.clear.host

Online-AUTH>
```

Example 6:

This example authorizes an additional three activities via the menu.

(Keyblock LMK)

```

Online-AUTH> A <Return>
Enter LMK id [0-9]: 01 <Return>
The following activities are authorized for LMK id 01:
pin.clear
List of authorizable activities:
generate      genprint      component      import
export        pin          audit          admin
diagnostic    misc          command
Select category: export <Return>

01            B0            C0            11
12            13            D0            21
22            E0            E1            E2
E3            E4            E5            E6
31            32            K0            51
52            M0            M1            M2
M3            M4            M5            61
62            63            64            65
P0            71            72            73
V0            V1            V2

Select sub-category, or <RETURN> for all: 72 <Return>
host          console
Select interface, or <RETURN> for all: C <Return>
Enter time limit for export.72.console, or <RETURN> for
permanent: 60 <Return>

Enter additional activities to authorize? [y/N]: y <Return>
List of authorizable activities:
generate      genprint      component      import
export        pin          audit          admin
diagnostic    misc          command
Select category: admin <Return>
host          console
Select interface, or <RETURN> for all: <Return>
Enter time limit for admin, or <RETURN> for permanent: 240
<Return>

Enter additional activities to authorize? [y/N]: y <Return>
List of authorizable activities:
generate      genprint      component      import
export        pin          audit          admin
diagnostic    misc          command
Select category: misc <Return>
host          console
Select interface, or <RETURN> for all: C <Return>
Enter time limit for admin, or <RETURN> for permanent:
<Return>
Make activity persistent? [Y/N]: n <Return>

Enter additional activities to authorize? [y/N]: n <Return>
The following activities are pending authorization for LMK id
00:
misc..console
admin:240
export.72.console:60

First Officer
Insert Card for Security Officer and enter the PIN: ****
<Return>
Second Officer
Insert Card for Security Officer and enter the PIN: ****
<Return>

The following activities are authorized for LMK id 01:
misc..console
admin:240 (240 mins remaining)
export.72.console (60 mins remaining)
pin.clear

```

Example 7:
(Keyblock LMK)

This example authorizes an additional three activities via the command line.

```
Online-AUTH> a exp.001.con:60 admin:240 misc..console
<Return>
```

```
Enter LMK id [0-1]: 01 <Return>
```

Console authorizations will expire in 720 minutes (12 hours).

The following activities are pending authorization for LMK id 01:

```
admin:240
export.001.console:60
misc..console:720
```

First Officer:

Insert Card for Security Officer and enter the PIN: ****
<Return>

Second Officer:

Insert Card for Security Officer and enter the PIN: ****
<Return>

The following activities are authorized for LMK id 01:

```
admin:240 (228 mins remaining)
export.001.console:60 (60 mins remaining)
export.001.host:persistent
generate.000.console:60 (48 mins remaining)
misc..console:720 (720 mins remaining)
pin.clear.console:720 (712 mins remaining)
pin.clear.host
```

```
Online-AUTH>
```

Cancel Authorized Activity

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **C**

Function: To cancel one or more Authorized Activities.

Authorization: The HSM does not require any authorization to run this command.

Inputs: • LMK Identifier: 2 numeric digits.

Outputs: • Text messages as shown in examples.

Notes: • This command is only available when the console command CS (Configure Security) setting "Enable multiple authorized activities [Y/N]" is set to "Y".

Syntax: **C** [*<Activity>*] [*<Activity>*] ...

Activity: *<Category>* [*<Sub-category>*] [*<Interface>*] [*<Timeout>*]

Category =
generate|component|genprint|import|export|pin|audit|admin|diag
|
misc| command

Sub-category (for 'generate|import|export') = key name, e.g. TPK, MK-AC, etc.

Sub-category (for 'pin') = mailer|clear

Interface = host|console

Timeout = value in minutes or 'p' for persistent

Names may be shortened but must remain unique.

When canceling an authorized activity which includes a timeout, the original value of the timeout should be specified.

Note: When omitting the sub-category, but including the interface, there should be two delimiters "." between them:

Example: export..host allows export of any (valid) key using a host command.

Errors: • Invalid LMK identifier - no LMK loaded or entered identifier out of range.
• Invalid input.

Notes: • Use of this command will always cause an entry to be made in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.

Example 1:
(Variant or
Keyblock LMK)

This example cancels an existing activity via the menu.

```
Online-AUTH> C <Return>
Enter LMK id [0-9]: 00 <Return>
Cancel pin.mailer? [y/N] y <Return>
No activities are authorized for LMK id 00.
Online>
```

Note: This example assumes that the activities in the Authorize Activity command Example 1 (above) are active.

Example 2:
(Variant or
Keyblock LMK)

This example cancels an existing activity via the command line.

```
Online-AUTH> C pin.mailer <Return>
Enter LMK id [0-1]: 00 <Return>
No activities are authorized for LMK id 00.
Online>
```

Note: This example assumes that the activities in the Authorize Activity command Example 2 (above) are active.

Example 3:
(Variant LMK)

This example cancels an existing activity via the menu.

```
Online-AUTH> C <Return>
Enter LMK id [0-4]: 00 <Return>
Cancel admin:240 (194 mins remaining) ? [y/N] y <Return>
Cancel export.001.host? [y/N] N <Return>
Cancel generate.000.console:60 (14 mins remaining)? [y/N] y
<Return>
Cancel pin.mailer? [y/N] N <Return>
The following activities are authorized for LMK id 00:
export.001.host
pin.mailer
Online-AUTH>
```

Note: This example assumes that the activities in the Authorize Activity command Example 3 (above) are active.

Example 4:
(Variant LMK)

This example cancels an existing activity via the command line.

```
Online-AUTH> C gene.000.c admin <Return>
Enter LMK id [0-9]: 00 <Return>
The following activities are authorized for LMK id 00.
export.001.host
pin.mailer
Online-AUTH>
```

Note: This example assumes that the activities in the Authorize Activity command Example 4 (above) are active.

Example 5:
(Variant or
Keyblock LMK)

This example cancels an existing activity via the command line.

```
Online-AUTH> C pin.clear <Return>
Enter LMK id [0-9]: 01 <Return>
No activities are authorized for LMK id 01.
Online>
```

Note: This example assumes that the activities in the Authorize Activity command Example 5 (above) are active.

View Authorized Activities

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **VA**

Function: To view all active authorized activities.

Authorization: The HSM does not require any authorization to run this command.

Inputs: • LMK identifier: 2 numeric digits.

Outputs: • List of active authorized activities.

Errors: • Invalid LMK identifier - no LMK loaded or entered identifier out of range.

Example 1: *This example applies when multiple authorized activities has been enabled..*

(Multiple authorized activities enabled)

```
Online-AUTH> VA <Return>
Enter LMK id: 00 <Return>
The following activities are authorized for LMK id 00:
admin:240 (228 mins remaining)
export.001.host:persistent
generate.000.console:60 (48 mins remaining)
```

```
Online-AUTH>
```

Note: This example assumes the activities in the Authorize Activity command Example 4 (above) were authorized 12 minutes ago.

Example 2: *This example applies when multiple authorized activities has not been enabled..*

(Multiple authorized activities disabled)

```
Online-AUTH> VA <Return>
Enter LMK id [0-9]: 0 <Return>
LMK id 00 is authorized.
Console authorization expires in 716 minute(s).
```

```
Online-AUTH>
```

Note: This example assumes that authorized state was enabled 4 minutes ago.

Logging Commands

An Error Log and an Audit Log are provided, each with a command to display the log and a command to clear the log. There is also a command to enable the user to set their time zone, so that the correct time is displayed in audit log reports.

The Error log stores fault information for use by Thales e-Security support personnel. The error log is used to log unexpected software errors, hardware failures and alarm events. Whenever an error occurs, that error code is stored, along with the time, date and severity level. Additional errors that have the same error code cause the time and date of that code to be updated. In this way, each error type remains in the log (with the most recent time and date) and is not lost. The severity levels are:

- > Informative (0) Something abnormal happened, but was not important.
- > Recoverable (1) Something abnormal happened, but the unit recovered from it without rebooting or losing data.
- > Major (2) Something abnormal happened, but the unit recovered from it but may have lost data/information due to restarting a process or re-initializing hardware. The unit may not function in a full capacity.
- > Catastrophic (3) Something abnormal happened, and the unit had to reboot to recover.

Only catastrophic errors cause the HSM to reboot. New errors cause the Fault LED on the front panel to flash.

Whenever the HSM state is altered through power-up, key-lock changes or console commands, the Audit log is updated with the action and the time and date. The Audit log can also be configured to record execution of almost any console or host command. The Audit log records state changes until it is 100% full and for each subsequent state change the earliest (i.e. oldest) record in the log is deleted to make room for the new record. A number of host commands are provided which allow the host computer to extract and archive (print) audit records from the HSM.

Management of the Audit journal is performed from the console using the command 'AUDITOPTIONS', whilst 'AUDITLOG' is used to retrieve the log and 'CLEARAUDIT' to clear the log. The HSM must be put into the secure-authorized state in order to execute the 'AUDITOPTIONS' and 'CLEARAUDIT' console commands.

Note: Auditing host or console commands may impact HSM performance.

The payShield 9000 HSM provides the following console commands to support storage and retrieval of HSM settings:

Command	Page
Display the Error Log (ERRLOG)	140
Clear the Error Log (CLEARERR)	141
Display the Audit Log (AUDITLOG)	142
Clear the Audit Log (CLEARAUDIT)	144
Audit Options (AUDITOPTIONS)	145
Print the Audit Log (AUDITPRINT)	149

Display the Error Log

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **ERRLOG**

Function: To display the entries in the error log.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- A listing of the errors in the error log, or text message: "Error log is empty".

Errors: None.

Notes: In software versions up to v2.1, power supply errors are added to the error log only when the HSM is restarted. From v2.2 onwards, power supply errors are logged as soon as they are detected.

Example 1: Offline> ERRLOG <Return>
Error log is empty
Offline>

Example 2: Offline> ERRLOG <Return>
Error Log (2 entries)

1: May 01 09:35:00 ERROR (1): Invalid queue size (Severity: 2,
Code = 00000001, Sub-code = 00000002)
2: May 01 09:35:02 ERROR (1): Key3 cannot be specified without
key2 (Severity: 0, Code = 00000004, Sub-code = 00000003)
3: May 06 13:55:00 ERROR: [Power Supply: FAILED (PSU 2
Failed)] (Severity: 3, Code = 0x00000001, Sub-Code =
0x0000000E)

Please copy this log to a text file and send it
to your regional Thales E-Security Support center.

Offline>

Clear the Error Log

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CLEARERR**

Function: To clear the entries in the error log.

Authorization: The HSM must be in the secure state to run this command.

Inputs: None.

Outputs: •A confirmation message.

Errors: None.

Example:

```
Secure> CLEARERR <Return>
Error log Cleared
Secure>
```

Display the Audit Log

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **AUDITLOG**

Function: To display the entries in the audit log.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- A listing of the entries in the audit log.
 - For authorizations, the period of authorization of Console commands will be indicated by attaching text of the form ":123" (representing 123 minutes) to the identity of the authorized activity.
- The following text messages can be output:
 - Audit Log (in entries)
 - Continue displaying audit log entries? Yes/No/Continuous

Notes:

- Certain items are always recorded in the Audit Log, irrespective of the selections made using AUDITOPTIONS – for further information see Chapter 10 of the *payShield 9000 General Information Manual*. These are:
 - Serial numbers of smartcards used to authenticate users at the HSM or to payShield Manager.
 - Authorization of activities
 - Cancellation of authorization.
 - Key and component entry at the Console or payShield Manager.

When key and component entry are forcibly logged in this way, the log entry indicates successful completion of the action.

The user can, as in earlier versions of software, use AUDITOPTIONS to specify that the key and component entry commands are logged: this will normally result in 2 entries in the audit log – one resulting from the AUDITOPTIONS setting indicating that the command was initiated, and the forcible logging indicating the successful completion of the command. If the command does not complete successfully (e.g. because it was cancelled by the user) then there will be no forcible logging, but the entry indicating the command was initiated will still be there if the command was specified in AUDITOPTIONS.

- The Audit Log is now displayed with the most recent entries shown first: up to software version 2.1 the Audit Log was displayed with oldest entries first. This change has been made because, with a maximum length of 50,000 records, it can take a long time to display the complete Audit Log because of the speed limitations of serial connections.

Errors: None.

Example 1: Offline> **AUDITLOG** <Return>
Audit log is empty
Offline>

Example 2: Offline> **AUDITLOG** <Return>
Audit Log (10 entries)
Counter Time Date Command/Event

0000000268 13:55:00 02/Jul/2013 Diagnostic self test failure: Power
0000000267 16:45:07 01/Jul/2013 Authorized activity admin..host was
cancelled for LMK id 0
0000000266 16:45:05 01/Jul/2013 Authorized activity
admin..console:123 was cancelled
0000000265 15:54:02 01/Jul/2013 Key I/O command BK executed
0000000264 15:35:55 01/Jul/2013 Activity component..console:123 was
authorized for LMK id 0
0000000263 15:08:48 01/Jul/2013 Smartcard activated: 20025151
0000000262 15:08:48 01/Jul/2013 Smartcard activated: 20025132
0000000261 10:42:32 01/Jul/2013 Host command CA, response 00
0000000260 10:36:03 01/Jul/2013 Host command CA, response 69
0000000259 10:34:57 01/Jul/2013 System restarted
0000000258 10:32:48 01/Jul/2013 Keylock turned to Online
0000000257 10:32:21 01/Jul/2013 Console command CH
0000000256 09:01:56 01/Jul/2013 Diagnostic self tests passed.

Offline>

After 20 entries are displayed continuously, the following text
is displayed:

Continue displaying audit log entries? [Y/N/C]:

Clear the Audit Log

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>audit.console</u>			

Command: **CLEARAUDIT**

Function: To clear the entries in the audit log.

Authorization: The HSM must be in the secure state to run this command. Additionally, the HSM must be either in the Authorized State, or the activity **audit.console** must be authorized, using the Authorizing Officer cards of the Management LMK.

Inputs: None.

Outputs:

- One of the following text messages:
 - Audit Log Cleared
 - Audit Log is empty

Errors:

- Command only allowed from Secure-Authorized - the HSM is not in Secure State, or the HSM is not authorized to perform this operation, or both.

Example 1:

```
Secure-AUTH> CLEARAUDIT <Return>
Warning! The HSM's audit log contains entries that have not yet
been printed.
Please confirm that you wish to delete the entire audit log.
[Y/N]: Y <Return>
Audit Log Cleared

Secure-AUTH>
```


Audit Options

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Required		
Activity: <u>audit.console</u>		

Command: **AUDITOPTIONS**

Function: To configure the HSM's auditing functionality.
The HSM can be configured to monitor and record the following events:

- Execution of individual host command
- Execution of individual console command
- User interactions, including:
 - System restart (e.g. power cycle)
 - State transitions (i.e. Offline, Online, Secure)
 - LMK installation / erasure
 - Authorization activation/cancelling
- The running and result of automatic self tests.
- Error responses to Host commands
- Host connection failures resulting from deployment of Access Control Lists.
- Secure Host Communication session negotiation failures resulting from attempted use of out-of-date certificates.

Authorization: The HSM must be in the offline or secure state to use this command to change the items to be audited. Additionally, the HSM must be either in the Authorized State, or the activity **audit.console** must be authorized, using the Authorizing Officer cards of the Management LMK.
The current list of items being audited can be viewed in online state.

Inputs:

- Changes to configuration:
 - Audited console commands:
 - +CXX to enable auditing of console command XX
 - -CXX to disable auditing of console command XX
 The "?" character can be used as a wildcard when specifying the commands.
 - Audited host commands
 - +HXX to enable auditing of host command XX
 - -HXX to disable auditing of host command XX
 The "?" character can be used as a wildcard when specifying the commands.
 - Audit Error responses to Host Commands (Y/N)
 - Audit user actions (Y/N)
 - Audit counter value
 - Audit Utilization Data Resets (Y/N)
 - Audit Automatic Self testing (Y/N)
 - Audit ACL connection failures (Y/N)
 - Audit out-of-date Certificates for Secure Host Sessions (Y/N)

Outputs: • Current & new configuration details:

- List of audited console commands
- List of audited host commands
- List of user actions
- Results of automatic self tests
- Audit counter value

Notes:

- Certain items are always recorded in the Audit Log, irrespective of the selections made using AUDITOPTIONS – See Chapter 10 of the *payShield 9000 General Information Manual*. These are:
 - Serial numbers of smartcards used to authenticate users at the HSM or to payShield Manager.
 - Authorization of activities
 - Cancellation of authorization.
 - Key and component entry at the Console or Payshield Manager. This relates to the following Console commands (or HSM equivalents):
 - BK Form a Key from Components
 - CV Generate a Card Verification Value
 - D Form a ZMK from Encrypted Components
 - DE Form a ZMK from Clear Components
 - FK Form Key from Components
 - IK Import a Key
 - IV Import a CVK or PVK
 - LK Load LMK
 - LO Move Old LMKs into Key Change Storage
 - PV Generate a Visa PIN Verification Value

When key and component entry are forcibly logged in this way, the log entry indicates successful completion of the action.

The user can, as in earlier versions of software, use AUDITOPTIONS to specify that the key and component entry commands are logged: this will normally result in 2 entries in the audit log – one resulting from the AUDITOPTIONS setting indicating that the command was initiated, and the forcible logging indicating the successful completion of the command. If the command does not complete successfully (e.g. because it was cancelled by the user) then there will be no forcible logging, but the entry indicating the command was initiated will still be there if the command was specified in AUDITOPTIONS.

- **Audit Error Responses to Host Commands:** this setting allows any relevant error responses to Host commands to be logged. In this context, “relevant” means error responses which may indicate situations that require investigation by the payShield 9000 Administrators or Security Officers. The use of this setting will therefore not log non-00 error responses which are purely for information or which indicate “business as usual” (e.g. a customer entering an incorrect PIN at a terminal). See *Appendix O* for information on which non-00 error responses are not logged.

- Auditing items (such as heavily used Host commands) which result in a high rate of update to the Audit Log will impact negatively on performance of the HSM.
- After completing the AUDITOPTIONS command, a reboot of the HSM may be required in order to activate the new settings.

Errors:

- Command only allowed from Offline-Authorized - the HSM is not in Offline (or Secure) State, or the HSM is not authorized to perform this operation, or both.
- Invalid Entry - the value entered is invalid.
- Card not formatted to save/retrieve HSM settings - Attempt with another card? [Y/N]

Example:

```
Offline-AUTH> AUDITOPTIONS <Return>
List of Audited Console Commands:
GC, GS, EC, FK
List of Audited Host Commands:
A0, A4, GG, GY
Audit Error Responses to Host Commands:
Disabled
Audit User Actions:
Enabled
Audit Counter Value:
0000000253
Audited utilization data resets:
Enabled
Audited diagnostic self tests:
Disabled

Modify Audited Command List? [Y/N]: Y <Return>
Enter command code (e.g. +CDE) or Q to Quit: +CDE <Return>
Console command DE added to list
Enter command code (e.g. +CDE) or Q to Quit: -HA4 <Return>
Host command A4 removed from list
Enter command code (e.g. +CDE) or Q to Quit: Q <Return>

Audit Error Responses to Host Commands? [Y/N]: Y <Return>

Audit User Actions (Y/N): N <Return>

Audit ACL connection failures? [Y/N]: Y <Return>

Audit out-of-date Certificates for Secure Host sessions? [Y/N]:
Y <Return>

Current Audit Counter value is: 0000000253
Enter new value or <RETURN> for no change: 2000 <Return>

Audit Utilization Data Resets? [Y/N]: Y <Return>

Audit Automatic Self Testing? [Y/N]: Y <Return>

Audit User Actions: YES
Audit Error Responses to Host Commands: YES
Audit utilization data resets: YES
Audit diagnostic self tests: YES
Audit ACL connection failures: YES
Audit out-of-date Certificates for Secure Host Sessions:
YES
Audit Counter Value:
0000002000
```

List of Audited Console Commands:

GC, GS, EC, FK, DE

List of Audited Host Commands:

A0, GG, G

Audit Error Responses to Host Commands:

Enabled

Audit User Actions:

Disabled

Audit Counter Value:

00002AAF

Audited utilization data resets:

Enabled

Audited diagnostic self tests:

Enabled

Save Audit Settings to smartcard? [Y/N]: Y <Return>

Insert Card and press Enter: <Return>

Audit Settings written to the smartcard.

Offline-AUTH>

Print the Audit Log

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not Required			

Command: **AUDITPRINT**

Function: To print the HSM's audit log at a printer attached to the HSM.

Authorization : Authorization is not required.

Inputs: • Whether to print all records, or only unarchived records.

Outputs: • A list of all the selected records showing the following data:

- The sequential audit counter
- The time of the event, in the format HHMMSS
- The date of the event, in the format DDMMYY
- The command description, including:
 - The command code type (H=Host, C=Console, F=Fraud Event, A=User Action)
 - The command or action code
 - For Host commands, the response error code.
- Random MAC key used to generate the MAC
- MAC calculated over the audit record.

For more detail, see the Audit Record Format in Appendix B of the Host Command Reference Manual.

Example of output:

Counter	Time	Date	Command	MAC Key	MAC
1001CF33	135209	180511	H M2 06	AA69C75033EA50810209D24F17E93786	ACBC947DA5E06947
1001CF34	135209	180511	H M2 06	5D53F23A43A7AC692C77754FB00EBCA6	E3DFFE68209F4A1E
1001CF35	135209	180511	H M2 06	787C6FC766E544CD4A2EF56DB1DE1C14	D5321C3CF8E36DCB
1001CF36	135209	180511	H M2 06	34D3B4CE59DDC0BA4C128EF88721D50C	86D18019F2E1D717
1001CF37	135209	180511	H M2 06	F893D165B7CAD8DC44A59CF33F895FE	C5C14C8D93892004
1001CF38	135209	180511	H M2 06	C364F9C499C89514A3EB6BBA75BC2C87	55B8024854727C41
1001CF39	135209	180511	H M2 06	D229ACB7F9C5EEA7FB55761EEB9947D7	BB6E67CA6DEF2584
1001CF3A	135209	180511	H M2 06	0F5A3BAB8A93FEC30E9C125E585FB005	1D84136FA9162B1B
1001CF3B	135209	180511	H M2 06	7F78D6858D729710477C0CEF18917281	CE6746ADAE4B65AC
1001CF3C	135209	180511	H M2 06	C1EA998068CD989A5383A8EA7B52EB1C	F2B5A526C100EAB3
1001CF3D	135209	180511	H M2 06	5BA7D93E19DA1EEA14AAA1BCDB1CB45B	2DCF25D8E0DE381F
1001CF3E	135209	180511	H M2 06	9C019A9DF544F2F31300CCD54DF44DF1	7FA5EA6DA98043C9
1001CF3F	135209	180511	H M2 06	D4AD0D70A5EBFE61B5BAF2DC509FB478	36D504B7E837778B
1001CF40	135209	180511	H M2 06	B29D7E22350640A702255D1A024777AE	C8495DF637BA3E6A
1001CF41	135209	180511	H M2 06	4B6C7887A7662663FDD76EEE6FE3BE27	749BD7153ADD5A01
1001CF42	135209	180511	H M2 06	A9048C7576CBE29227FA824AE51B0323	4FB59F661352A05B
1001CF43	135209	180511	H M2 06	27D6C576FE6F1B0537A51175777C5820	B6EE89EF4F65F7BC

Notes: • Printing of an Audit Log record causes its "Archived" flag to be set.

Example: Offline> **AUDITPRINT** <Return>

Print All records or just Unarchived records? [A/U]: **A** <Return>

Commence Printing? [Y/N]: **Y** <Return>

Printing complete: 2000 record(s) printed.

Offline>

Time and Date Commands

The SETTIME command is used to set the system time and date used by the payShield 9000 HSM for the audit log entries. The user should use this command to adjust the time for the local timezone. The time and date can be queried using the GETTIME command.

The payShield 9000 HSM provides the following console commands to support storage and retrieval of HSM settings:

Command	Page
Set the Time and Date (SETTIME)	151
Query the Time and Date (GETTIME)	152
Set Time for Automatic Self-Tests (ST)	153

Set the Time and Date

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>admin.console</u>			

Command: **SETTIME**

Function: To set the system time and date used by the HSM.

Authorization: The HSM must be in the secure state to run this command. Additionally, the HSM must be either in the Authorized State, or the activity **admin.console** must be authorized, using the Authorizing Officer cards of the Management LMK.

Inputs:

- The time in hours and minutes.
- The date in year, month and day.

Outputs:


- Text messages, as in the example below.

Errors:

- Command only allowed from Secure-Authorized - the HSM is not in Secure State, or the HSM is not authorized to perform this operation, or both.
- Response invalid. Re-enter - an invalid value has been entered.

Example:

```
Secure-AUTH> SETTIME <Return>
Enter hours [HH] (24 hour format): 10 <Return>
Enter minutes [MM]: 08 <Return>
Enter year [YYYY] (2009 or above): 2014 <Return>
Enter month [MM]: 02 <Return>
Enter day [DD]: 12 <Return>
The system time has been modified.
Secure-AUTH>
```

 Setting the date or time back may prevent the payShield Manager from allowing a user to login. Care must be taken when changing the date back such that it is not earlier than the creation date/time of any of the smartcards that will be used to access the HSM.

Query the Time and Date

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **GETTIME**

Function: To query the system time and date.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- The year, month and date.
- The time in hours, minutes and seconds.

Errors: None.

Example: Online> GETTIME <Return>
 System date and time: Feb 12 10:08:19 2014
 Online>

Set Time for Automatic Self-Tests

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **ST**

Function: Reports the time of day when the daily automatic self-tests required for PCI HSM compliance will be run, and allows this time to be changed.

Authorization: The HSM does not require any authorization to run this command.

Inputs: Time of day.

Outputs: None

Errors: None.

Notes:

- The default time for running the diagnostics is 0900.

Example:

```
Secure> ST <Return>

Self test run time is 09:00.

Change? [Y/N]: y <Return>

Enter hour [HH] (24 hour format): 13 <Return>
Enter minute [MM]: 55 <Return>

Self test run time changed to 13:55.

Secure>
```

Settings, Storage and Retrieval Commands

Commands are provided to save the payShield 9000 HSM's Alarm, Host and Security settings to a smartcard and to restore the settings to the HSM. Besides the dedicated command to Save HSM Settings to Smartcard, the following individual configuration commands have the option to save settings to smartcard:

- > CL (Configure Alarms) to save the Alarm configuration.
- > CH (Configure Host) to save the Host port configuration.
- > CS (Configure Security) to save the Security configuration.
- > AUDITOPTIONS (Audit Options) to save the Audit configuration.

The payShield 9000 HSM provides the following console commands to support storage and retrieval of HSM settings:

Command	Page
Save HSM Settings to a Smartcard (SS)	155
Retrieve HSM Settings from a Smartcard (RS)	156

Save HSM Settings to a Smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>admin.console</u>			

Command: **SS**

Function: To save the Alarm, Host Port, Security, Audit, Command, and PIN Block settings to a smartcard (RACCs are supported).

Authorization: The HSM must be in the secure state to run this command. Additionally, the HSM must be either in the Authorized State, or the activity **admin.console** must be authorized, using the Authorizing Officer cards of the Management LMK.

Outputs: • Confirmation messages that Alarm, Host, Security, Audit, Command, and PIN Block settings are saved.

Errors: • Card not formatted to save/retrieve HSM settings. Attempt with another card? [Y/N] - card is formatted for LMK storage or is a license card.
• Card not formatted. Attempt with another card? [Y/N] - card is not formatted.
• Command only allowed from Secure-Authorized - the HSM is not in Secure State, or the HSM is not authorized to perform this operation, or both.

Example: Secure-AUTH> **ss** <Return>
Insert card and press ENTER: <Return>
ALARM settings saved to the smartcard.
HOST settings saved to the smartcard.
SECURITY settings saved to the smartcard.
AUDIT settings saved to the smartcard.
COMMAND settings saved to the smart card.
PIN BLOCK settings saved to the smart card.
Secure-AUTH>

Retrieve HSM Settings from a Smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>admin.console</u>			

Command: **RS**

Function: To read the Alarm, Host Port, Security, Audit, Command, and PIN Block settings from a smartcard. The user is then prompted to use these to overwrite the existing HSM settings. If the settings on the smartcard were saved using a configuration command (CL, CH, CS and AUDITOPTIONS), then only those settings are overwritten.

Authorization: The HSM must be in the secure state to run this command. Additionally, the HSM must be either in the Authorized State, or the activity **admin.console** must be authorized, using the Authorizing Officer cards of the Management LMK.

Inputs: • Whether to overwrite each of the groups of saved settings.

Outputs: • The Alarm, Host, Security, Audit, Command, and PIN Block settings stored on the smartcard are listed.

Errors:

- Card not formatted to save/retrieve HSM settings. Attempt with another card? [Y/N] - card is formatted for LMK storage or is a license card.
- Card not formatted. Attempt with another card? [Y/N] - card is not formatted.
- Command only allowed from Secure-Authorized - the HSM is not in Secure State, or the HSM is not authorized to perform this operation, or both.

Example:

```
Secure-AUTH> RS <Return>
Insert card and press ENTER: <Return>
Temperature Alarm: ON
Motion Alarm: HIGH
Self Test Run Time: 09:00
Overwrite alarm settings with the settings above? [Y/N]: Y <Return>
ALARM settings retrieved from smartcard

Message header length: 4
Protocol: ETHERNET
Character format: ASCII
UDP active: YES
TCP active: YES
TLS/SSL active: YES
TLS enforced: YES
Number of TCP connections: 1
Well-Known-Port: 1500
Well-Known-TLS-Port: 2500
Number of host interfaces: 1

Overwrite host settings with the settings above? [Y/N]: n <Return>

PIN length: 04
Old encrypted PIN length: 05
Echo: OFF
Atalla ZMK variant support: OFF
Transaction key support: AUSTRALIAN
User storage key length: SINGLE
Select clear PINs: NO
Enable ZMK translate command: NO
Enable X9.17 for import: YES
Enable X9.17 for export: YES
Solicitation batch size: 1024
Single-DES: ENABLED
Prevent single-DES keys from masquerading as double or triple-length
keys: NO
ZMK length: DOUBLE
Decimalization tables: PLAINTEXT
Decimalization table checks enabled: YES
PIN encryption algorithm: A
Authorized state required when importing DES key under RSA key: YES
Minimum HMAC length in bytes: 10
Enable PKCS#11 import and export for HMAC keys: NO
Enable ANSI X9.17 import and export for HMAC keys: NO
Enable ZEK/TEK encryption of ASCII data or Binary data or None: BINARY
Restrict key check values to 6 hex chars : YES
Enable multiple authorized activities: YES
Enable 2DES LMK encryption of 3DES/2048-bit RSA keys: YES
Enable variable length PIN offset: NO
Enable weak PIN checking: NO
Enable PIN block format 34 as output format for PIN translations to
ZPK: NO
Enable PIN block account number translations: NO
Default LMK identifier: 00
Management LMK identifier: 00
Use HSM clock for date/time validation: YES
Additional padding to disguise key length: NO
Key export and import in trusted format only: NO
Protect MULTOS cipher data checksums: YES
Enforce Atalla variant match to Thales key type: NO
Card/password authorization: C
Enable use of Tokens in PIN Translation: NO
Enable use of Tokens in PIN Verification: NO
Restrict PIN block usage for PCI Compliance: NO
Enforce key type separation for PCI Compliance: NO
Enforce Authorization Time Limit: YES
Overwrite security settings with the settings above? [Y/N]: Y <Return>
SECURITY settings retrieved from smartcard.
```

```
User Action: ENABLED
Audit Counter: 00000183
24 Audited Mgmt commands
0 Audited Host commands
Audit Host Errors: DISABLED
0 Audited Console commands
Overwrite auditlog settings with the settings above? [Y/N]: n <Return>

0 Blocked Host commands
0 Blocked Console commands
Overwrite command settings with the settings above? [Y/N]: n <Return>

Pin Block Format 01: ENABLED
Pin Block Format 02: ENABLED
Pin Block Format 03: ENABLED
Pin Block Format 04: ENABLED
Pin Block Format 05: ENABLED
Pin Block Format 34: ENABLED
Pin Block Format 35: ENABLED
Pin Block Format 41: ENABLED
Pin Block Format 42: ENABLED
Pin Block Format 46: ENABLED
Pin Block Format 47: ENABLED
Overwrite pin block settings with the settings above? [Y/N]: n
<Return>

Secure-AUTH>
```

Key Management Commands

The payShield 9000 HSM provides the following host commands to support generic key management operations:

Command	Page
Generate Key Component (GC)	160
Generate Key and Write Components to Smartcard (GS)	163
Encrypt Clear Component (EC)	167
Form Key from Components (FK)	170
Generate Key (KG)	176
Import Key (IK)	180
Export Key (KE)	183
Generate a Check Value (CK)	186

Generate Key Component

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required Activity: <u>component.{key}.console</u>			

Command: **GC**

Function: To generate a key component and display it in plain and encrypted forms.

	Variant LMK	Keyblock LMK
Authorization:	The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key type code of the key component being generated.	The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key usage code of the key component being generated.
Inputs:	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Length: 1 (single), 2 (double), 3 (triple). • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Scheme: 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Algorithm (if AES LMK): 3DES or AES • Key Length: Single/Double/Triple length DES key or (if AES LMK) 128/192/256-bit AES key. • Key Scheme: • Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual • Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual. • Component Number: 1-9. • Exportability: See the Exportability Table in Chapter 5 of the General Information Manual. • Optional Block data.
Outputs:	<ul style="list-style-type: none"> • Clear text key component. • Key component encrypted under an appropriate variant of the selected LMK. • Component check value. 	<ul style="list-style-type: none"> • Clear text key component. • Keyblock containing the component encrypted under the selected LMK. • Component check value.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
- Invalid key scheme for key length - the Key Scheme is inappropriate for Key length.
- Invalid key scheme - an invalid key scheme is entered.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

Example 1:
(Variant LMK)

This example generates a double length DES key component in plaintext & encrypted form.

```
Online-AUTH> GC <Return>
Enter LMK id: 00 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key type: 001 <Return>
Enter key scheme: U <Return>

Clear Component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
Encrypted Component: UYYYY YYY YYY YYY YYY YYY YYY YYY YYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 2:
(3DES
Keyblock LMK)

This example generates a double length DES key component in plaintext & encrypted form.

```
Online-AUTH> GC <Return>
Enter LMK id: 01 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter component number [1-9]: 2 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: N <Return>

Clear component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
Encrypted component: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 3:
(AES Keyblock
LMK)

This example generates a double length DES key component in plaintext & encrypted form.

```
Online-AUTH> GC <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: 3 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter component number [1-9]: 2 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: N <Return>

Clear component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
Encrypted component: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 4:
(AES Keyblock
LMK)**

This example generates a 128-bit AES key component in plaintext & encrypted form.

```
Online-AUTH> GC <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: A <Return>
Enter key length [128,192,256]: 128 <Return>
Enter key scheme: S <Return>
Enter key usage: K0 <Return>
Enter mode of use: N <Return>
Enter component number [1-9]: 2 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: N <Return>

Clear component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
Encrypted component: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Generate Key and Write Components to Smartcard

Variant <input checked="" type="checkbox"/>	Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Required Activity: <u>component.{key}.console</u>		

Command: **GS**

Function: Generates a key in 2 to 3 component and write the components to smartcards.

	Variant LMK	Keyblock LMK
Authorization:	The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key type code of the key being generated.	The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key usage code of the key being generated.
Inputs:	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Length: 1 (single), 2 (double), 3 (triple). • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Scheme. • Number of components: 2-3. • Smartcard PINs. PINs must be entered within 60 seconds of being requested. 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Algorithm (if AES LMK): 3DES or AES • Key Length: Single/Double/Triple length DES key or (if AES LMK) 128/192/256-bit AES key. • Key Scheme. • Number of components: 2-3. • Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual. • Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual • Key Version Number: 00-99. • Exportability: See the Exportability Table in Chapter 5 of the General Information Manual. • Optional Block data. • Smartcard PINs. PINs must be entered within 60 seconds of being requested.
Outputs:	<ul style="list-style-type: none"> • Key encrypted under an appropriate variant of the selected LMK. • Key check value. 	<ul style="list-style-type: none"> • Keyblock containing the key encrypted under the selected LMK. • Key check value.
Errors:	<ul style="list-style-type: none"> • Invalid LMK identifier - no LMK loaded or entered identifier out of range. • Invalid PIN; re-enter - a PIN of less than 4 or greater than 8 is entered. • Smartcard error; command/return: 0003 - invalid PIN is entered. • Warning - card not blank. Proceed? [Y/N] - the smartcard 	

- entered is not blank.
- Overwrite key component? [Y/N] - the smartcard already contains a key component. It can be overwritten if desired.
- Device write failed - the component could not be verified.
- Invalid key scheme for key length - the Key scheme is inappropriate for Key length.
- Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
- Invalid key scheme - an invalid key scheme is entered.
- Invalid entry - an invalid number of components has been entered.
- Not a LMK card - card formatted for HSM storage or is a license card.
- Card not formatted - card is not formatted.
- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

**Example 1:
(Variant LMK)**

This example writes two double length DES key components to two smartcards, and encrypts the formed key.

```
Online-AUTH> GS <Return>
Enter LMK id: 00 <Return>
Enter key length [1,2,3]: 1 <Return>
Enter key type: 001 <Return>
Enter key scheme: 0 <Return>
Enter number of components [2-3]: 2 <Return>
Insert card 1 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Insert card 2 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Encrypted key: YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 2:
(3DES
Keyblock LMK)**

This example generates and writes two double length 3DES key components to two smartcards, and encrypts the formed key.

```
Online-AUTH> GS <Return>
Enter LMK id: 01 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Enter number of components [2-3]: 2 <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 00 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 00 <Return>
Enter optional block data: L <Return>
Enter more optional blocks? [Y/N]: N <Return>
Insert card 1 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Insert card 2 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 3:
(AES Keyblock
LMK)**

This example generates and writes two double length 3DES key components to two smartcards, and encrypts the formed key.

```
Online-AUTH> GS <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: 3 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Enter number of components [2-3]: 2 <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 00 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 00 <Return>
Enter optional block data: L <Return>
Enter more optional blocks? [Y/N]: N <Return>
Insert card 1 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Insert card 2 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 4:
(AES Keyblock
LMK)**

This example generates and writes two 128-bit AES key components to two smartcards, and encrypts the formed key.

```
Online-AUTH> GS <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: A <Return>
Enter key length [128,192,256]: 128 <Return>
Enter key scheme: S <Return>
Enter number of components [2-3]: 2 <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 00 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 00 <Return>
Enter optional block data: L <Return>
Enter more optional blocks? [Y/N]: N <Return>
Insert card 1 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
```

```
Insert card 2 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Encrypt Clear Component

Variant <input checked="" type="checkbox"/>	Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Required Activity: <u>component.{key}.console</u>		

Command: **EC**

Function: To encrypt a clear text component and display the result at the console.
If the component does not have odd parity, odd parity will be forced before encryption by the selected LMK.

	Variant LMK	Keyblock LMK
Authorization:	The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key type code of the component being encrypted.	The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key usage code of the component being encrypted.
Inputs:	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Scheme. • Clear Component: 16/32/48 hex digits. 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Component Algorithm (if AES LMK): 3DES or AES • Component Length: Single/Double/Triple length DES key or (if AES LMK) 128/192/256-bit AES key. • Key Scheme. • Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual. • Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual. • Component Number: 1-9. • Exportability: See the Exportability Table in Chapter 5 of the General Information Manual. • Optional Block data. • Clear Component: 16/32/48 hex digits.
Outputs:	<ul style="list-style-type: none"> • Component encrypted under an appropriate variant of the selected LMK. • Component check value. 	<ul style="list-style-type: none"> • Keyblock containing the component encrypted under the selected LMK. • Component check value.
Errors:	<ul style="list-style-type: none"> • Invalid LMK identifier - no LMK loaded or entered identifier out of range. • Data invalid; please re-enter - the input data does not contain 16 or 32 or 48 hexadecimal characters. Re-enter the correct number of hexadecimal characters. 	

- Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
- Invalid key scheme - an invalid key scheme is entered.
- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

**Example 1:
(Variant LMK)**

This example encrypts a plaintext double length DES key component.

```
Online-AUTH> EC <Return>
Enter LMK id: 00 <Return>
Enter key type: 001 <Return>
Enter key Scheme: U <Return>
Enter component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Encrypted component: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 2:
(3DES
Keyblock LMK)**

This example encrypts a plaintext double length DES key component.

```
Online-AUTH> EC <Return>
Enter LMK id: 01 <Return>
Enter component length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter component number [1-9]: 2 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 00 <Return>
Enter optional block data: L <Return>
Enter more optional blocks? [Y/N]: N <Return>
Enter component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Encrypted component: S YYYYYYYYY.....YYYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 3:
(AES Keyblock
LMK)**

This example encrypts a plaintext double length DES key component.

```
Online-AUTH> EC <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: 3 <Return>
Enter component length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Enter key usage: D0 <Return>
Enter mode of use: N <Return>
Enter component number [1-9]: 2 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 00 <Return>
Enter optional block data: L <Return>
Enter more optional blocks? [Y/N]: N <Return>
Enter component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Encrypted component: S YYYYYYYYY.....YYYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```


Example 4:
(AES Keyblock
LMK)

This example encrypts a plaintext 128-bit AES key component.

```
Online-AUTH> EC <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: A <Return>
Enter component length [128,192,256]: 128 <Return>
Enter key scheme: S <Return>
Enter key usage: K0 <Return>
Enter mode of use: N <Return>
Enter component number [1-9]: 2 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: N <Return>
Enter optional block identifier: 00 <Return>
Enter optional block data: L <Return>
Enter more optional blocks? [Y/N]: N <Return>
Enter component: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Encrypted component: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Form Key from Components

Variant <input checked="" type="checkbox"/>	Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Required Activity: <u>component.{key}.console</u>		

Command: **FK**

Function: To build a key from components. If clear components are used, they will not be checked for parity, but odd parity will be forced on the final key before encryption under the selected LMK.

	Variant LMK	Keyblock LMK
Authorization:	<p>The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key type code of the key being formed.</p>	<p>The HSM must be in the Authorized State, or the activity <u>component.{key}.console</u> must be authorized, where 'key' is the key usage code of the key being formed.</p>
Inputs:	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Length: 1 (single), 2 (double), 3 (triple). • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Scheme. Must be U, T, or None/Z. • Component Type: X (xor), H (half), E (encrypted), S (smartcard), T (third). • Number of Components: 1-9 if the security setting "Enforce Multiple Key Components" has been set to "NO", otherwise 2-9. • Clear Components: 16/32/48 hex digits. 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Algorithm (if AES LMK): 3DES or AES • Key Length: Single/Double/Triple length DES key or (if AES LMK) 128/192/256-bit AES key. • Key Scheme. • Component Type (for AES keys): X (xor), E (encrypted), S (smartcard), • Component Type (for DES keys): X (xor), E (encrypted), S (smartcard), H (half), T (third). • Number of Components: 1-9 if the security setting "Enforce Multiple Key Components" has been set to "NO", otherwise 2-9. • Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual. • Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual. • Key Version Number: 00-99. • Exportability: See the Exportability Table in Chapter 5 of the General Information Manual. • Optional Block data. • Clear Components: 16/32/48 hex digits.
Outputs:	<ul style="list-style-type: none"> • Key encrypted under an appropriate variant of the selected LMK. • Key Check Value. 	<ul style="list-style-type: none"> • Keyblock containing the component encrypted under the selected LMK. • Key Check Value.

Notes: • PINs must be entered within 60 seconds of being requested.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Incompatible header values - the field values are incompatible between components.
- Incompatible key status optional blocks - there is a mismatch between the values contained in one or more key status optional blocks.
- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Invalid key scheme - an invalid key scheme is entered.
- Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
- Key all zero - the key is invalid.
- Invalid entry - an invalid number of components has been entered.
- Data invalid; please re-enter - the amount of input data is incorrect. Re-enter the correct number of hexadecimal characters.
- Invalid PIN; re-enter - a PIN of less than 4 or greater than 8 is entered.
- Smartcard error; command/return: 0003 - invalid PIN is entered.
- No component card - no key component on the provided smartcard.
- Not a LMK card - card formatted for HSM storage or is a license card.
- Card not formatted - card is not formatted.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

Notes:

- Component type H is not permitted for Triple – DES keys.
- Use of this command will always create an entry in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.

Example 1: (Variant LMK)

This example forms a key from plaintext component.

```
Online-AUTH> FK <Return>
Enter LMK id: 00 <Return>
Enter key length[1,2,3]: 2 <Return>
Enter key type: 002 <Return>
Enter key scheme: U <Return>
Component type [X,H,E,S,T]: X <Return>
Enter number of components [1-9]: 2 <Return>

Enter component 1: *****
<Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>
```

```
Enter component 2: **** **** **** **** **** **** **** ****
<Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: y <Return>

Encrypted key: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 2:
(Variant LMK)

This example forms a key from components on a smartcard.

```
Online-AUTH> FK <Return>
Enter LMK id: 00 <Return>
Enter key length[1,2,3]: 2 <Return>
Enter key type: 002 <Return>
Enter key scheme: U <Return>
Component type [X,H,E,S,T]: S <Return>
Enter number of components (1-9): 2 <Return>

Insert card 1 and enter PIN: ***** <Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>

Insert card 2 and enter PIN: ***** <Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: y <Return>

Encrypted key: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 3:
(Variant LMK)

This example forms a key from encrypted components.

```
Online-AUTH> FK <Return>
Enter LMK id: 00 <Return>
Enter key length[1,2,3]: 2 <Return>
Enter key type: 002 <Return>
Enter key scheme: U <Return>
Component type [X,H,E,S,T]: E <Return>
Enter number of components (1-9): 2 <Return>

Enter component 1: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>

Enter component 2: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: y <Return>

Encrypted key: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 4:
(Variant LMK)**

The security settings require that multiple components are used to form keys, but the user attempts to form a key from one component.

```
Online-AUTH> FK <Return>
Enter LMK id: 00 <Return>
Enter key length[1,2,3]: 2 <Return>
Enter key type: 002 <Return>
Enter key scheme: U <Return>
Component type [X,H,E,S,T]: E <Return>
Enter number of components (2-9): 1 <Return>

Invalid Entry
Enter number of components (2-9): 2 <Return>

Enter component 1: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>

Enter component 2: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: y <Return>

Encrypted key: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 5:
(3DES
Keyblock
LMK)

This example forms a single length DES key from plaintext components.

```
Online-AUTH> FK <Return>
Enter LMK id: 01 <Return>
Enter key length [1,2,3]: 1 <Return>
Enter key scheme: S <Return>
Component type [X,H,E,S,T]: X <Return>
Enter number of components [1-9]: 2 <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 99 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: N <Return>

Enter component 1: **** *  **** *  **** *  **** *   <Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>

Enter component 2: **** *  **** *  **** *  **** *   <Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: y <Return>

Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 6:
(3DES
Keyblock
LMK)

This example forms a double length 3DES key from components on a smartcard.

```
Online-AUTH> FK <Return>
Enter LMK id: 01 <Return>
Enter Key Length[1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Component type [X,H,E,S,T]: S <Return>
Enter number of components (1-9): 2 <Return>

Insert card 1 and enter PIN: ***** <Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>

Insert card 2 and enter PIN: ***** <Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: y <Return>

Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 7:
(AES Keyblock
LMK)

This example forms a double length 3DES key from plaintext components.

```
Online-AUTH> FK <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: 3 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme: S <Return>
Component type [X,H,E,S,T]: X <Return>
Enter number of components [1-9]: 2 <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 99 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: N <Return>

Enter component 1: **** *  **** *  **** *  **** *  **** *  **** *  
<Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: y <Return>
```

```
Enter component 2: **** *  **** *  **** *  **** *  **** *  **** *  ****
<Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: Y <Return>

Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 8:
(AES Keyblock LMK)

This example forms a 128-bit AES key from components on a smartcard.

```
Online-AUTH> FK <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: A <Return>
Enter key length [128,192,256]: 128 <Return>
Enter key scheme: S <Return>
Component type [X,E,S]: S <Return>
Enter number of components [1-9]: 2 <Return>
Enter key version number: 00 <Return>
Enter optional blocks? [Y/N]: N <Return>

Insert card 1 and enter PIN: ***** <Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: Y <Return>

Insert card 2 and enter PIN: ***** <Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: Y <Return>

Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 8:
(AES Keyblock LMK)

This example forms a 128-bit AES key from encrypted components.

```
Online-AUTH> FK <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: A <Return>
Enter key length [128,192,256]: 128 <Return>
Enter key scheme: S <Return>
Component type [X,E,S]: E <Return>
Enter number of components [1-9]: 3 <Return>
Enter key version number: 00 <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 03 <Return>
Enter optional block data: 2005:12:21:00 <Return>
Enter more optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 04 <Return>
Enter optional block data: 2007:12:21:00 <Return>
Enter more optional blocks? [Y/N]: N <Return>

Enter component 1: S XXXXXXXX.....XXXXXX <Return>
Component 1 check value: XXXXXX
Continue? [Y/N]: Y <Return>

Enter component 2: S XXXXXXXX.....XXXXXX <Return>
Component 2 check value: XXXXXX
Continue? [Y/N]: Y <Return>

Enter component 3: S XXXXXXXX.....XXXXXX <Return>
Component 3 check value: XXXXXX
Continue? [Y/N]: Y <Return>

Encrypted key: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Generate Key

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>		Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Variant LMK	Authorization: Determined by KTT(G&E) Activity: <u>generate.{key}.console</u> and <u>export.{key}.console</u>		
Keyblock LMK	Authorization: If export to non-KB. Activity: export.{key}.console		

Command: **KG**

Function: To generate a random key and return it encrypted under the LMK and optionally under a ZMK (for transmission to another party).

Authorization:

Variant LMK	Keyblock LMK												
<p>This command examines the 'Generate' flag of the given key type within the Key Type Table to determine the authorization requirement. If the flag is 'A', the HSM must either be in the Authorized State, or the activity <u>generate.{key}.console</u> must be authorized, where 'key' is the key type code of the key being generated.</p> <p>If the generated key is required to be exported under the ZMK, this command also examines the 'Export' flag of the given key type within the Key Type Table. If the flag is 'A', the HSM must either be in the Authorized State, or the activity <u>export.{key}.console</u> must be authorized, where 'key' is the key type code of the key being exported.</p>	<p>The authorization requirement for this command depends solely on the type of export being requested:</p> <table> <tr> <th>Exported key scheme</th><th>Authorization</th></tr> <tr> <td>No export</td><td>None</td></tr> <tr> <td>'S' (<i>Thales Keyblock</i>)</td><td>None</td></tr> <tr> <td>'R' (<i>TR-31 Keyblock</i>)</td><td>None</td></tr> <tr> <td>'U', 'T' (<i>Variant</i>)</td><td>Required</td></tr> <tr> <td>'Z', 'X', 'Y' (<i>X9.17</i>)</td><td>Required</td></tr> </table> <p>If authorization is required, the HSM must either be in the Authorized State, or the activity <u>export.{key}.console</u> must be authorized, where 'key' is the key usage code of the key being exported.</p>	Exported key scheme	Authorization	No export	None	'S' (<i>Thales Keyblock</i>)	None	'R' (<i>TR-31 Keyblock</i>)	None	'U', 'T' (<i>Variant</i>)	Required	'Z', 'X', 'Y' (<i>X9.17</i>)	Required
Exported key scheme	Authorization												
No export	None												
'S' (<i>Thales Keyblock</i>)	None												
'R' (<i>TR-31 Keyblock</i>)	None												
'U', 'T' (<i>Variant</i>)	Required												
'Z', 'X', 'Y' (<i>X9.17</i>)	Required												
<p>Inputs:</p> <ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Length: 1 (single), 2 (double), 3 (triple). • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Scheme (LMK). • Key Scheme (ZMK) (if exporting). • ZMK (if exporting). • Key block values if exporting to TR-31 format 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Algorithm (if AES LMK): 3DES or AES • Key Length: Single/Double/Triple length DES key or (if AES LMK) 128/192/256-bit AES key. • Key Scheme (LMK). • Key Scheme (ZMK) (if exporting). • ZMK (if exporting). • Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual. 												

Outputs:

- Key encrypted under an appropriate variant of the selected LMK.
- Key/Keyblock encrypted under the ZMK (if exporting).
- Key Check Value.
- Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual.
- Key Version Number: 00-99.
- Exportability: See the Exportability Table in Chapter 5 of the General Information Manual.
- Optional Block data.
- Exportability of exported key (if exporting).
- Keyblock containing the key encrypted under the selected LMK.
- Key/Keyblock encrypted under the ZMK (if exporting).
- Key Check Value.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Must be in Authorized State or Activity not authorized - the key type provided requires the HSM to be in Authorized State.
- Data invalid; please re-enter - the encrypted ZMK does not contain the correct characters, or the key check value does not contain 6 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
- Key parity error; please re-enter - the ZMK does not have odd parity on each byte. Re-enter the encrypted ZMK and check for typographic errors.
- Invalid key scheme for key length - the Key scheme is inappropriate for Key length.
- Invalid key scheme - the key scheme is invalid.
- Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

Example 1:
(Variant LMK)

This example generates a new double length DES key.

```
Online> KG <Return>
Enter LMK id: 00 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key type: 002 <Return>
Enter key scheme (LMK): U <Return>
Enter key scheme (ZMK): <Return>
Enter ZMK: <Return>
Key under LMK: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key Check value: ZZZZZZ
Online>
```

**Example 2:
(Variant LMK)**

This example generates a new double length DES key, and exports it to X9.17 format.

```
Online-AUTH> KG <Return>
Enter LMK id: 00 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key type: 002 <Return>
Enter key scheme (LMK): U <Return>
Enter key scheme (ZMK): X <Return>
Enter ZMK: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Key under LMK: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key under ZMK: X YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 3:
(Variant LMK)**

This example generates a new double length DES key, and exports it to TR-31 format.

```
Online-AUTH> KG <Return>
Enter LMK id: 00 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key type: 001 <Return>
Enter key scheme (LMK): U <Return>
Enter key scheme (ZMK): R <Return>
Enter ZMK: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 44 <Return>
Enter exportability: N <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under LMK: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key under ZMK: R YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

**Example 4:
(3DES
Keyblock LMK)**

This example generates a new double length DES key, and exports it to X9.17 format.

```
Online-AUTH> KG <Return>
Enter LMK id: 01 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme (LMK): S <Return>
Enter key scheme (ZMK): X <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 22 <Return>
Enter exportability: N <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under LMK: S YYYYYYYY.....YYYYYY
Key under ZMK: X YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 5:
(3DES
Keyblock
LMK)

This example generates a new double length DES key, and exports it to TR-31 format.

```
Online> KG <Return>
Enter LMK id: 01 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme (LMK): S <Return>
Enter key scheme (ZMK): R <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key usage: 72 <Return>
Enter mode of use: N <Return>
Enter key version number: 33 <Return>
Enter exportability: E <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 03 <Return>
Enter optional block data: 2005:12:21:00 <Return>
Enter more optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 04 <Return>
Enter optional block data: 2007:12:21:00 <Return>
Enter more optional blocks? [Y/N]: N <Return>
Enter exportability field for exported key block: <Return>
Key under LMK: S YYYYYYYY.....YYYYYY
Key under ZMK: R YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online>
```

Example 6:
(AES Keyblock
LMK)

This example generates a new double length DES key.

```
Online-AUTH> KG <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: 3 <Return>
Enter key length [1,2,3]: 2 <Return>
Enter key scheme (LMK): S <Return>
Enter key scheme (ZMK): <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 00 <Return>
Enter exportability: N <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under LMK: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 7:
(AES Keyblock
LMK)

This example generates a new 128-bit AES key.

```
Online-AUTH> KG <Return>
Enter LMK id: 02 <Return>
Enter algorithm [3DES/AES]: A <Return>
Enter key length [128,192,256]: 128 <Return>
Enter key scheme (LMK): S <Return>
Enter key scheme (ZMK): <Return>
Enter key usage: K0 <Return>
Enter mode of use: N <Return>
Enter key version number: 00 <Return>
Enter exportability: N <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under LMK: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Import Key

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Activity: command.ik.console			

Command: **IK**

Function: To import a key from encryption under a ZMK to encryption under an LMK. If the key imported does not have odd parity a warning will be issued and odd parity will be forced on the key before encryption under the specified LMK.

Authorization: The HSM must either be in the Authorized State, or the activity **command.ik.console** must be authorized.
For AES LMKs, keys can only be exported in Thales Keyblock format.

	Variant LMK	Keyblock LMK
Inputs:	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Scheme (LMK). • ZMK to be used to decrypt the key. • Key/Keyblock to be imported. 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Scheme (LMK). • ZMK to be used to decrypt the key. • Key/Keyblock to be imported. <p>For import from Variant/X9.17:</p> <ul style="list-style-type: none"> • Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual. • Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual. • Key Version Number: 00-99. • Exportability: See the Exportability Table in Chapter 5 of the General Information Manual. • Optional Block data. <p>For import from a keyblock format:</p> <ul style="list-style-type: none"> • Modified Key Usage • Optional Block data.
Outputs:	<ul style="list-style-type: none"> • Key encrypted under an appropriate variant of the selected LMK. • Key Check Value. 	<ul style="list-style-type: none"> • Keyblock containing the key encrypted under the selected LMK. • Key Check Value.
Notes:	<ul style="list-style-type: none"> • Use of this command will always create an entry in the Audit Log – see Chapter 10 of the <i>payShield 9000 General Information Manual</i>. • If the option “Enforce Atalla variant match to Thales key type” is set to YES in the CS console command, the following matchings between Atalla variant and Thales variant key types will be enforced: 	

Key Type	Atalla Variant	Thales Variant (*)	Thales Variant (∅)
TPK	1 or 01	002 LMK 14-15	70D LMK 36-37/7
ZPK		001 LMK 06-07	001 LMK 06-07
ZEK	2 or 02	00B LMK 32-33	00B LMK 32-33
		00A LMK 30-31	00A LMK 30-31
TAK	3 or 03	003 LMK 16-17	003 LMK 16-17
ZAK		008 LMK 26-27	008 LMK 26-27
CVK		402 LMK 14-15/4	402 LMK 14-15/4
TMK	4 or 04	002 LMK 14-15	80D LMK 36-37/8
TPK		002 LMK 14-15	70D LMK 36-37/7
PVK		002 LMK 14-15	002 LMK 14-15
TMK	5 or 05	002 LMK 14-15	80D LMK 36-37/8
BDK type-1	8 or 08	009 LMK 28-29	009 LMK 28-29
MK-AC	9 or 09	109 LMK 28-29/1	109 LMK 28-29/1
MK-SMI	9 or 09	209 LMK 28-29/2	209 LMK 28-29/2
MK-SMC	9 or 09	309 LMK 28-29/3	309 LMK 28-29/3
TEK	26	30B LMK 32-33/3	30B LMK 32-33/3
BDK type-2	30	609 LMK 28-29/6	609 LMK 28-29/6
BDK type-3	8 or 08	809 LMK 28-29/8	809 LMK 28-29/8

* Applies if the security setting "Enforce key type 002 separation for PCI HSM compliance" has the value "N"

∅ Applies if the security setting "Enforce key type 002 separation for PCI HSM compliance" has the value "Y"

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Must be in Authorized State or Activity not authorized - the key type provided requires the HSM to be in Authorized State.
- Data invalid; please re-enter - the encrypted ZMK does not contain the correct characters, or the key check value does not contain 6 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
- Key parity error; re-enter key - the parity of the ZMK is not odd.
- Warning: key parity corrected - the parity of the key encrypted under the ZMK is not odd.
- Invalid key scheme - the key scheme is invalid.
- Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors - the value entered is invalid, or incompatible with previously entered values.

Example 1: (Variant LMK)

This example imports a key from X9.17 format.

```
Online> IK <Return>
Enter LMK id: 00 <Return>
Enter Key type: 002 <Return>
Enter Key Scheme: U <Return>
Enter ZMK: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
```

```
Enter key: X XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Encrypted key: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
```

Example 2:
(Variant LMK)

This example imports a key from TR-31 format.

```
Online> IK <Return>
Enter LMK id: 00 <Return>
Enter key type: 009 <Return>
Enter key scheme (LMK): U <Return>
Enter ZMK: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter key: R XXXXXXXX.....XXXXXX <Return>
Key under LMK: U YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online>
```

Example 3:
(3DES
Keyblock LMK)

This example imports a key from X9.17 format.

```
Online-AUTH> IK <Return>
Enter LMK id: 01 <Return>
Enter key scheme (LMK): S <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key: X XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 27 <Return>
Enter exportability: N <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under LMK: S YYYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 4:
(3DES
Keyblock LMK)

This example imports a key from TR-31 format. Note that a new (more restrictive) value for the imported keyblock's Key Usage field is entered during the import process.

```
Online> IK <Return>
Enter LMK id: 01 <Return>
Enter key scheme (LMK): S <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key: R XXXXXXXX.....XXXXXX <Return>
Enter modified key usage: 72 <Return>
Enter optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 03 <Return>
Enter optional block data: 2005:12:21:00 <Return>
Enter more optional blocks? [Y/N]: Y <Return>
Enter optional block identifier: 04 <Return>
Enter optional block data: 2007:12:21:00 <Return>
Enter more optional blocks? [Y/N]: N <Return>
Key under LMK: S YYYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online>
```

Example 5:
(3DES or AES
Keyblock LMK)

This example imports a key from Thales Keyblock format.

```
Online> IK <Return>
Enter LMK id: 01 <Return>
Enter key scheme (LMK): S <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key: S XXXXXXXX.....XXXXXX <Return>
Enter modified key usage: <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under LMK: S YYYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online>
```

Export Key

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>		Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Variant LMK	Authorization: Determined by KTT(E) Activity: <u>export.{key}.console</u>		
Keyblock LMK	Authorization: If export to non-KB. Activity: <u>export.{key}.console</u>		

Command: **KE**

Function: To translate a key from encryption under the specified LMK to encryption under a ZMK.

	Variant LMK	Keyblock LMK										
Authorization:	<p>This command examines the 'Export' flag of the given key type within the Key Type Table to determine whether authorization is required. If required, the HSM must either be in the Authorized State, or the activity export.{key}.console must be authorized, where 'key' is the key type code of the key being exported.</p>	<p>The authorization requirement for this command depends on the type of export being requested:</p> <table><tr><th>Exported key scheme</th><th>Authorization</th></tr><tr><td>'S' (<i>Thales Keyblock</i>)</td><td>None</td></tr><tr><td>'R' (<i>TR-31 Keyblock</i>)</td><td>None</td></tr><tr><td>'U', 'T' (<i>Variant</i>)</td><td>Required</td></tr><tr><td>'Z', 'X', 'Y' (<i>X9.17</i>)</td><td>Required</td></tr></table> <p>If authorization is required, the HSM must either be in the Authorized State, or the activity export.{key}.console must be authorized, where 'key' is the key usage code of the key being exported.</p> <p>For AES LMKs, keys can only be exported in Thales Keyblock format.</p>	Exported key scheme	Authorization	'S' (<i>Thales Keyblock</i>)	None	'R' (<i>TR-31 Keyblock</i>)	None	'U', 'T' (<i>Variant</i>)	Required	'Z', 'X', 'Y' (<i>X9.17</i>)	Required
Exported key scheme	Authorization											
'S' (<i>Thales Keyblock</i>)	None											
'R' (<i>TR-31 Keyblock</i>)	None											
'U', 'T' (<i>Variant</i>)	Required											
'Z', 'X', 'Y' (<i>X9.17</i>)	Required											
Inputs:	<ul style="list-style-type: none">• LMK Identifier: 00-99.• Key Type: See the Key Type Table in Chapter 4 of the General Information Manual.• Key Scheme (ZMK).• ZMK to be used to encrypt the key.• Key to be exported. <p>For export to Thales Keyblock & TR-31:</p> <ul style="list-style-type: none">• Key Usage: See the Key Usage Table in Chapter 5 of the General Information Manual.	<ul style="list-style-type: none">• LMK Identifier: 00-99.• Key Scheme (ZMK).• ZMK to be used to encrypt the key.• Key to be exported. <p>For export to keyblock format:</p> <ul style="list-style-type: none">• Exportability of exported key.										

	<ul style="list-style-type: none"> • Mode of Use: See the Mode of Use Table in Chapter 5 of the General Information Manual. • Key Version Number: 00-99. • Exportability: See the Exportability Table in Chapter 5 of the General Information Manual. • Optional Block data. <i>Note export from a Variant LMK to Thales Keyblock is not permitted.</i> 	
Outputs:	<ul style="list-style-type: none"> • Key/Keyblock encrypted under the ZMK. • Key Check Value. 	<ul style="list-style-type: none"> • Key/Keyblock encrypted under the ZMK. • Key Check Value.
Errors:	<ul style="list-style-type: none"> • Invalid LMK identifier - no LMK loaded or entered identifier out of range. • Must be in Authorized State or Activity not authorized - the key type provided requires the HSM to be in Authorized State. • Data invalid; please re-enter - the encrypted ZMK or key does not contain 16 or 32 hex or 1 alpha + 32 hex or 1 alpha + 48 hex. Re-enter the correct number of hexadecimal characters. • Key parity error; re-enter key - the ZMK or key does not have odd parity on each byte. Re-enter the key and check for typographic errors. • Invalid key scheme - the key scheme is invalid. • Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual. • Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department. • Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values. 	

Example 1:
(Variant LMK)

This example exports a key to X9.17 format.

```
Online-AUTH> KE <Return>
Enter Key type: 002 <Return>
Enter Key Scheme: X <Return>
Enter ZMK: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter key: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Key under ZMK: X YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 2:
(Variant LMK)

This example exports a key to TR-31 format.

```
Online-AUTH> KE <Return>
Enter LMK id: 00 <Return>
Enter key type: 001 <Return>
Enter key scheme (ZMK): R <Return>
Enter ZMK: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter key: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter key usage: P0 <Return>
Enter mode of use: N <Return>
Enter key version number: 44 <Return>
Enter exportability: N <Return>
Enter optional blocks? [Y/N]: N <Return>
Key under ZMK: R YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 3:
(3DES
Keyblock LMK)

This example exports a key to X9.17 format.

```
Online-AUTH> KE <Return>
Enter LMK id: 01 <Return>
Enter key scheme (ZMK): X <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key: S XXXXXXXX.....XXXXXX <Return>
Key under ZMK: X YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online-AUTH>
```

Example 4:
(3DES
Keyblock LMK)

This example exports a key to TR-31 format.

```
Online> KE <Return>
Enter LMK id: 01 <Return>
Enter key scheme (ZMK): R <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key: S XXXXXXXX.....XXXXXX <Return>
Enter exportability field for exported key block: <Return>
Key under ZMK: R YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online>
```

Example 5:
(3DES or AES
Keyblock LMK)

This example exports a key to Thales Keyblock format.

```
Online> KE <Return>
Enter LMK id: 01 <Return>
Enter key scheme (ZMK): S <Return>
Enter ZMK: S XXXXXXXX.....XXXXXX <Return>
Enter key: S XXXXXXXX.....XXXXXX <Return>
Enter exportability field for exported key block: <Return>
Key under ZMK: S YYYYYYYY.....YYYYYY
Key check value: ZZZZZZ
Online>
```

Generate a Check Value

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>		Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Variant LMK	Authorization: Required if ≠ 6 digits Activity: <u>generate.{key}.console</u>		
Keyblock LMK	Authorization: Not required.		

Command: **CK**

Function: To generate a key check value (KCV) for a key encrypted under a specified LMK.

	Variant LMK	Keyblock LMK
Authorization:	<p>This command only requires authorization when calculating either 8 or 16 digit Key Check Values. If required, the HSM must either be in the Authorized State, or the activity <u>generate.{key}.console</u> must be authorized, where 'key' is the key type of the key being used. Regardless of the authorization requirement, this command examines the 'Generate' flag of the given key type within the Key Type Table to determine whether the check value can be calculated.</p>	<p>The HSM does not require any authorization to run this command. Note: Key Check Values of Keyblocks are always 6-digits in length.</p>
Inputs:	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key Type: See the Key Type Table in Chapter 4 of the General Information Manual. • Key Length: 1 (single), 2 (double), 3 (triple). • Key. 	<ul style="list-style-type: none"> • LMK Identifier: 00-99. • Key.
Outputs:	<ul style="list-style-type: none"> • Key Check Value. 	<ul style="list-style-type: none"> • Key Check Value.

- Errors:
- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
 - Incompatible LMK schemes - the LMK schemes are different.
 - Data invalid; please re-enter - incorrect number of characters.
 - Key parity error; re-enter key - the entered key does not have odd parity on each byte. Re-enter the complete line (key and Key-Type code) and check for typographic errors.
 - Invalid key type; re-enter - the key type is invalid. See the Key Type Table in Chapter 4 of the General Information Manual.
 - Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
 - Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

**Example 1:
(Variant LMK)**

This example generates a check value of a key.

```
Online-AUTH> CK <Return>
Enter LMK id: 00 <Return>
Enter key type code: 001 <Return>
Enter key length flag [S/D/T]: D <Return>
Enter encrypted key: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
<Return>
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

**Example 2:
(Keyblock LMK)**

This example generates a check value of a key.

```
Online> CK <Return>
Enter LMK id: 01 <Return>
Enter key block: S XXXXXXXXXXXXXXXX.....XXXXXXXXXX <Return>
Key check value: ZZZZZZ
Online>
```

Payment System Commands

The payShield 9000 HSM provides the following console commands to support some of the card payment systems host commands.

Command	Page
Generate a Card Verification Value (CV)	189
Generate a VISA PIN Verification Value (PV)	191
Load the Diebold Table (R)	193
Encrypt Decimalization Table (ED)	195
Translate Decimalization Table (TD)	197
Generate a MAC on an IPB (MI)	199

Generate a Card Verification Value

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>misc.console</u>			

Command: **CV**

Function: To generate a VISA CVV or MasterCard CVC.

Authorization: The HSM must be either in the Authorized State, or the activity **misc.console** must be authorized, using the Authorizing Officer cards of the relevant LMK.

Inputs:

- LMK identifier: indicates the LMK to use when decrypting the supplied CVK(s).
- Encrypted CVK
- Primary account number (PAN) for the card: up to 19 decimal digits.
- Card Expiry date: 4 decimal digits.
- Service code: 3 decimal digits.

Outputs:

- Card Verification Value: 3 decimal digits.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Data invalid; please re-enter - possibly incorrect key length. Could also be incorrect PAN, card expiry date, or service code length or non-decimal PAN, card expiry date or service code.
- Key parity error; please re-enter - the parity of the key entered is not odd.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

Notes: Use of this command will always create an entry in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.

**Example 1:
(Variant LMK)**

This example generates a CVV using a CVK pair encrypted in variant format.

```
Online-AUTH> CV <Return>
Enter LMK id: 00 <Return>
Enter key A: XXXX XXXX XXXX XXXX <Return>
Enter key B: XXXX XXXX XXXX XXXX <Return>
Enter PAN: 1234567812345678 <Return>
Enter expiry date: 0694 <Return>
Enter service code: 123 <Return>
CVV: 321
Online-AUTH>
```

**Example 2:
(Variant LMK)**

This example generates a CVV using a double length CVK in variant format.

```
Online-AUTH> CV <Return>
Enter LMK id: 00 <Return>
Enter key A: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter PAN: 1234567812345678 <Return>
Enter expiry date: 0694 <Return>
Enter service code: 123 <Return>
CVV: 321
Online-AUTH>
```

**Example 3:
(Keyblock
LMK)**

This example generates a CVV using a CVK in keyblock format.

```
Online-AUTH> CV <Return>
Enter LMK id: 01 <Return>
Enter key block: S XXXXXXXX.....XXXXXX <Return>
Enter PAN: 1234567812345678 <Return>
Enter expiry date: 0694 <Return>
Enter service code: 123 <Return>
CVV: 321
Online-AUTH>
```

Generate a VISA PIN Verification Value

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>misc.console</u>			

Command: **PV**

Function: To generate a VISA PIN Verification Value (PVV).

Authorization: The HSM must be either in the Authorized State, or the activity **misc.console** must be authorized, using the Authorizing Officer cards of the relevant LMK.

Inputs:

- LMK identifier: indicates the LMK to use when decrypting the supplied PVK(s).
- Encrypted PVK.
- The PVV data block comprising:
 - The 11 right-most digits of the account number (excluding check digit): 11 decimal digits.
 - The PIN verification key indicator (PVKI): 1 decimal digit.
 - The 4 left-most digits of the clear PIN: 4 decimal digits.

Outputs: • The PIN Verification Value (PVV): 4 decimal digits.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Data invalid; please re-enter - the PVK A, PVK B or the PVV data block field is not 16 characters long. Re-enter the correct number of characters.
- Key parity error; please re-enter - the PVK A or PVK B does not have odd parity on each byte. Re-enter the encrypted PVK A or PVK B and check for typographic errors.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.
- Various keyblock field errors – the value entered is invalid, or incompatible with previously entered values.

Notes: • The completion of this activity will always be entered in the audit log irrespective of the AUDITOPTIONS settings,

**Example 1:
(Variant LMK)**

This example generates a PVV using a PVK pair in variant format.

```
Online-AUTH> PV <Return>
Enter LMK id: 00 <Return>
Enter key A: XXXX XXXX XXXX XXXX <Return>
Enter key B: XXXX XXXX XXXX XXXX <Return>
Enter PVV data block: XXXXXXXXXXXX N NNNN <Return>
PVV: NNNN
Online-AUTH>
```

**Example 2:
(Variant LMK)**

This example generates a PVV using a double length PVK in variant format.

```
Online-AUTH> PV <Return>
Enter LMK id: 00 <Return>
Enter key A: U XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter PVV data block: XXXXXXXXXXXX N NNNN <Return>
PVV: NNNN
Online-AUTH>
```

**Example 3:
(Keyblock
LMK)**

This example generates a PVV using a PVK in keyblock format.

```
Online-AUTH> PV <Return>
Enter LMK id: 01 <Return>
Enter key block: S XXXXXXXX.....XXXXXX <Return>
Enter PVV data block: XXXXXXXXXXXX N NNNN <Return>
PVV: NNNN
Online-AUTH>
```


Load the Diebold Table

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>misc.console</u>			

Command: **R**

Function: To load the Diebold table into user storage in the HSM.

Authorization: The HSM must be online and must be either in the Authorized State, or the activity **misc.console** must be authorized, using the Authorizing Officer cards of the relevant LMK.

Inputs:

- LMK identifier: indicates the LMK to use when encrypting the supplied values.
- Location in user storage at which to store the Diebold table. See notes below.

Outputs:

- The 512-character encrypted table: 16 lines of 32 hexadecimal characters each.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Command only allowed from Online-Authorized - the HSM is not online, or the HSM is not authorized to perform this operation, or both.
- Invalid index - the specified location in user storage is out of range. Enter a valid value.
- Data invalid; please re-enter - the entered index is not 3 hexadecimal characters long, or a table entry is not 16 hexadecimal characters long. Re-enter the correct number of hexadecimal characters.
- Invalid table: duplicate or missing values - some of the data entered is not a valid entry for a Diebold table. Check the table and re-enter the data, checking for typographic errors.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Notes:

- Encryption of the Diebold Table:
 - If the security setting "Enforce key type 002 separation for PCI HSM compliance" has the value "N", the Diebold table is encrypted using LMK pair 14-15 variant 0.
 - If the security setting "Enforce key type 002 separation for PCI HSM compliance" has the value "Y", the Diebold table is encrypted using LMK pair 36-37 variant 6.
- User Storage is structured in different ways depending on whether the security setting "User storage key length" has a fixed length value (setting = S(ingle), D(ouble), T(riple)) or is variable (setting = V(ariable)).
 - If the length is fixed, the Diebold table is stored as 32 contiguous blocks of 16 characters. The index for the first block must be in the range 000-FE0.

- If the length is variable, the Diebold table is stored as a single block of 512 characters. Because this needs to use one of the larger slots capable of handling blocks larger than 100 bytes, the index must be in the range 000-07F. See Chapter 16 of the *payShield 9000 General Information Manual* for further information.
- If the security setting "Enforce key type 002 separation for PCI HSM compliance" is changed, the Diebold Table must be re-entered by using this command. Therefore it is important that the cleartext version of the table is retained.

Example:

The security setting "User storage key length" has a fixed length value.

```
Online-AUTH> R <Return>
Enter LMK id: 00 <Return>
Enter index (000 - FE0): XXX <Return>
Now enter table, 16 hex digits/line
Line 01: XXXX XXXX XXXX XXXX <Return>
XXXX XXXX XXXX XXXX OK? [Y/N] Y <Return>
Line 02:
...
...
Line 32: XXXX XXXX XXXX XXXX <Return>
XXXX XXXX XXXX XXXX OK? [Y/N] Y <Return>

XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
...
...
XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX (16 lines of encrypted
table are displayed)
Online-AUTH>
```

Note: The result of the "R" command gives no indication as to the LMK scheme or LMK identifier used in the command. When this value is used with other (host) commands, the user must ensure that the correct LMK is specified in the command.

Encrypt Decimalization Table

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>misc.console</u>			

Command: **ED**

Function: To encrypt a 16 digit decimalization table for use with host commands using IBM 3624 PIN Generation & Verification.

Authorization: The HSM must be either in the Authorized State, or the activity **misc.console** must be authorized, using the Authorizing Officer cards of the relevant LMK.

Inputs:

- LMK identifier: indicates the LMK to use when encrypting the decimalization table.
- Decimalization table. 16 decimal digits that specify the mapping between hexadecimal & decimal numbers.
- The HSM by default checks that the decimalization table contains at least 8 different digits, with no digit repeated more than 4 times. This feature may be disabled using the Configure Security parameter "Enable decimalization table check". Disabling of this feature is not recommended.

Outputs:

- Encrypted decimalization table:
 - 16 Hex characters when using a Variant LMK or a 3DES Keyblock LMK.
 - 32 Hex characters when using an AES LMK.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Not Authorized - the HSM is not authorized to perform this operation.
- Decimalization table invalid - the decimalization table is not all decimal or does not contain at least 8 different digits with no digit repeated more than 4 times.
- Master Key Parity Error - the contents of the HSM storage have been corrupted or erased. Do not continue. Inform the security department.

Example:
(Variant or 3DES Keyblock LMK)

This example encrypts a decimalization table using a Variant LMK (same applies with 3DES Keyblock LMK).

```
Online-AUTH> ED <Return>
Enter LMK id: 00 <Return>
Enter decimalization table: 0123456789012345 <Return>
Encrypted decimalization table: XXXX XXXX XXXX XXXX
Online-AUTH>
```

Example:
(AES Keyblock LMK)

This example encrypts a decimalization table using an AES LMK.

```
Online-AUTH> ED <Return>
Enter LMK id: 00 <Return>
Enter decimalization table: 0123456789012345 <Return>
Encrypted decimalization table: XXXX XXXX XXXX XXXX XXXX XXXX
XXXX XXXX
Online-AUTH>
```

Note:

- The result of the “ED” command gives no indication as to the LMK scheme or LMK identifier used in the command. When this value is used with other (host) commands, the user must ensure that the correct LMK is specified in the command.

Translate Decimalization Table

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>misc.console</u>			

Command: **TD**

Function: To translate an encrypted decimalization table from Encryption under an old LMK to encryption under the corresponding new LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **misc.console** must be authorized, using the Authorizing Officer cards of the relevant LMK.

Inputs:

- LMK identifier: indicates the LMK to use when translating the decimalization table.
- Encrypted Decimalization table. This is the result of encrypting a decimalization table using the ED command. The size of the encrypted decimalization table depends on the LMK used to encrypt it: for DES-based Variant and 3DES Keyblock LMKs, the size is 16 hex digits. For AES Keyblock LMKs, the size is 32 hex digits.
- The HSM by default checks that the decimalization table contains at least 8 different digits, with no digit repeated more than 4 times. This feature may be disabled using the Configure Security parameter "Enable decimalization table check". Disabling of this feature is not recommended.

Outputs:

- Encrypted decimalization table:
 - 16 Hex characters when using a Variant LMK or a 3DES Keyblock LMK.
 - 32 Hex characters when using an AES LMK.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Not Authorized - the HSM is not authorized to perform this operation.
- Decimalization Table Invalid - decimalization table not all decimal or does not contain at least 8 different digits with no digit repeated more than 4 times.
- Master Key Parity Error - the contents of the HSM storage have been corrupted or erased. Do not continue. Inform the security department.
- No LMK in Key Change Storage - Key Change storage is empty.

Example:
(Variant or
3DES
Keyblock
LMK)

```
Online-AUTH> TD <Return>
Enter LMK id: 00 <Return>
Enter decimalization table encrypted under old LMK : XXXXXXXXXXXXXXXXXX
<Return>
Decimalization table encrypted under new LMK      : YYYYYYYYYYYYYYYY
Online-AUTH>
```

Example:
(AES
Keyblock
LMK)

```
Online-AUTH> TD <Return>
Enter LMK id: 00 <Return>
Enter decimalization table encrypted under old LMK : XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXX <Return>
Decimalization table encrypted under new LMK      : YYYYYYYYYYYYYYYY
YYYYYYYYYYYYYYYY
Online-AUTH>
```

Note:

- The result of the "TD" command gives no indication as to the LMK scheme or LMK identifier used in the command. When this value is used with other (host) commands, the user must ensure that the correct LMK is specified in the command.

Generate a MAC on an IPB

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>misc.console</u>			

Command: **MI**

Function: To generate a MAC on the Cryptogram component of a CAP IPB.

Authorization: The HSM must be either in the Authorized State, or the activity **misc.console** must be authorized, using the Authorizing Officer cards of the relevant LMK.

Inputs:

- LMK identifier: indicates the LMK to use when generating the MAC.
- 8 byte IPB represented as 16 hex ASCII characters.

Outputs:

- 4 byte MAC over the plaintext IPB input data.

Errors:

- Invalid LMK identifier - no LMK loaded or entered identifier out of range.
- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- IPB is not 8 bytes. Please re-enter - the validation of the IPB failed.
- Warning: Less than 16 '1'bits in IPB - the IPB contains less than 16 '1' bits.

Example:

```
Online-AUTH> MI <Return>
Enter LMK id: 00 <Return>
Enter IPB: FFFFFFFF00000000 <Return>
          MAC: FB1A 3C1A
Online-AUTH>
```

Note:

- The result of the "MI" command gives no indication as to the LMK scheme or LMK identifier used in the command. When this value is used with other (host) commands, the user must ensure that the correct LMK is specified in the command.

Smartcard Commands

The payShield 9000 HSM provides the following console commands to support HSM smartcards. Please note that some of these commands are designed to operate only with the legacy HSM smartcards while other may support both the legacy and new smartcards used in the payShield Manager.

Command	Page
Format an HSM Smartcard (FC)	201
Create an Authorizing Officer Smartcard (CO)	203
Verify the Contents of a Smartcard (VC)	204
Change a Smartcard PIN (NP)	205
Read Unidentifiable Smartcard Details (RC)	206
Eject a Smartcard (EJECT)	207

NOTE: DO NOT REPEATEDLY ENTER INVALID PINS. A LEGACY SMARTCARD "LOCKS" AFTER EIGHT SUCCESSIVE INVALID PINS HAVE BEEN ENTERED. LEGACY SMARTCARDS CAN BE "UNLOCKED" BY REFORMATTING, WHICH DELETES THE ENTIRE CONTENTS OF THE CARD. NEW SMARTCARDS USED BY THE PAYSHIELD MANAGER LOCK AFTER FIVE SUCCESSIVE INVALID PINS HAVE BEEN ENTERED. THEY MAY BE UNLOCKED BY RECOMMISSIONING THEM.

Format an HSM Smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **FC**

Function: To format an HSM smartcard for use by the HSM. Different formats are used for LMK storage and saving HSM settings. payShield Manager cards do not need to be formatted.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- (LMK cards): Smartcard PIN: 5 to 8 alphanumeric characters.
- Date: 6 numeric character format DDMMYY.
- Time: 6 numeric characters; format hhmmss.
- Issuer ID: maximum 35 alphanumeric characters.
- User ID: maximum 35 alphanumeric characters.

Outputs:

- Text messages:
 - Insert card and press ENTER.
 - Format card for HSM settings/LMKs? [H/L]
 - Enter new PIN for smartcard.
 - Re-enter new PIN.
 - Enter format code.
 - Enter date.
 - Enter time.
 - Enter Issuer ID.
 - Enter User ID.
 - Format complete.
 - Card already formatted, continue? [Y/N].

Note: •This command only operates with legacy HSM smartcards.

Errors:

- Invalid PIN; re-enter - the PIN entered is fewer than 5 or greater than 8 digits.
- PINs did not agree - the new PINs entered for the card did not match each other.
- Invalid input. Entry must be in numeric format - non numeric value is entered for time or date.

Example 1:

```
Online> FC <Return>
Insert card and press ENTER: <Return>
Card already formatted, continue? [Y/N]: Y <Return>
Format card for HSM settings/LMKs? [H/L]: L <Return>
Erasing card
Formatting card . . .
Enter new PIN for Smartcard: ***** <Return>
Re-enter new PIN: ***** <Return>
Enter time [hhmmss]: 153540 <Return>
Enter date [ddmmyy]: 261093 <Return>
Enter User ID: Joe Small <Return>
Enter Issuer ID: Big Bank plc <Return>
Format complete
Online>
```

Example 2:

```
Online> FC <Return>
Insert card and press ENTER: <Return>
Card already formatted, continue? [Y/N]: Y <Return>
Format card for HSM settings/LMKs? [H/L]: H <Return>
Erasing card
Formatting card . . .
Format complete
Online>
```

Create an Authorizing Officer Smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **CO**

Function: To copy the Password for an Authorizing Officer to another smartcard (RLMKs are supported) so that it can be used to set the HSM into the Authorized State. Note that only LMK component cards 1 and 2 contain the Password.

Authorization: The HSM must be in the offline or secure state to run this command.

Inputs:

- Smartcard PIN: 5 to 8 alphanumeric characters. PINs must be entered within 60 seconds of being requested.

Outputs:

- Text messages:
Insert Card for Component Set 1 or 2 and enter the PIN.
Insert Card for Authorizing Officer and enter the PIN.
Copy Complete.

Errors:

- Card not formatted - card not formatted
- Not a LMK card - card formatted for HSM settings or license card.
- Smartcard error; command/return: 0003 - an invalid PIN was entered.
- Invalid PIN; re-enter - PIN is fewer than 5 or greater than 8 digits.
- Card not blank - copy failed.

Example:

```
Offline> co <Return>
Insert Card for Component Set 1 or 2 and enter PIN: *****
<Return>
Insert Card for Authorizing Officer and enter PIN: *****
<Return>
Copy complete.
Offline>
```

Verify the Contents of a Smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **VC**

Function: To verify the contents of the smartcards (RLMKs are supported) held by a Component Holder. The HSM reads the LMK Component Set from the smartcard, computes the check value, compares this with the check value stored on the card and displays the result.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- Smartcard PIN: 5 to 8 alphanumeric characters. PINs must be entered within 60 seconds of being requested.

Outputs:

- Component Set check value:
 - For Variant LMKs, the length of the displayed check value is determined by the CS (Configure Security) setting "Restrict Key Check Value to 6 hex chars".
 - For Keyblock LMKs, the length of the displayed check value is always 6 hex digits.
- Comparison: Pass or Fail.
- Text messages:
 - Check:
 - Compare with card:

Errors:

- Card not formatted - card not formatted
- Not a LMK card - card formatted for HSM settings or license card.
- Smartcard error; command/return: 0003 - an invalid PIN was entered.
- Invalid PIN; re-enter - PIN is fewer than 5 or greater than 8 digits.

Example:

```
Online> vc <Return>
Insert card and enter PIN: ***** <Return>

Scheme: Variant
Check: 012345.
Compare with card: Pass.
Online>
```

If a smartcard is defective or cannot be successfully verified, replace it. Copy a verified smartcard (from the same set of components) onto a replacement.

NOTE: DISPOSE OF THE FAULTY SMARTCARD IN A SECURE MANNER.

Change a Smartcard PIN

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **NP**

Function: To select a new PIN for a smartcard (RACCs and RLMKs are supported) without changing any of the other details stored on the card.
The old PIN must be submitted before a change is effected and the new PIN must be supplied correctly at two consecutive prompts.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- Smartcard PIN: 5 to 8 alphanumeric characters. PINs must be entered within 60 seconds of being requested.

Outputs:

- Text messages:
 - Insert Card and press ENTER.
 - Enter current PIN.
 - Enter new PIN for smartcard.
 - Re-enter new PIN.
 - PIN change completed.

Errors:

- Card not formatted - card not formatted
- Not a LMK card - card formatted for HSM settings or license card.
- Smartcard error; command/return: 0003 - an invalid PIN was entered.
- Invalid PIN; re-enter - PIN is fewer than 5 or greater than 8 digits.
- PINs did not agree - the new PINs entered for the smartcard did not match.

Example:

```
Online> NP <Return>
Insert card and press ENTER: <Return>
Enter current PIN: **** <Return>
Enter new PIN for smartcard: **** <Return>
Re-enter new PIN: **** <Return>
PINs did not agree
Enter new PIN for smartcard: **** <Return>
Re-enter new PIN: **** <Return>
PIN change completed
Online>
```

Read Unidentifiable Smartcard Details

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **RC**

Function: To read otherwise unidentifiable smartcards (RACCs and RLMKs supported).

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- Text messages:
 - Insert Card and press ENTER when ready.
 - This card is formatted for saving and retrieving HSM settings.
 - Version, as stored on card: decimal integer.
 - Date, as stored on card; format: YY/MM/DD.
 - Time, as stored on card; format: hh:mm:ss.
 - User ID, as stored on card; free format alphanumeric.
 - Issuer ID, as stored on card; free format alphanumeric.
 - Data Zone Size, as stored on card: decimal integer.
 - Max Data Free, as stored on card: decimal integer.

Errors:

- Card not formatted - card not formatted
- Not a LMK card - card formatted for HSM settings or license card.

Example 1:

```
Online> RC <Return>
Insert card and press ENTER: <Return>
Format version: 0001
Issue time: 11:53:00
Issue date: 93/10/25
User ID: Bill Weasel
Issuer ID: Big Bank plc
User-data zone size: 0000
Free: 0392
Online>
```

Example 2:

```
Online> RC <Return>
Insert card and press ENTER: <Return>
This card is formatted for saving and retrieving HSM settings.
Online>
```

Eject a Smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **EJECT**

Function: To eject the smartcard from the smartcard reader.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs: None.

Errors: None.

Example: `Online> EJECT <Return>`
`Online>`

DES Calculator Commands

The payShield 9000 HSM provides the following console commands to support the encryption and decryption of data with a given plaintext single, double or triple-length DES key:

Command	Page
Single-Length Key Calculator (N)	209
Double-Length Key Calculator (\$)	210
Triple-Length Key Calculator (T)	211

Single-Length Key Calculator

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **N**

Function: To encrypt and decrypt the given data block with the given single-length key.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- Key (no parity required): 16 hexadecimal characters.
- Data block: 16 hexadecimal characters.

Outputs:

- The data encrypted with the key.
- The data decrypted with the key.

Errors:

- Data invalid; please re-enter - the entered data does not comprise 16 hexadecimal characters. Re-enter the correct number of hexadecimal characters.

Example:

```
Online> N <Return>
Enter key: XXXX XXXX XXXX XXXX <Return>
Enter data: XXXX XXXX XXXX XXXX <Return>
Encrypted: YYYY YYYY YYYY YYYY
Decrypted: YYYY YYYY YYYY YYYY
Online>
```

Double-Length Key Calculator

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **\$**

Function: To encrypt and decrypt the given data block with the given double-length key.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- The double-length key (odd parity is required): 32 hexadecimal characters.
- Data block: 16 hexadecimal characters.

Outputs:

- The data encrypted with the key.
- The data decrypted with the key.

Errors:

- Data invalid; please re-enter - the entered data does not comprise 32 hexadecimal characters. Re-enter the correct number of hexadecimal characters.

Example:

```
Offline> $ <Return>
Enter key: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter data: XXXX XXXX XXXX XXXX <Return>
Encrypted: YYYY YYYY YYYY YYYY
Decrypted: YYYY YYYY YYYY YYYY
Offline>
```

Triple-Length Key Calculator

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **T**

Function: To encrypt and decrypt the given data block with the given triple-length key.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- The triple-length key (odd parity is required): 48 hexadecimal characters.
- Data block: 16 hexadecimal characters.

Outputs:

- The data encrypted with the key.
- The data decrypted with the key.

Errors:

- Data invalid; please re-enter - Re-enter the correct number of hexadecimal characters.

Example:

```
Offline> T <Return>
Enter key: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Single, Double, or Triple length data? (S,D,T): S <Return>
Enter data: XXXX XXXX XXXX XXXX <Return>
Encrypted: YYYY YYYY YYYY YYYY
Decrypted: YYYY YYYY YYYY YYYY
Offline>
```

Legacy Commands

The following console commands are redundant, but are retained for backwards compatibility. They have been superseded by newer (usually more generic) commands – refer to the individual commands for details.

Note: The following commands always use the default LMK, which must be a variant LMK.

Command	Page
Generate a ZMK Component (F)	213
Generate a ZMK & Write to Smartcards (GZ)	214
Encrypt a Clear ZMK Component (Z)	216
Form a ZMK from Encrypted Components (D)	217
Form a Key from Components (BK)	219
Import a CVK or PVK (IV)	221
Generate a Zone PIN Key (B)	223
Translate a Zone PIN Key (WK)	225
Generate a CVK Pair (KA)	226
Translate a CVK Pair from LMK to ZMK (KB)	227
Generate a Double-Length ZMK Component (DD)	228
Form a ZMK from Clear Components (DE)	229
Generate a BDK (DG)	231
Generate & Export a KML (DA)	233
Generate a CSCK (YA)	234
Export a CSCK (YB)	235

Generate a ZMK Component

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **F** (superseded by GC)

Function: To generate a ZMK component and display it in plain and encrypted forms.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- Clear text ZMK component: 16 or 32 hexadecimal characters.
- ZMK component encrypted under a variant of LMK pair 04-05: 16 or 32 hexadecimal characters.
- Component check value; formed by encrypting 64 binary zeros with the component and returning the left-most 24 bits: 6 hexadecimal characters.

Errors:

- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Notes:

- The F, Z and D console commands create and manipulate ZMK components encrypted under LMK 04-05 variant 1. This is to maintain backward compatibility with previous releases of firmware, in which ZMK components were the only types of components supported.
- The recommended method of using ZMK components is to use the GC, EC and FK console commands, which process components of any key type.
- The length of the generated ZMK component will be dictated by the console Configure Security setting "ZMK Length: Single/Double".

Example:

```
Online> F <Return>
Clear ZMK Component: XXXX XXXX XXXX XXXX
Encrypted ZMK Component: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online>
```

Generate a ZMK & Write to Smartcards

Variant <input checked="" type="checkbox"/>	Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Required Activity: <u>generate.000.console</u>		

Command: **GZ** (superseded by GS)

Function: To generate a ZMK in 2 to 9 component and write the components to Smartcards.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **generate.000.console** must be authorized.

Inputs: • Number of components, 1 numeric digit.

Outputs: • ZMK encrypted under LMK pair 04-05: 16 or 32 hexadecimal characters.
• ZMK Check value; formed by encrypting 64 binary zeros with the ZMK; 16 hexadecimal characters, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

Errors: • Command only allowed from Authorized - the HSM is not authorized to perform this operation.
• Invalid PIN; re-enter - the entered PIN is not 4 – 8 digits.
• Smartcard error; command/return: 0003 - invalid PIN is entered.
• Not a LMK card - card formatted for HSM storage or is a license card.
• Card not formatted - card is not formatted.
• Warning - card not blank. Proceed? [Y/N] - the smartcard entered is not blank.
• Overwrite ZMK component? [Y/N] - a ZMK component already exists on the card.
• Invalid entry - invalid number of components entered.
• Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

- Notes:
- The length of the generated ZMK component will be dictated by the console Configure Security setting "ZMK Length: Single/Double".
 - PINs must be entered within 60 seconds of being requested.

Example:

```
Online-AUTH> GZ <Return>
Enter number of components [2-3]: 2 <Return>
Insert card 1 and enter PIN: ***** <Return>
Make additional copies? [Y/N]: N <Return>
Insert card 2 and enter PIN: ***** <Return>
Make additional copies? [Y/N] N <Return>
Encrypted ZMK: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Encrypt a Clear ZMK Component

Variant <input checked="" type="checkbox"/>	Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>
Authorization: Required Activity: <u>component.000.console</u>		

Command: **Z** (superseded by EC)

Function: To encrypt a clear text component and display the result at the console.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **component.000.console** must be authorized.

Inputs: • Clear text ZMK component: 16 or 32 hexadecimal characters.

Outputs: • The ZMK component encrypted under a variant of LMK pair 04-05: 16 or 32 hexadecimal characters.
• Component check value; formed by encrypting 64 binary zeros with the component and returning the left-most 24 bits: 6 hexadecimal characters.

Errors: • Command only allowed from Authorized - the HSM is not authorized to perform this operation.
• Data invalid; please re-enter - the input data does not contain 16 or 32 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
• Component parity error; re-enter component - the entered component does not have odd parity on each byte. Ensure the component has odd parity and re-enter.
• Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Notes: • The F, Z and D console commands create and manipulate ZMK components encrypted under LMK 04-05 variant 1. This is to maintain backward compatibility with previous releases of firmware, in which ZMK components were the only types of components supported.
• The recommended method of using ZMK components is to use the GC, EC and FK console commands, which process components of any key type.
• The length of the generated ZMK component will be dictated by the console Configure Security setting "ZMK Length: Single/Double".

Example:

```
Online-AUTH> Z <Return>
Enter ZMK Component: ***** <Return>
Encrypted ZMK Component: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```


Form a ZMK from Encrypted Components

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required Activity: <u>component.000.console</u>			

Command: **D** (superseded by FK)

Function: To form a ZMK from encrypted components. The components may either be entered from the console or read from Smartcards.
 The manually entered components must have been encrypted using the Z command, or generated using the F command.
 Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **component.000.console** must be authorized.

Inputs:

- Type of input, smartcard or keyboard.
- The number of key components to be entered: 2 to 9.
- The ZMK components, each encrypted under a variant of LMK pair 04-05: 16 or 32 hexadecimal characters.

Outputs:

- The ZMK encrypted under LMK 04-05: 16 or 32 hexadecimal characters.
- The key check value, formed by encrypting 64 binary zeros with the ZMK, and returning all 64 bits: 16 or 32 hexadecimal characters, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

Errors:

- Command only allowed from Authorized - The HSM is not authorized to perform this operation.
- Invalid entry - invalid number of components entered.
- Data invalid; please re-enter - the input data does not contain 16 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
- Component parity error; re-enter component - the entered component does not have odd parity on each byte. Re-enter the encrypted component and check for typographic errors.
- Invalid PIN; re-enter - the entered PIN is not 4 to 8 digits or the PIN does not match the PIN of the card.
- Card checksum mismatch - the components on the cards do not match.
- Smartcard error; command/return: 0003 - invalid PIN is entered.
- Not a LMK card - card formatted for HSM settings or is a license card.
- Card not formatted - card is not formatted.
- No component card - there are no ZMK components on the card.
- Internal failure 12: function aborted - the contents of LMK

storage have been corrupted or erased. Do not continue. Inform the Security Department.

Notes:

- The F, Z and D console commands create and manipulate ZMK components encrypted under LMK 04-05 variant 1. This is to maintain backward compatibility with previous releases of firmware, in which ZMK components were the only types of components supported.
- The recommended method of using ZMK components is to use the GC, EC and FK console commands, which process components of any key type.
- The length of the generated ZMK component will be dictated by the console Configure Security setting "ZMK Length: Single/Double".
- PINs must be entered within 60 seconds of being requested.
- Use of this command will always create an entry in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.

Example 1:

This example forms a ZMK from plaintext components.

```
Online-AUTH> D <Return>
Input components from smartcards? [Y/N]: N <Return>
Enter number of components (2-9): 2 <Return>
Enter encrypted component 1: ***** <Return>
Enter encrypted component 2: ***** <Return>
Encrypted ZMK: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Example 2:

This example forms a ZMK from components on smartcards.

```
Online-AUTH> D <Return>
Input components from smartcards? [Y/N]: Y <Return>
Enter number of components (2-9): 2 <Return>
Insert card 1 and enter PIN: ***** <Return>
Insert card 2 and enter PIN: ***** <Return>
Encrypted ZMK: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Form a Key from Components

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required Activity: <u>component.{key}.console</u>			

Command: **BK** (superseded by FK)

Function: To build a key from clear components. The components are not checked for parity, but odd parity is forced on the final key before encryption under the LMK.
 Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **component.{key}.console** must be authorized, where 'key' is the key type code of the key being formed.

Inputs:

- Key Type; 1 numeric digit:
 "0" - Base Derivation Key (BDK)
 "1" - Card Verification Key (CVK)
 "2" - Zone PIN Key (ZPK)
- The number of key components to be entered: 2 to 9.
- The clear key component. Each BDK component must contain 32 hexadecimal characters and each CVK or ZPK component must contain 16 hexadecimal characters.

Outputs:

- The key formed by exclusive-ORing the entered components, forcing odd parity and encrypting under the appropriate LMK pair:
 - Key type "0" - LMK pair 28 - 29, 32 hexadecimal digits.
 - Key type "1" - LMK pair 14 - 15 variant 4, 16 hexadecimal digits.
 - Key type "2" - LMK pair 06 - 07, 16 hexadecimal digits.
- The key check value, formed by encrypting a block of zeros with the key, and returning all 64 bits: 16 hexadecimal characters, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

Errors:

- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Invalid entry - invalid number of components has been entered.
- Data invalid; please re-enter - the amount of input data is incorrect or non-hexadecimal characters have been entered. Re-enter the correct number of hexadecimal characters.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Notes:

- Use of this command will always create an entry in the Audit Log - see Chapter 10 of the *payShield 9000 General Information Manual*.

Example 1: *This example forms a BDK from components.*

```
Online-AUTH> BK <Return>
Enter key type [0=BDK, 1=CVK, 2=ZPK]: 0 <Return>
Enter number of components (2-9): 2 <Return>
Enter component 1: ***** <Return>
Enter component 2: ***** <Return>

Encrypted key: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Example 2: *This example forms a CVK from components.*

```
Online-AUTH> BK <Return>
Enter key type [0=BDK, 1=CVK, 2=ZPK]: 1 <Return>
Enter number of components (2-9): 3 <Return>
Enter component 1: ***** <Return>
Enter component 2: ***** <Return>
Enter component 3: ***** <Return>

Encrypted key: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Example 3: *This example forms a ZPK from components.*

```
Online-AUTH> BK <Return>
Enter key type [0=BDK, 1=CVK, 2=ZPK]: 2 <Return>
Enter number of components (2-9): 2 <Return>
Enter component 1: ***** <Return>
Enter component 2: ***** <Return>

Encrypted key: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Import a CVK or PVK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required Activity: <u>import.{key}.console</u>			

Command: **IV** (superseded by IK)

Function: To import VISA PVK or CVK from encryption under ZMK to encryption under LMK.
 Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **import.{key}.console** must be authorized, where 'key' is either '402' (CVK) or '002' (PVK).

Inputs:

- ZMK encrypted under LMK pair 04-05: 16 or 32 hexadecimal characters.
- Key type: C or P (for CVK or PVK respectively).
- Key A and B encrypted under the ZMK: 16 hexadecimal characters.
- ZMK variant: 1 or 2 digit, value 0-99 (or <Enter> to ignore). Used only when interworking with Atalla systems. Refer to the CS command. Note that this input is not requested when the ZMK variant support is set to off.

Outputs:

- Key A and B encrypted under LMK 14-15 or variant: 16 hexadecimal characters.
- Key check value: 16 hexadecimal characters, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

Errors:

- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Data invalid; please re-enter - incorrect input data length or invalid ZMK variant.
- Key parity error; re-enter - the ZMK or key entered does not have odd parity.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Notes:

- The completion of this activity will always be entered in the audit log irrespective of the AUDITOPTIONS settings,

Example:

```
Online-AUTH> IV <Return>
Key type [Pvk/Cvk]: C <Return>
Enter ZMK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
(Enter ZMK variant: X <Return>, if enabled by CS command)
Enter key A: XXXXXXXXXXXXXXXXXX <Return>
Enter key B: XXXXXXXXXXXXXXXXXX <Return>
Key A under LMK: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Key B under LMK: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Generate a Zone PIN Key

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **B** (superseded by KG)

Function: To generate a random ZPK and return it encrypted under the LMK and under a ZMK (for transmission to another party). The ZPK can be a VISA Acquirer or Issuer Working key.
Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- The ZMK (VISA Zone Control Master Key, ZCMK) encrypted under LMK pair 04-05 (as generated using the D command): 16 or 32 hexadecimal characters.
- The ZMK key check value (as generated using the D command or by extracting the first 6 digits generated using the CK command): 6 hexadecimal characters.
- The ZMK variant: 1 or 2 digit, value 0-99 (or <Enter> to ignore). Used only when interworking with Atalla systems. Refer to the CS command. Note that this input is not requested when the ZMK variant support is set to off.

Outputs:

- The ZPK encrypted under the ZMK: 16 hexadecimal characters.
- The ZPK encrypted under LMK pair 06-07: 16 hexadecimal characters.
- The ZPK check value, formed by encrypting 64 binary zeros with the ZPK and returning the left-most 48 bits: 12 hexadecimal characters, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

- Errors:
- Data invalid; please re-enter - the encrypted ZMK does not contain 16 or 32 hexadecimal characters, or the key check value is not 6 characters or the ZMK variant is invalid. Re-enter the correct number of hexadecimal characters.
 - Key parity error; re-enter - the ZMK does not have odd parity on each byte. Re-enter the encrypted ZMK and check for typographic errors.
 - Check failed, re-enter check value or abort - invalid 6 character check value has been entered.
 - Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example:

```
Online> B <Return>
Enter encrypted ZMK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter ZMK check value: XXXXXX <Return>
(Enter ZMK variant: X <Return>, if enabled by CS command)
ZPK encrypted for transmission: YYYY YYYY YYYY YYYY
ZPK encrypted for bank: YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online>
```


Translate a Zone PIN Key

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required			
Activity: <u>export.001.console</u>			

Command: **WK** (superseded by KE)

Function: To translate a ZPK from encryption under the LMK to encryption under a ZMK.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **export.001.console** must be authorized.

Inputs:

- ZMK encrypted under LMK pair 04-05: 16 or 32 hexadecimal characters.
- The ZPK encrypted under LMK pair 06-07: 16 hexadecimal characters.
- The ZMK variant: 1 or 2 digit, value 0-99 (or <Enter> to ignore). Used only when interworking with Atalla systems. Refer to the CS command. Note that this input is not requested when the ZMK variant support is set to Off.

Outputs:

- The ZPK encrypted under the ZMK: 16 hexadecimal characters.
- The key check value for the ZPK; generated by encrypting 64 binary zeros with the key: 16 hexadecimal characters, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

Errors:

- Command only allowed from Authorized - the HSM is not authorized to perform this operation.
- Data invalid; please re-enter - the encrypted ZMK does not contain 16 or 32 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
- Key parity error; re-enter key - the ZMK does not have odd parity on each byte. Re-enter the key and check for typographic errors.
- Key parity error - the ZPK does not have odd parity on each byte. Re-enter the key and check for typographic errors.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example:

```
Online-AUTH> WK <Return>
Enter encrypted ZMK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
(Enter ZMK variant: X <Return>, if enabled by CS command)
Enter encrypted ZPK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
ZPK encrypted under ZMK: YYY YYY YYY YYY
Key check value: ZZZZ ZZZZ ZZZZ ZZZZ
Online-AUTH>
```

Generate a CVK Pair

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **KA** (superseded by KG)

Function: To generate a CVK pair and output the key encrypted under a variant of LMK pair 14-15.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs:

- CVK A encrypted under a variant of LMK pair 14-15: 16 hexadecimal characters.
- The key check value for CVK A; formed by encrypting 64 binary zeros with the key and returning the left-most 24 bits: 6 hexadecimal characters.
- CVK B encrypted under a variant of LMK pair 14-15: 16 hexadecimal characters.
- The key check value for CVK B; formed by encrypting 64 binary zeros with the key and returning the left-most 24 bits: 6 hexadecimal characters.

Errors:

- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example:

```
Online> KA <Return>
Encrypted CVK A: YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Encrypted CVK B: YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online>
```

Translate a CVK Pair from LMK to ZMK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **KB** (superseded by KE)

Function: To translate a CVK pair from encryption under a variant of LMK pair 14-15 to encryption under a ZMK.
 Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- CVK A encrypted under a variant of LMK pair 14-15: 16 hexadecimal characters.
- CVK B encrypted under a variant of LMK pair 14-15: 16 hexadecimal characters.
- ZMK encrypted under LMK pair 04-05: 16 or 32 hexadecimal characters.
- The ZMK variant: 1 or 2 digit, value 0-99 (or <Enter> to ignore). Used only when interworking with Atalla systems. Refer to the CS command. Note that this input is not requested when the ZMK variant support is set to off.

Outputs:

- CVK A encrypted under the ZMK.
- The key check value for CVK A, formed by encrypting 64 binary zeros with the key and returning the left-most 24 bits: 6 hexadecimal characters.
- CVK B encrypted under the ZMK.
- The key check value for CVK B, formed by encrypting 64 binary zeros with the key and returning the left-most 24 bits: 6 hexadecimal characters.

Errors:

- Data invalid; please re-enter - the encrypted key does not contain the correct number of hexadecimal characters or an invalid ZMK variant was entered.
- Key parity error - the key does not have odd parity on each byte. Re-enter the key and check for typographic errors.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example:

```
Online> KB <Return>
Enter encrypted CVK A: XXXX XXXX XXXX XXXX <Return>
Enter encrypted CVK B: XXXX XXXX XXXX XXXX <Return>
Enter encrypted ZMK: XXXX XXXX XXXX XXXX <Return>
(Enter ZMK variant: X <Return>, if enabled by CS command)
Encrypted CVK A: YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Encrypted CVK B: YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online>
```

Generate a Double-Length ZMK Component

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **DD** (superseded by GC)

Function: To generate a double-length random ZMK component and display the value at the console screen.
The command ignores the S/D (single/double length) parameter set by the CS (Configure Security) command.
Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs: None.

Outputs: • The clear ZMK component.

Errors: None.

Example:

```
Online> DD <Return>
Clear ZMK component: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Online>
```

Form a ZMK from Clear Components

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Required Activity: <u>component.000.console</u>			

Command: **DE** (superseded by FK)

Function: To enter a ZMK as either two single-length components (halves) or as two to nine double-length components.
 Note: This command will only operate using a variant default LMK.

Authorization: The HSM must be either in the Authorized State, or the activity **component.000.console** must be authorized.

Inputs:

- A half-length or full-length flag.
- The number of components.
- The clear components: each 16 or 32 hexadecimal characters.

Outputs:

- The ZMK encrypted under LMK pair 04-05.
- The key check value (KCV) for the ZMK, if restrict KCV is enabled in the CS command the output will be restricted to the 6 most significant digits with padding zeros for the remainder.

Errors:

- Command only allowed from Authorized - the HSM must be in Authorized State.
- Data invalid; please re-enter - the input data does not contain 16 or 32 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

- Notes:
- The DE command differs from the D command as follows:
 - It uses clear components (not encrypted components).
 - It forms the ZMK from two 16-character halves, or from two to nine 32-character components.
 - When H/F is set to H, two 16-character halves are used: the user is prompted to enter 16 left characters, then 16 right characters. (The unit concatenates the left and right halves).
 - When H/F is set to F, two to nine 32-character components are used: the user is prompted to enter the first component, then the second component, then the third, etc., according to the number of components to be entered. (The unit exclusive-OR combines the 32-character components).
 - The parity of the components is not checked, but the resulting ZMK has odd parity forced before encryption.
 - If the Echo parameter entered in the CS (Configure Security) command has been set to N (on), the clear components are echoed onto the screen as they are entered. If this is not required, either:
 - Use the CS command to set the Echo parameter to F (off); or
 - Enter ^ (i.e. press the Shift and 6 keys) before entering each component.
 - Use of this command will always create an entry in the Audit Log – see Chapter 10 of the *payShield 9000 General Information Manual*.

Example 1:

```
Online-AUTH> DE <Return>
Half or full-length components? (H/F): H <Return>
Enter clear left half: XXXX XXXX XXXX XXXX <Return>
Enter clear right half: XXXX XXXX XXXX XXXX <Return>
Encrypted ZMK: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ
Online-AUTH>
```

Example 2:

```
Online-AUTH> DE <Return>
Half or full-length components? (H/F): F <Return>
Enter number of clear components (2-9): 3 <Return>
Enter component 1: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter component 2: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter component 3: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Encrypted ZMK: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ
Online-AUTH>
```

Generate a BDK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **DG** (superseded by KG)

Function: To generate a random BDK, displaying it encrypted under the LMK pair and under a ZMK, and a BDK check value.
Equivalent to Host BI command

Notes: The command also prompts for a variant. If the recipient requires a variant to the ZMK, enter the appropriate variant number.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- ZMK encrypted under LMK pair 04-05 (generated by the DE command): 32 hexadecimal characters.
- ZMK variant (or <Re ignore). (The command ignores the setting of the Atalla ZMK variant support parameter entered in the CS (Configure Security) command).
- ZMK key check value (generated by the DE command) or the value generated by the console CK command or Host BU command.

Outputs:

- BDK encrypted under the ZMK: 32 hexadecimal characters.
- BDK encrypted under LMK pair 28-29: 32 hexadecimal characters.
- BDK check value.

Errors:

- Data invalid; please re-enter - the encrypted ZMK does not contain 32 hexadecimal characters or the key check value does not contain 8 hexadecimal characters. Re-enter the correct number of hexadecimal characters.
- Key parity error; please re-enter - the entered ZMK does not have odd parity on each byte. Re-enter the encrypted ZMK and check for typographic errors.
- Check failed; re-enter check value or abort - the ZMK check key value is not correct. Re-enter the correct check value.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example:

```
Online> DG <Return>
Enter encrypted ZMK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
Enter ZMK variant: X <Return>
Enter ZMK check value: XXXX XXXX <Return>
BDK encrypted for transmission: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
BDK encrypted under LMK: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZ ZZZZ
Online>
```


Generate & Export a KML

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **DA** (superseded by KG)

Function: To generate a double-length Master Load Key (KML) and return it encrypted under Variant 2 of LMK pair 04-05, and under a Zone Control Master Key (ZCMK). A check value for the KML is also returned.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- ZCMK, encrypted under LMK pair 04-05: 32 hexadecimal characters.
- (Optional) Atalla Variant – 1 or 2 numeric digit; this value is required only if support for Atalla variants is set using the "CS" console command (see Ref.2)

Outputs:

- KML, encrypted under the ZCMK: 32 hexadecimal characters.
- KML, encrypted under Variant 2 of LMK pair 04-05.
- KML check value, formed by encrypting a block of binary zeros with the key and returning the left-most 24 bits of the result: 6 hexadecimal characters.

Errors:

- Data invalid; please re-enter - the entered value does not contain 32 hexadecimal characters or invalid ZMK variant was entered. Re-enter the correct number of characters.
- Key parity error - the plaintext key does not have odd parity on each byte. Re-enter the correct value.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example:

```
Online> DA <Return>
Enter ZMK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
(Enter ZMK variant: V <Return>, if enabled by CS command.)
KML encrypted for transmission: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
KML encrypted under LMK: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Key check value: ZZZZZZ
Online>
```

Generate a CSCK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **YA** (superseded by KG)

Function: Generates a new CSCK and displays it encrypted under the LMK.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs: • A CSCK length flag.

Outputs: • The new CSCK, encrypted under LMK 14-15 variant 4.

Errors: • Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example 1:

```
Online> YA <Return>
Enter CSCK length [S/D]: D <Return>
CSCK encrypted under LMK: YYYY YYYY YYYY YYYY YYYY YYYY YYYY YYYY
Online>
```

Example 2:

```
Online> YA <Return>
Enter CSCK length [S/D]: S <Return>
CSCK encrypted under LMK: YYYY YYYY YYYY YYYY
Online>
```

Export a CSCK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **YB** (superseded by KE)

Function: This command accepts a Zone Master Key (ZMK) and a CSCK encrypted under the LMK. It decrypts and checks parity on both keys, and if correct encrypts the CSCK under the ZMK and displays it.

Note: This command will only operate using a variant default LMK.

Authorization: The HSM does not require any authorization to run this command.

Inputs:

- A flag to indicate the length of the ZMK.
- A ZMK encrypted under LMK 04-05 (generated by the "DE" command), 16/32 hexadecimal characters.
- A ZMK variant (or <Return> to ignore).
- Note: the Atalla variant support parameter (set with the "CS" command) is ignored. CSCK encrypted under LMK 14-15 variant 4, 16/32 hexadecimal characters.

Outputs:

- The CSCK encrypted under the ZMK.
- A Key Check Value (KCV) for the CSCK.

Errors:

- Data invalid; please re-enter - the keys are not 16 or 32 hexadecimal digits in length or invalid ZMK variant was entered.
- Key parity error - the key just entered did not have odd parity; check for typographical errors and re-enter.
- Internal failure 12: function aborted - the contents of LMK storage have been corrupted or erased. Do not continue. Inform the Security Department.

Example 1:

```
Online> YB <Return>
Enter ZMK length [S/D]: D <Return>
Enter ZMK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
(Enter ZMK variant: V <Return>, if enabled by CS command.)
Enter CSCK: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX <Return>
CSCK encrypted for transmission: YYY YYY YYY YYY YYY YYY YYY YYY
Key check value: ZZZZZZ
Online>
```

Example 2:

```
Online> YB <Return>
Enter ZMK length [S/D]: S <Return>
Enter ZMK: XXXX XXXX XXXX XXXX <Return>
(Enter ZMK variant: V <Return>, if enabled by CS command.)
Enter CSCK: XXXX XXXX XXXX XXXX <Return>
CSCK encrypted for transmission: YYY YYY YYY YYY
Key check value: ZZZZZZ
Online>
```

Chapter 5 – payShield Manager

Introduction

This chapter describes the commands used to configure the HSM for use with the payShield Manager.

Note: payShield 9000 HSMs must contain an appropriate license (HSM9-LIC037) before they can be remotely managed.

The payShield 9000 HSM provides the following console commands to support the payShield Manager:

Command	Page
Initialize (RI)	237
Generate an HSM Certificate (RH)	239
Backup Domain Authority Card (RZ)	241
Add a RACC to the whitelist (XA)	242
Decommission the HSM (XD)	243
Remove RACC from the whitelist (XE)	244
Commission the HSM (XH)	245
Generate Customer Trust Anchor (XI)	246
Make an RACC left or right key (XK)	247
Commission a smartcard (XR)	248
Transfer existing LMK to RLMK (XT)	249
Decommission a smartcard (XX)	251
HSM commissioning status (XY)	252
Duplicate CTA share (XZ)	253

From version 2.2a software, the HSM's private key, the certified public key and the Domain Authority self-signed public key certificate are recovered by use of the HSM Master Key (HRK) if a tamper attempt has occurred. Console commands to manage the HRK are included in *Chapter 6 – Secure Host Comms*.

Initialize Domain Authority

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **RI**

Function: To configure the Domain Authority parameter table and generate a Domain Authority RSA key pair and write the results to smart cards.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- Domain Authority parameters (if modified from the default)
- Number of Domain Authority share cards (parameter "n", $3 \leq n \leq 9$)
- Number of shares to recover the Domain Authority private key (parameter "m", $3 \leq m \leq n$)

Outputs:

- Prompts, as above
- Key generation message
- Prompt to enter smart card and PIN
- Continuation message
- Message giving number of copies of public key card
- Summary of Domain Authority information

Errors:

- Invalid value
- Smart card warning – smart card already contains a Domain Authority private key share or a public key certificate

Notes:

- Legacy HSM smartcards are used to store the Domain Authority.
- The Domain Authority private key is broken into a number of shares to be used by the threshold scheme and each share is written to a separate smart card.
- In addition to the Domain Authority private key share, the self-certified Domain Authority public key and the Domain Authority parameter table will be written to each smart card.
- The length of the RSA modulus and the public exponent are determined by the values held in the Domain Authority parameter table.
- The Domain Authority information will be summarized at the end of the operation to ensure that no errors were made.

Example: *This example demonstrates the use of the **RI** console command to generate a Domain Authority consisting of 5 (previously formatted) Domain Authority cards, any 3 of which are required to recover (and therefore use) the Domain Authority's private key.*

```
Secure> RI <Return>
Issuer name: [default = DomAuth]: <Return>
Signature algorithm [RSA]: (press enter) <Return>
Hash Algorithm: [SHA-1, SHA-256 (default = SHA-256)]: <Return>
Domain Authority RSA key length: [1024-2048 (default = 2048)]: <Return>
HSM RSA key length: [1024-2048 (default = 2048)]: 1536 <Return>
Card RSA key length: [1024-2048 (default = 2048)]: 1024 <Return>
Public exponent: [3, 65537 (default = 65537)]: <Return>
Enter number of Domain Authority private key shares: [3-9]: 5 <Return>
Enter number of shares to recover the Domain Authority private key: [3-5]: 3 <Return>
Enter 9 character alpha-numeric Domain Authority serial number : DA0000001
<Return>
Generating Domain Authority key pair ...

Insert first Domain Authority private key card and enter PIN: *****
<Return>
Insert second Domain Authority private key card and enter PIN: *****
<Return>
Insert third Domain Authority private key card and enter PIN: *****
<Return>
Insert fourth Domain Authority private key card and enter PIN: *****
<Return>
Insert fifth Domain Authority private key card and enter PIN: *****
<Return>

Domain Authority generation complete as follows:

Issuer name: DomAuth
Signature algorithm: RSA
Hash Algorithm: SHA-256
Domain Authority RSA key length: 2048
HSM RSA key length: 1536
Card RSA key length: 1024
Public exponent: 65537

Number of Domain Authority private key shares: 5
Number of shares to recover private key: 3

Secure>
```

Generate an HSM Certificate

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **RH**

Function: To generate the HSM's public/private key pair for use with remote management, and produce the HSM's public key certificate (signed by the Domain Authority), and store it inside the HSM.
The HSM's private key, the certified public key and the Domain Authority self-signed public key certificate are stored in secure memory. They are backed up internally when an HSM Master Key (HRK) is installed – see commands SK/SL for details.

Authorization: The HSM can be in any state to run this command.

Inputs: • None.

Outputs: • Prompt to enter smart card and PIN. (NOTE: the PIN must be entered within 60 seconds.)
• Key generation message
• Confirmation message
• Domain Authority Parameter Table (as retrieved from the Domain Authority share cards)

Errors: • Public key error
• Private key error
• Invalid serial number

Notes: • The Domain Authority private key is recovered from "m" share cards. The self-signed Domain Authority public key certificate, the Domain Authority parameter table, and the threshold scheme parameters are read from each card.
• The processing ensures that all "m" Domain Authority share cards contain identical copies of the Domain Authority parameter table and the threshold scheme parameters.
• After the Domain Authority private key is recovered, the Domain Authority Parameter Table is displayed to the user to ensure that the information is correct.
• The HSM generates an RSA key pair and uses the Domain Authority private key to create the HSM's public key certificate. The length of the RSA modulus and the public exponent for the generated key are determined by the values held in the Domain Authority parameter table.

Example:

This example shows the use of the RH command to generate an HSM's certificate. In this example, 3 shares are required to recover the Domain Authority private key.

```
Online> RH <Return>
```

```
Insert Domain Authority private key card and enter PIN: ***** <Return>
```

```
Insert another Domain Authority private key card and enter PIN: *****  
<Return>
```

```
Insert another Domain Authority private key card and enter PIN: *****  
<Return>
```

```
Domain Authority parameters as follows:
```

```
Issuer name: CertAuth  
Signature algorithm: RSA  
Hash Algorithm: SHA-256  
Domain Authority RSA key length: 2048  
HSM RSA key length: 1536  
Card RSA key length: 1024  
Public exponent: 65537
```

```
Continue generating HSM Certificate using the above Domain Authority  
parameters [Y/N]: Y <Return>
```

```
Generating HSM key pair ...
```

```
HSM certificate generated and stored.
```

```
Online>
```


Backup Domain Authority Card

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>		Secure <input checked="" type="checkbox"/>
Authorization: Not required			

Command: **RZ**

Function: To backup an existing Domain Authority card that was previously created using the RI console command.

Authorization: The HSM must be in the Secure state to run this command.

Inputs: • None

Outputs: • Prompts to enter smart cards
• Prompt to enter number of backup cards

Errors: • Not in Secure state
• Invalid PIN; re-enter
• Card not formatted or card inserted incorrectly

Example: *This example shows the use of the RZ command to create a backup of an existing Domain Authority card.*

```
Secure> RZ <Return>
```

```
Insert Domain Authority component card to be copied and enter PIN: ****
```

```
<Return>
```

```
Enter number of back-up cards required: 1 <Return>
```

```
Insert Domain Authority component card to be written to and enter PIN: ****
```

```
<Return>
```

```
Secure>
```

Add a RACC to the whitelist

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XA**

Function: To add a RACC to the whitelist on the HSM.

Authorization: The HSM must be in Secure state to run this command.

Inputs: •None

Outputs: •None

Example 1:

```
Secure> XA <Return>

Insert payShield Manager Smartcard and press ENTER: <Return>
Enter PIN: ***** <Return>

Do you want to add card XYZ123 to the whitelist? y <Return>

Card XYZ123 added to whitelist.

Secure>
```

Decommission the HSM

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XD**

Function: To decommission the HSM by deleting the payShield Managers keys and groups.

Authorization: The HSM must be in Secure state to run this command.

Inputs: • None

Outputs: • None

Example 1:

```
Secure> XD <Return>
Do you want to erase the payShield Manager's keys and groups? [Y/N]: Y
<Return>
Secure>
```

Remove RACC from the whitelist

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XE**

Function: To remove an RACC from the whitelist.

Authorization: The HSM must be in Secure state to run this command.

Inputs: •None

Outputs: •None

Example 1:

```
Secure> XE <Return>

Choice  ID      Type
  1     ABC321   restricted
  2     XYZ123   restricted
Which RACC do you want to remove? 1 <Return>

Card ABC321 removed from whitelist

Secure>
```

Commission the HSM

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XH**

Function: To commission the HSM

Authorization: The HSM must be in Secure state to run this command.

Inputs: • None

Outputs: • None

Example 1:

```
Secure> XH <Return>

Please have all Customer Trust Anchor (CTA) payShield Manager
smartcards available
Insert first CTA payShield Manager Smartcard and press ENTER: <Return>
Enter PIN: ***** <Return>
Insert CTA payShield Manager Smartcard 2 of 3 and press ENTER: <Return>
Enter PIN: ***** <Return>
Insert CTA payShield Manager Smartcard 3 of 3 and press ENTER: <Return>
Enter PIN: ***** <Return>

Starting the commissioning of the HSM process...
Please insert left key card and press ENTER: <Return>
Enter PIN: ***** <Return>
Please insert right key card and press ENTER: <Return>
Enter PIN: ***** <Return>

Successfully commissioned HSM

Secure>
```

Generate Customer Trust Anchor

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XI**

Function: Generates the Customer Trust Anchor and stores them on smartcards.

Authorization: The HSM must be in Secure state to run this command.

Inputs:

- Country
- State
- Locality
- Organization
- Organizational Unit
- Common Name
- Email
- Number of private shares
- Number of shares needed to recover private key

Outputs: • None

Example 1:

Secure> **XI** <Return>

```
Please enter the certificate Subject information:
Country Name (2 letter code) [US]: US <Return>
State or Province Name (full name) []: Florida <Return>
Locality Name (eg, city) []: Plantation <Return>
Organization Name (eg, company) []: Thales <Return>
Organizational Unit Name (eg, section) []: Production <Return>
Common Name (e.g. server FQDN or YOUR name) [CTA]: CTA <Return>
Email Address []: info@thalessec.com <Return>

Enter number of Customer Trust Authority private key shares [3-9]: 3
<Return>
Enter number of shares to recover the Customer Trust Authority private
key [3-3]: 3 <Return>

Issued to: CTA, Issued by: CTA
Validity : Jan  9 10:28:49 2015 GMT to Jan  3 10:28:49 2040 GMT
Unique ID: EE3CB7CE8343B464CC04278188CF7EB3 - 3DE05514 (Root)

Insert payShield Manager Smartcard 1 of 3 and press ENTER: <Return>
Enter new PIN for smartcard: ***** <Return>
Re-enter new PIN: ***** <Return>
Working....
CTA share written to smartcard.

Insert payShield Manager Smartcard 2 of 3 and press ENTER: <Return>
Enter new PIN for smartcard: ***** <Return>
Re-enter new PIN: ***** <Return>
Working....
CTA share written to smartcard.

Insert payShield Manager Smartcard 3 of 3 and press ENTER: <Return>
Enter new PIN for smartcard: ***** <Return>
Re-enter new PIN: ***** <Return>
Working....
CTA share written to smartcard.

Successfully generated a Customer Trust Anchor
Secure>
```

Make an RACC left or right key

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XK**

Function: Defines a RACC as either a left or right key in the whitelist on the HSM.

Authorization: The HSM must be in Secure state to run this command.

Inputs: Left or Right (card type)

Outputs: •None

Example 1:

```
Secure> XK <Return>
```

```
Insert payShield Manager Smartcard and press ENTER: <Return>
```

```
Enter PIN: ***** <Return>
```

```
Do you want to make ABC321 a [L]eft or [R]ight key? L <Return>
```

```
Card ABC321 is now a left key.
```

```
Secure>
```

Commission a smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XR**

Function: To commission a smartcard.

Authorization: The HSM must be in Secure state to run this command.

Inputs: •None

Outputs: •None

Example 1: Secure> **XR** <Return>

```
Please have all Customer Trust Anchor (CTA) payShield Manager
smartcards available
Insert first CTA payShield Manager Smartcard and press ENTER: <Return>
Enter PIN: *****
Insert CTA payShield Manager Smartcard 2 of 3 and press ENTER: <Return>
Enter PIN: *****
Insert CTA payShield Manager Smartcard 3 of 3 and press ENTER: <Return>
Enter PIN: *****
Enforce a PIN change on first use? [Y/N]: N <Return>
Insert a payShield Manager Smartcard to be commissioned and press
ENTER: <Return>
Enter new PIN for smartcard: ***** <Return>
Re-enter new PIN: ***** <Return>
Do you wish to add the smartcard A3 to the HSM whitelist [Y/N]: Y
<Return>
Assign smartcard as a Left or Right Key RACC? [L/R/N]: N <Return>
Would you like to commission another card? [Y/N]: N <Return>

Secure>
```


Transfer existing LMK to RLMK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XT**

Function: To transfer an existing HSM LMK stored on legacy smartcards to payShield Manager RLMK cards for use through the payShield Manager.

In order to transfer a Variant LMK you will be required to fully reassemble the LMK (bring all the components together). Then, the fully formed Variant LMK is split among shares onto the pre-comissioned payShield Manager RLMK cards.

For Keyblock LMKs, they are not stored as components on non-payShield Manager smart cards, but as shares. However, you must bring a quorum of share holders together, reconstitute the LMK, and then split it among shares onto the pre-comissioned payShield Manager RLMK cards.

Authorization: The HSM must be in Secure state to run this command.

Inputs:

- Number of shares to split LMK into
- Number of Components required to reconstitute LMK

Outputs:

- None

Example 1:

```
Secure> XT <Return>

Please have all the local LMK components and enough commissioned RACCs
to receive the LMK ready.

Insert card and press ENTER: <Return>
Enter PIN: ***** <Return>

Check: 268604
Load more components? [Y/N]: N <Return>

LMK Check:      268604
LMK key scheme: Variant
LMK algorithm:  3DES(2key)
LMK status:     Test

Is this the LMK you wish to transfer? [Y/N]: Y <Return>

Enter the number of shares to split the LMK into: [2-9]: 2 <Return>
The number of shares required to reconstitute the LMK is fixed for
variants: 2 <Return>

Insert a commissioned card 1 of 2 and press ENTER: <Return>
Enter PIN: ***** <Return>

Card Check: E0CBF4
LMK share written to smartcard.

Insert a commissioned card 2 of 2 and press ENTER: <Return>
Enter PIN: ***** <Return>

Card Check: E0CBF4
LMK share written to smartcard.
Want to test the reassembly of the LMK? Y <Return>

Please have all the RLMK shares ready
Insert RLMK card and press ENTER: <Return>
Enter PIN: ***** <Return>
```

```
LMK share 1 read (1 of 2) Card Check: E0CBF4
Insert RLMK card and press ENTER: <Return>
Enter PIN: ***** <Return>
LMK share 2 read (2 of 2) Card Check: E0CBF4

LMK Check 268604

Secure>
```

Decommission a smartcard

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XX**

Function: To decommission a payShield Manager smartcard.

Authorization: The HSM may be in any state to run this command.

Inputs: •None

Outputs: •None

Example 1:

```
Secure> XX <Return>

Please insert card to decommission and press ENTER: <Return>
Warning: Resetting a payShield Manager Smartcard to its original state
will erase all key material from the card.

Are you sure? [Y/N]: Y <Return>

payShield Manager Smartcard successfully decommissioned
Would you like to decommission another card? [Y/N]: N <Return>

Secure>
```

HSM commissioning status

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XY**

Function: To show the state of the HSM Management commissioning and whitelist.

Authorization: The HSM may be in any state to run this command.

Inputs: •None

Outputs:

- Customer Trust Anchor installed
- HSM Public Key installed
- Is HRK password user defined
- Is HRK available for use
- Authorized RACCs

Example 1:

```
Secure> XY <Return>

Customer Trust Anchor Installed      : Yes
  1 - Issued to: CTA, Issued by: CTA
  Validity : Dec 11 16:20:17 2014 GMT to Dec  5 16:20:17 2039 GMT
  Unique ID: A86AF14A28253F313B00516875E69C9B - 21722E26 (Root)

HSM Public Key Certificate Installed : Yes
  2 - Issued to: A4665275330S, Issued by: CTA
  Validity : Jan  9 10:44:20 2015 GMT to Jan  3 10:44:20 2040 GMT
  Unique ID: 99734BD96B59EFF036B8218FD3DA2EDD - 21722E26

Is HRK passphrase user defined      : No
Is HRK available for use             : Yes

Authorized RACCs                     : 4
ID                                   RACC Type
ABC321                              left key
SCA00000001                          left key
SCB00000001                          right key
XYZ123                              restricted

Secure>
```

Duplicate CTA share

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **XZ**

Function: To duplicate a CTA share smartcard.

Authorization: The HSM must be in Secure state to run this command.

Inputs: •None

Outputs: •None

Example 1: Secure> **XZ** <Return>

```

Insert a CTA share payShield Manager Smartcard to be duplicated:
Enter PIN: ***** <Return>
Working...
Please insert a commissioned payShield Manager smartcard and press
ENTER: <Return>
Enter PIN: ***** <Return>
Working...
CTA share written to smartcard.

Secure>
    
```

Chapter 6 – Secure Host Comms

Introduction

This chapter describes the commands used to configure a payShield 9000 HSM such that the host connection is protected using TLS/SSL (known as Secure Host Communications). A description of the payShield 9000 Secure Host Communications capability is provided in Chapter 14 of the *payShield 9000 General Information Manual*.

The Certificate Requests and Certificates may be stored on / loaded from a regular USB memory stick.

The required format for the USB memory stick is FAT32. The Operating System used in the payShield 9000 supports most types of USB memory stick, but may not have the drivers for some of the newer types. If difficulties are experienced when trying to read from or write to a USB device, an alternative memory stick should be used.

The HSM's certificate signing request (CSR) structure is compliant with PKCS#10. The client must use the same key type as is included in the HSM's CSR.

The HSM uses certificate formats compliant with X.509.

Note: payShield 9000 HSMs must contain an appropriate license (HSM9-LIC036) before the host connection can use TLS/SSL.

The payShield 9000 HSM provides the following console commands to manage the HSM's private key, the certified public key and the CA self-signed public key certificate to support secure host communications:

Command	Page
Generate Certificate Signing Request (SG)	255
Import Certificate (SI)	258
Export HSM Certificate's Chain of Trust (SE)	260
View Installed Certificate(s) (SV)	262
Delete Installed Certificate(s) (SD)	264
Generate HRK (SK)	265
Change HRK Passphrase (SP)	266
Restore HRK (SL)	267

The HRK is also required to allow recovery of the HSM's private key, the certified public key and the CA self-signed public key certificate used for payShield Manager - see *Chapter 5 – payShield Manager*.

Generate Certificate Signing Request

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SG**

Function: To generate the HSM's public/private key pair for use with secure host communications, and extract the public key in the form of a Certificate Signing Request (.CSR).
The private key is stored in tamper protected memory. It is backed up internally using the HSM Master Key (HRK) – see commands SK for details.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- Certificate fields (Country, State, Locality, Org Name, Org Unit Name, Common Name, E-mail Address).
- Key Type (RSA, ECDSA)
- Filename when saving to USB memory stick

Outputs:

- Prompts, as above
- Key generation message
- Prompt to save to USB memory stick
- Certificate Signing Request

Errors:

- File exists – replace?

Notes:

- See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.
- The HRK must be installed (using the SK console command) prior to using this command.
- The exported file will automatically have the extension ".CSR".
- The size of RSA keys used is 2048-bits.
- The size of ECDSA keys used is either 256-bits, 384-bits or 521-bits (user selectable).
- The client must use the same RSA/ECDSA key type as is included in the HSM's CSR.
- A maximum certificate chain length of 6 is supported.
- The required format for the USB memory stick is FAT32. The Operating System used in the payShield 9000 supports most types of USB memory stick, but may not have the drivers for some of the newer types. If difficulties are experienced when trying to read from or write to a USB device, an alternative memory stick should be used.

Example 1: *This example demonstrates the use of the **SG** console command to generate a 521-bit ECDSA public/private key pair and output a certificate signing request.*

```
Secure> SG <Return>
Please enter the Subject Information for the Certificate Request:

Country Name (2 letter code) []: UK <Return>
State or Province Name (full name) []: Greater London <Return>
Locality Name (eg, city) []: London <Return>
Organization Name (eg, company) []: Bank XYZ <Return>
Organizational Unit Name (eg, section) []: Operations <Return>
Common Name (e.g. server FQDN or YOUR name) []: HSM-0001 <Return>
Email Address []: bill@bankxyz.com <Return>

Select key type:
1 - RSA
2 - ECDSA P-256
3 - ECDSA P-384
4 - ECDSA P-521
Type [4]: 4 <Return>

Generating key pair .....+++
.....+++
DONE

Do you wish to save to a file [Y/N]: Y <Return>
Enter filename: HSM-0001 <Return>
-----BEGIN CERTIFICATE REQUEST-----
MIIC2TCCAcECAQAwZMxCZAJBgNVBAYTAlVLMRcwFQYDVQQIEw5HcmVhdGVyIEExv
bmRvbWJEPMA0GA1UEBxMGTG9uZG9uMREwDwYDVQQKEWhCYW5rIFhZWJETMBEGA1UE
CxMKT3BlcmF0aW9uczERMA8GA1UEAxMISFNNLTAwMDIxHAdBgqhkiG9w0BCQEW
EGJpbGxAYmFua3h5ei5jb20wggeiMA0GCSpGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQc+JhIisca5k7l5YIRNcDcq/QMb3jHzhQIbME409zDhTtmINFM7YrvZ6N2Sy1TU
za1cPf9JKR2X5D3ukaICTkTwArjlWRnU2UnINTYe00RWeBaoux04ijSvzx5mCCg
RtcSQDK748+0xgWlZezkKkv+akOh4vYPdiOKx47wiS7UAENBaQI14C5cbnj6JMLe
f3hmzQzzu3vACAIDbuQXZ5A7w7ecGLSLahjEyx1H7PXpLnul2lPRlBcemVdqHi8f
dfXTAKE1RrKSrvU22sOn6uQLGFRTseIuC4tFvtZNRHAtqCYpabV4vrBmNQDaw8W
p2FFu+e71ybgSLY0R5xt7ZABAgMBAAGgADANBgkqhkiG9w0BAQUFAAOCAQEAvZS
iy5gJkAjUdqaBjr5MUoAXvk15fEg6gO+SV39X3mSsQklQdoHwFSNGOUWYHkTKPvN
vZnCxm1UK2nBhlu2Xz44yC/U7+E7FsaQz2nXrNx/gF3SY/a/ODA+Y9iSERIpwRCM
9CKapYONeBHqK/NicgTOZ3SMsC9JXsvtxPyQ7vmbu4a/JpMantWfclCA+z6i+S+H
WavGnPVGt9ERD5Cij7B6qSbbrkn+xoJARIGsXhbVQmdSxR8I8HUAQDYV+2VJo3ba
ct9ubVjaw2SSiQZp9xB7BOJjk/NQrTk5gG3BkDI/Ukp9A9s7YoW1oMY8YdIg/YRo
Y+LI5trvXN73V2X0Ow==
-----END CERTIFICATE REQUEST-----

Secure>
```


Example 2: *This example demonstrates the use of the **SG** console command to generate a 2048-bit RSA public/private key pair and output a certificate signing request.*

```
Secure> SG <Return>
Please enter the Subject Information for the Certificate Request:

Country Name (2 letter code) []: UK <Return>
State or Province Name (full name) []: Greater London <Return>
Locality Name (eg, city) []: London <Return>
Organization Name (eg, company) []: Bank XYZ <Return>
Organizational Unit Name (eg, section) []: Operations <Return>
Common Name (e.g. server FQDN or YOUR name) []: HSM-0002 <Return>
Email Address []: bill@bankxyz.com <Return>

Select key type:
1 - RSA
2 - ECDSA P-256
3 - ECDSA P-384
4 - ECDSA P-521
Type [4]: 1 <Return>

Generating key pair .....+++
.....+++
DONE

Do you wish to save to a file [Y/N]: Y <Return>
Enter filename: HSM-0002 <Return>
-----BEGIN CERTIFICATE REQUEST-----
MIIC2TCCAcECAQAwZMxCZAJBgNVBAYTA1VLMRcwFQYDVQQIEw5HcmVhdGVyIEExv
bmRvbWJEPMA0GA1UEBxMGTG9uZG9uMREwDwYDVQQKEWhCYW5rIFhZWJETMBEGA1UE
CxMKT3BlcmF0aW9uc2ERMA8GA1UEAxMISFNNLTAwMDIxH2AdBgkqhkiG9w0BCQEW
EGJpbGxAYmFua3h5ei5jb20wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQDBJAjJVtpE2Covk13BpZCACN6hUoQeLRv62+M3Lioa/ckvrIDaFxRTmLBGAof/
nZR3uRXSRz5oo3MX+fG4QXuLcGuJFPHUfdnJRFIGNxoxkrrXn5OyxtokLwdE3HrK
VgKeUPQvDluZVXCbfJ1rGGaBk6bRQCfb7hBI7gcb6NfLIPms/bXYgy5hKUBkf+N
rMGtKAHz70E7BRMyY95GFo6nDne579rUi8RDxC4vqIJgkaXbuv4evYx1liTsQ69O
wr0iRSygyHSHYzA8TVcwJ1pNTO1Jeg2xJ8r4axs0r5IKxxpD2PDav4DdyQ0TsZkTB
QfSxPnlD4sTeQW5s42Y0B02ZAgMBAAGgADANBgkqhkiG9w0BAQUFAAOCAQEAJqPX
alHvtQKsfxgzTn2nWiw/v/9v8Qs1lMIRJ5/Y3x+fdRSSK55uwPmRIRlCYdM0xQ4C
tSW3jWUib1P0a3XxC5O4cWfbXJSxWkoSiN6V5gZrCI9W1z05xAuJZtjdVcFbUvVI
pPw3LXXS2CxAsAbgtz3QG+MIdyiicE5vUN2kKxhhZac8Ev3tpy2Uue8XGylsDybu
8qx5I5tMUSAsYx4M956gJEL0Mt9k8phIhsbKz5IKDDEwuyurJlYoOqkVVZeuBKZu
YKJKdOtwwzzuUesEcGQfbAleBR0nteZm0irWJRaCXeyg0e5DF0FfWGIE08ojx4dvh
w3mX7lZX4RGchVESYQ==
-----END CERTIFICATE REQUEST-----

Secure>
```

Import Certificate

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SI**

Function: To import a certificate for storage inside the HSM for use with secure host communications.

The certificate may be one of the following:

- HSM certificate
- Client certificate
- Sub-CA certificate (for either HSM or client)
- Root-CA certificate (for either HSM or client)

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- File selection
- Prompt for import of additional certificates

Outputs:

- Prompts, as above
- Filenames of certificates on USB memory stick
- Summary of imported certificate (Issued to/by, Validity, ID)
- Chain of Trust statement (for an HSM certificate)

Notes:

- See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.
- The HSM's public/private key pair must be installed (using the SG console command) prior to using this command.
- The file(s) to be imported must have the extension ".CRT".
- A maximum certificate chain length of 6 is supported.
- The required format for the USB memory stick is FAT32. The Operating System used in the payShield 9000 supports most types of USB memory stick, but may not have the drivers for some of the newer types. If difficulties are experienced when trying to read from or write to a USB device, an alternative memory stick should be used.

Example 1: *This example demonstrates the use of the **SI** console command to import the root CA certificate (that signed the HSM's certificate) into the HSM.*

```
Secure> SI <Return>

Select File
  1 - HSM-0001.crt
  2 - BankXYZRootCA.crt
  3 - Client.crt
  4 - ClientRootCA.crt
File: 2 <Return>

Imported Trusted CA Certificate
  Issued to: Bank XYZ, Issued by: Bank XYZ
  Validity : May 9 10:59:22 2013 GMT to May 7 10:59:22 2023 GMT
  Unique ID: 9C8FC713FAA31010 - AC03FAD5 (Root)

Do you wish to import another certificate? N <Return>

Secure>
```

Example 2: *This example demonstrates the use of the **SI** console command to import the HSM's (now signed) certificate back into the HSM.
(Note that the root CA certificate has already been installed (see Example 1), and so the HSM indicates that the "Chain of Trust" is complete.*

```
Secure> SI <Return>

Select File
  1 - HSM-0001.crt
  2 - BankXYZRootCA.crt
  3 - Client.crt
  4 - ClientRootCA.crt
File: 1 <Return>

Imported CA-signed HSM Certificate
  Issued to: HSM-0001, Issued by: Bank XYZ
  Validity : May 21 15:05:51 2013 GMT to May 21 15:05:51 2014 GMT
  Unique ID: 2050 - AC03FAD5

Chain of Trust validated
  Bank XYZ (Root)

Do you wish to import another certificate? N <Return>

Secure>
```

Export HSM Certificate's Chain of Trust

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SE**

Function: To export the HSM certificate's chain of trust (i.e. the chain of certificates required to authenticate the HSM's certificate, up to and including the root CA certificate).

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- Filename when saving to USB memory stick

Outputs:

- Prompts, as above
- Prompt to save to USB memory stick
- Certificate Chain of Trust is displayed at the console, and (if requested) saved to the USB memory stick

Errors:

- File exists – replace?

Notes:

- See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.
- The HSM's public/private key pair must be installed (using the SG console command) prior to using this command.
- The exported file will automatically have the extension ".CRT".
- A maximum certificate chain length of 6 is supported.
- The required format for the USB memory stick is FAT32. The Operating System used in the payShield 9000 supports most types of USB memory stick, but may not have the drivers for some of the newer types. If difficulties are experienced when trying to read from or write to a USB device, an alternative memory stick should be used.

Example 1: *This example demonstrates the use of the **SE** console command to export the HSM certificate's chain of trust (in this case, just the root CA certificate) to a USB memory stick.*

```
Secure> SE <Return>

Do you wish to save to a file [Y/N]: Y <Return>
Enter filename: BankXYZRootCA <Return>
```

Bank XYZ

```

-----BEGIN CERTIFICATE-----
MIID+TCCAAGgAMIBAgIJAJyPxxP6oxAQMA0GCSqGSIb3DQEBBQUAMIGyMQswCQYD
VQQGEwJVSzEYMBYGA1UECBMPONVja2JlZ2hhbGNoXNoaXJlMRUwEwyDVQGEHwMb25n
IENyZW5kb24zDzANBgNVBAQTB1R0YXlzcEMMA0GA1UECXMDEUHMHR4wHAYDVQQD
ExVWYXlTaGl1bGQgQ2VydG1maWNhdGUxMzAxGbkqhkiG9w0BCQEWJGphbWVzLnRl
cmplc3NlbkBOaGFzZXMTZXR1Y3Y3YXR5LmNvTAEwF0xzMzAlMDkxMDU5MjJJaFw0
MzAlMDkxMDU5MjJJaMIGyMQswCQYDVQGEHwJVSzEYMBYGA1UECBMPONVja2JlZ2hh
bGNoXNoaXJlMRUwEwyDVQGEHwMb25nIENyZW5kb24zDzANBgNVBAQTB1R0YXlzcEM
MA0GA1UECXMDEUHMHR4wHAYDVQQDEXVWYXlTaGl1bGQgQ2VydG1maWNhdGUxMzAx
GbkqhkiG9w0BCQEWJGphbWVzLnRlcmplc3NlbkBOaGFzZXMTZXR1Y3Y3YXR5LmNv
bTCCASiG9wDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANtFR+dFeafMzSwgeOK
vWxjmaUOP6z5mK+qeD4w7yNP5cv1GVGqOMFTNkJL+jEBsyo39IR0T4AoalroUb6F
yi76nmv0VVqFgPWI592bRBozGp8dZU09aJQGCuOIjEvKuUtdWrrpp0ClFEntXXsx
LpfjTa45vS1+D91azkMf1Xdi70UQyfg6ciVuoch7bq0A4nmcjSlfYE/b3fpJn6zul
S+/DvRo4NwJBHkZfAiYPHZUYAv84perRG4CRbriFUFprh1kVC+P6Gal/KMKW1ze
kKJO1zqtaU973/AD4C9QZqtMurFC9m9p84u0W2SInMEKhdolVTFgVo+h3KJfHM/
yVScAwEAAaMQMA4wDAYDVROTBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAoHEN
1QyqWSTXkhtAnu+F3qg/Qs/wLYsZaY1LBUSQasjN866SzcR/jYtVt6UYabvOkE5B
9Z4KNSiCkrTcmptc0kj4K0RjUdw4QZu43c+EM4eY8tA7fSaH1nrxkPAEUWNK0
3Re+3jQe1x6gi5znLf/FZlCEP1fySh0zhzuSo2xSY/hwUwH1ZJYKBU3wzfHG1d
GB7D4xu4jUtvtkKJQDuChUdSrf+cMstN9dkrhYNNw49L9tYrD0Zz1PM3rVXD28uAL
Wt+CPotsjIixNR18vZmEVJDWJaRibCCfrTeDBs403hmAgx/Mdv5FX/NsjhZ015m
X4FkyiQv2CJb7J/vAw==
-----END CERTIFICATE-----

```

Secure>

View Installed Certificate(s)

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SV**

Function: To view the list of currently installed certificates (for use with secure host communications). Individual certificates can be displayed in full.

Authorization: The HSM can be in any state to run this command.

Inputs: • Certificate to be displayed in full.

Outputs: • The HSM's public/private key pair must be installed (using the SG console command) prior to using this command.
 • Prompts, as above
 • List of currently installed certificates.
 • Status of HSM's private key – installed or not installed
 • HSM Certificate installed – maximum of 1 certificate
 • Client Certificate(s) installed – maximum of 10 certificates
 • CA Certificate(s) installed – maximum of 10 certificates
 • Chain of trust validity – for the HSM's certificate chain
 • Contents of selected certificate.
 • A maximum certificate chain length of 6 is supported.

Notes: • See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.

Example 1: *This example demonstrates the use of the **SV** console command to view the list of currently installed certificates, and to display the contents of the HSM's certificate.*

```
Secure> SV <Return>

HSM Private Key installed: Yes

HSM Certificate installed:
  1 - Issued to: HSM-0002, Issued by: Bank XYZ
      Validity : May 21 15:05:51 2013 GMT to May 21 15:05:51 2014 GMT
      Unique ID: 2050 - AC03FAD5

Client certificate(s) installed:
  2 - Issued to: APP-0001, Issued by: Applications
      Validity : May  7 09:37:18 2013 GMT to May  7 09:37:18 2014 GMT
      Unique ID: 2016 - D221289A

CA Certificate(s) installed:
  3 - Issued to: Applications, Issued by: Applications
      Validity : May  7 09:24:10 2013 GMT to May  5 09:24:10 2023 GMT
      Unique ID: C14FF9DE78FB441A - D221289A (Root)

  4 - Issued to: Bank XYZ, Issued by: Bank XYZ
      Validity : May  9 10:59:22 2013 GMT to May  7 10:59:22 2023 GMT
      Unique ID: 9C8FC713FAA31010 - AC03FAD5 (Root)

Chain of Trust validated:
  Bank XYZ (Root)

Select an item to view: 1 <Return>

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 8273 (0x2051)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=UK, ST=Greater London, L=London, O=Bank XYZ, OU=RootCA,
    CN=Bank XYZ/emailAddress=root@bankxyz.com
    Validity
      Not Before: May 21 15:05:51 2013 GMT
      Not After : May 21 15:05:51 2014 GMT
    Subject: C=UK, ST=Greater London, O=Bank XYZ, OU=Operations, CN=HSM-
    0002/emailAddress=bill@bankxyz.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:aa:31:e6:90:46:fe:e9:26:8b:93:39:5a:8c:be:
        ...
        3d:39:2b:d7:06:47:04:6a:54:d2:12:4e:ac:9a:a3:
        5b:49
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha1WithRSAEncryption
    b8:e9:e9:8f:2e:f9:50:93:a1:8b:8d:0b:e5:fd:ef:6f:6c:05:
    ...
    59:0d:df:85:b7:48:c6:02:d9:16:f9:80:e5:c9:c2:69:7f:06:
    2b:ba:18:9f

Do you wish to view another certificate? N <Return>

Online>
```

Delete Installed Certificate(s)

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SD**

Function: To delete a currently installed certificate (for use with secure host communications).

Authorization: The HSM must be in the secure state to run this command.

Inputs: • Certificate to be deleted.

Outputs: • Prompts, as above
 • List of currently installed certificates.
 • Status of HSM's private key – installed or not installed
 • HSM Certificate installed – maximum of 1 certificate
 • Client Certificate(s) installed – maximum of 10 certificates
 • CA Certificate(s) installed – maximum of 10 certificates
 • Chain of trust validity – for the HSM's certificate chain
 • Prompt to delete another certificate

Notes: • See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.

Example 1: *This example demonstrates the use of the **SD** console command to remove a client certificate from the HSM.*

```
Secure> SD <Return>

HSM Private Key installed: Yes

HSM Certificate installed:
  1 - Issued to: HSM-0002, Issued by: Bank XYZ
    Validity : May 21 15:05:51 2013 GMT to May 21 15:05:51 2014 GMT
    Unique ID: 2050 - AC03FAD5

Client certificate(s) installed:
  2 - Issued to: APP-0001, Issued by: Applications
    Validity : May  7 09:37:18 2013 GMT to May  7 09:37:18 2014 GMT
    Unique ID: 2016 - D221289A

CA Certificate(s) installed:
  3 - Issued to: Applications, Issued by: Applications
    Validity : May  7 09:24:10 2013 GMT to May  5 09:24:10 2023 GMT
    Unique ID: C14FF9DE78FB441A - D221289A (Root)

  4 - Issued to: Bank XYZ, Issued by: Bank XYZ
    Validity : May  9 10:59:22 2013 GMT to May  7 10:59:22 2023 GMT
    Unique ID: 9C8FC713FAA31010 - AC03FAD5 (Root)

Chain of Trust validated:
  Bank XYZ (Root)

  5 - HSM Private Key

Select an item to delete (6 for ALL): 2 <Return>
Do you wish to delete another certificate? N <Return>
Secure>
```


Generate HRK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SK**

Function: To generate a new HSM Recovery Key (HRK). Once installed, the HRK will be used to back-up secret key material inside the HSM into persistent memory (a process known as key synchronization).

The following secret key material is backed-up in this process:

- Secure Host Communications key material:
 - HSM's private key
- Remote Management key material:
 - HSM's private key
 - HSM's public key certificate
 - CA public key certificate

Authorization: The HSM must be in the secure state to run this command.

Inputs: • Passphrases 1 & 2 (each entered twice for verification).

Outputs: • Prompts, as above.
• Passphrase rules.
• Creating HRK message.
• Key synchronization message.

Notes: • See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.
• The HRK replaces the RMK (used in previous versions of software).

Example 1: *This example demonstrates the use of the **SK** console command to generate an HRK.*

```
Secure> SK <Return>

**** NOTE ****
Passphrase rules as follows:
1 - Must be between 8 and 30 characters long.
2 - Can contain spaces
3 - Must be comprised of (at a minimum):
    2 digits
    2 uppercase characters
    2 lowercase characters
    2 symbols (e.g. !/?.#:')

Enter administrator 1 passphrase: *****
Re-enter administrator 1 passphrase: *****

Enter administrator 2 passphrase: *****
Re-enter administrator 2 passphrase: *****

Creating HRK. Please, wait ... DONE

HRK generated successfully

Key synchronization complete

Secure>
```

Change HRK Passphrase

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SP**

Function: To change one of the passphrases associated with the HRK.

Authorization: The HSM must be in the secure state to run this command.

Inputs:

- Existing passphrase 1 or 2.
- New passphrase 1 or 2 (entered twice for verification).

Outputs:

- Prompts, as above.
- Passphrase rules.
- Creating HRK message.
- Key synchronization message.

Notes:

- The HRK replaces the RMK (used in previous versions of software).
- See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.

Example 1: *This example demonstrates the use of the **SP** console command change administrator #1's HRK passphrase.*

```
Secure> SP <Return>

**** NOTE ****
Passphrase rules as follows:
1 - Must be between 8 and 30 characters long.
2 - Can contain spaces
3 - Must be comprised of (at a minimum):
    2 digits
    2 uppercase characters
    2 lowercase characters
    2 symbols (e.g. !/?.#:')
4 - Cannot use the same passphrase that was used within the past 10 previous
attempts

Select administrator password to change [1,2]: 1
Enter administrator 1 current passphrase: *****
Enter administrator 1 new passphrase: *****
Re-enter administrator 1 new passphrase: *****

Changing passphrases. Please, wait ... DONE

HRK generated successfully

Secure>
```

Restore HRK

Variant <input checked="" type="checkbox"/>		Keyblock <input checked="" type="checkbox"/>	
Online <input checked="" type="checkbox"/>	Offline <input checked="" type="checkbox"/>	Secure <input checked="" type="checkbox"/>	
Authorization: Not required			

Command: **SL**

Function: To restore the HRK (and also the secret key material backed-up by the HRK) in the event of erasure of tamper protected memory.

Authorization: The HSM must be in the secure state to run this command.

Inputs: • Passphrases 1 & 2.

Outputs: • Prompts, as above.
• Restoring HRK message.
• Key synchronization message.

Errors: • HRK already loaded.

Notes: • See Chapter 14 of the *payShield 9000 General Information Manual* for a description of how Secure Host Communications works on the payShield 9000.
• The HRK replaces the RMK (used in previous versions of software).

Example 1: *This example demonstrates the use of the **SL** console command to generate an HRK.*

```
Secure> SL <Return>

Enter administrator 1 passphrase: *****
Enter administrator 2 passphrase: *****

Recovering HRK. Please, wait ... DONE

HRK recovered successfully

Key synchronization complete

Secure>
```

Appendix A – Error Codes

The information from this Appendix has been moved to Appendix A of the *payShield 9000 General Information Manual*.

Appendix B – Core HSM Commands

The information from this Appendix has been moved to Appendix B of the *payShield 9000 General Information Manual*.

Appendix C – PIN Block Formats

The information from this Appendix has been moved to Chapter 6 of the *payShield 9000 General Information Manual*.

Appendix D – Key Scheme Table

The information from this Appendix has been moved to Appendix C of the *payShield 9000 General Information Manual*.

Appendix E – Variant LMKs

The information from this Appendix has been moved to Chapter 4 of the *payShield 9000 General Information Manual*.

Appendix F – Keyblock LMKs

The information from this Appendix has been moved to Chapter 5 of the *payShield 9000 General Information Manual*.

Appendix G – List of Authorizable Activities

The information from this Appendix has been moved to Appendix D of the *payShield 9000 General Information Manual*.

Appendix H – Reduced Character Sets

The information from this Appendix has been moved to Appendix E of the *payShield 9000 General Information Manual*.

Appendix I – Configure Security Settings

For a description of the security parameters referenced in the CS and QS Console commands, see the section “Configure Security” in Chapter 2 of the *payShield 9000 Security Operations Manual*.

Appendix J – Fraud Detection Functions

The information from this Appendix has been moved to Chapter 7 of the *payShield 9000 General Information Manual*.

Appendix K – Thales Keyblock / TR-31 Key Usage Conversion

The information from this Appendix has been moved to Appendix F of the *payShield 9000 General Information Manual*.

Appendix L – Utilization Data

The information from this Appendix has been moved to Chapter 8 of the *payShield 9000 General Information Manual*.

Appendix M – Health Check Data

The information from this Appendix has been moved to Chapter 9 of the *payShield 9000 General Information Manual*.

Appendix N – PCI HSM Compliance

The information from this Appendix has been moved to Chapter 10 of the *payShield 9000 General Information Manual*.

Appendix O – Error Responses Excluded from Audit Log

If the option to Audit Error Responses to Host Commands has been selected using AUDITOPTIONS, those errors which may require attention by the HSM Administrators or Security Officers are logged.

The following non-00 error responses will not be included in the Audit Log:

Cmnd	Not Audited if error response is:		
	01	02	43
A6	X		
BC	X		
BE	X		
BK		X	
BY	X		
CG	X		
CK	X	X	
CM	X		
CO	X		
CQ	X		
CU	X		
DA	X	X	
DC	X		
DE		X	
DU	X	X	
EA	X	X	
EC	X		
EE		X	
EG	X		
EI			X
F0	X		
F2	X		
FA	X		
FU	X		
G2	X		
G4	X		
GO	X		
GQ	X		
GS	X		
GU	X		
J0			X
K2	X		
KE			X
KO			X
P0	X		
PG	X		
PY	X		
QQ	X		
QS	X		
QU	X		
QW	X		
XM	X		
XK	X		
ZU	X		

Glossary

The information from this Appendix has been moved to Appendix G of the *payShield 9000 General Information Manual*.

General Abbreviations

The information from this Appendix has been moved to Appendix H of the *payShield 9000 General Information Manual*.

About Thales e-Security

Thales e-Security is a leading global provider of data encryption and cyber security solutions to the financial services, high technology manufacturing, government and technology sectors. With a 40-year track record of protecting corporate and government information, Thales solutions are used by four of the five largest energy and aerospace companies, 22 NATO countries, and they secure more than 80 percent of worldwide payment transactions. Thales e-Security has offices in Australia, France, Hong Kong, Norway, United Kingdom and United States. For more information, visit www.thales-esecurity.com

Follow us on:

