LANGUAGE REFERENCE MANUAL

for the:

Flexible Testing Tool

Prepared by:

Suite 102
7031 E. Oakland St.
Chandler, AZ 85226
(480) 429-0369

Revision History

Revision	Date	Summary of Change	Author
-	15 Feb 2004	Initial version.	Arlen Baker
A	29 Mar 2004	Added ds101.fcs.calculate function Enhanced DS101.Initialize function parameter descriptions Updated parameter descriptions for the FDU.608.SetTimeRequest.Create function. Added DS-101 Frame Layer Functions	Arlen Baker
B	16 Aug 2004	Partitioned the following into Fill and Issue CIK Split Request DTD Split Request TrKEK Requst Updated FDU.ExtendedData.Fill.Create and FDU.ExtendedData.Issue.Create to include the XDUControl parameter Added FDU.GetTimeRequest.Create Added FDU.GetTimeResponse.*.Get functions Updated the DS-100 Tag creation functions Updated the Key/Data.request functions Updated FDU.KeyDataRequest.Text.Get return value description	Arlen Baker
C	26 Aug 2004	Added DS-101 Monitoring functions Added EKMS 608 Extended-Command.request functions for both Primary and Secondary.	Arlen Baker
D	13 Sep 2004	Added DS-101 Secondary function to block until connection is received Added ftt.version.get	Arlen Baker
E	30 Sep 2004	Added ftt.timed.prompt.user Added ftt.serialnumber.get Added ftt.environmentvariable.get Added ftt.computername.get Added ftt.prompt.user.choices Added ftt.timed.prompt.user.choices Added ftt.textfile.read, ftt.textfile.write, ftt.binaryfile.read, ftt.binaryfile.write	Arlen Baker

Revision	Date	Summary of Change	Author
F	02 Nov 2004	Added ftt.username.get Added ftt.CurrrentTime.*.get Added ftt.ConvertToByte Added ftt.ConvertToWork Added ftt.ConvertToDword	Arlen Baker
G	10 Nov 2004	Added DTD-Compatible flag to ds101.initialize	Arlen Baker
H	18 Nov 2004	Added transmit and receive buffer sizes to DS101.Initialize and DS101.FrameLayer.Initialize functions Added functions to control the DTD Compatibility processing	Arlen Baker
I	2006	Added DS-102 functions and updates to DS-101 Initialize function.	Keith Hines
J	2006	DS-102 Monitor	Keith Hines
K	2007	Added Loop Statement Updated Copyright date	Chuong Vuong
L	2009	Fix errors. Add undocumented functions.	Keith Hines
M	2009	Ds101.framelayer.initialize is missing most parameters. Ds101.initialize missing parameter for delay timer of RR.	Keith Hines

Table Of Contents

1 SCOPE 3
1.1 Document Overview 3
1.2 System Overview 3
1.3 Definitions 3

2 REFERENCE DOCUMENTS 3

3 SYNTAX 3
3.1 Backus-Naur-Form 3
3.2 BNF for FSL 3

4 SEMANTICS 3
4.1 General 3
4.2 Variables 3
4.3 Literals 3
4.4 Operators 3
4.5 Functions 3
4.6 Logical Statements 3
4.7 Loop Statements 3

5 PREDEFINED VARIABLES 3
5.1 Predefined Variables Table – Common 3
5.2 Predefined Variables Table – Smart Card 3
5.3 Predefined Variables Table – DS-101 3

6 FUNCTION LIBRARY 3
6.1 Function Expressions 3
6.2 Smart Card Related Functions 3
6.2.1 Common Functions 3
6.2.2 *IAm* Terminal Functions 3
6.2.3 *Iam* "None" Functions (Low-level bypassing protocols) 3
6.3 FTT Common Functions 3
6.4 DS-101 Common Functions 3
6.4.1 *IAm* DS101Primary Functions 3
6.4.2 *IAm* DS101Secondary Functions 3
6.4.3 *IAm* DS101Monitoring Functions 3
6.5 DS-102 Functions 3

7 DIRECTIVES 3
7.1 Directive Table 3

8 EXAMPLE SCRIPTS 3
8.1 Smart Card Related Scripts 3
8.1.1 Keys.ftt 3
8.1.2 Mutual_Authentication.ftt 3
8.1.3 Commit.ftt 3
8.1.4 TestCase_01.ftt 3
8.1.5 TestCase_02.ftt 3
8.2 DS-101 Related Scripts 3
8.2.1 VerifyAXIDResponse.inc 3
8.2.2 AXID.ftt 3
8.2.3 DS-101 Monitoring 3

9 INDEX 3

Table of Figures

Figure 1.2 Functional Areas of FTT. 3

1 SCOPE

1.1 DOCUMENT OVERVIEW

This Language Reference Manual (LRM) formally defines the syntax and semantics of the Flexible Scripting Language (FSL). This manual also contains detailed explanations of the use of the FSL. This LRM also contains a description of the standard functions available. Example scripts are included in the appendix.

1.2 SYSTEM OVERVIEW

Flexible Testing Tool (FTT) is comprised of several layers of functionality. These layers provide the ability for Scripts written in FSL to access the functionality for application level testing and protocol level testing of various systems.

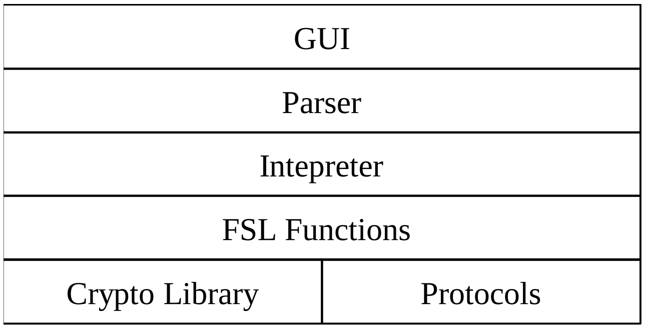

Figure 1.2 Functional Areas of FTT.

1.3 DEFINITIONS

ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
ASCII	American Standard Code for Information Interchange
AXID	Association and Exchange Identifier
BNF	Backus-Naur Form
C-APDU	Command APDU
CIK	Crypto Ignition Key
CRC	Cyclic Redundancy Check
Cyclic File	An Elementary File (EF) that contains individually identifiable records in a ring structure.
DES	Data Encryption Standard
DF	Dedicated File - a directory file.
DTD	Data Transfer Device
DU	Data Unit
EDC	Error Detection Code (CRC or LRC).
EF	Elementary File - a file with data. Can be internal (used by the card only) or working (not interpreted by the card).
EKMS	Electronic Key Management System
FCI	Function Control Information
FDU	Function Data Unit
FSL	Flexible Scripting Language
FTT	Flexible Testing Tool
ID	Identifier
	Identification
ISO	International Organization for Standardization
LRC	Longitudinal Redundancy Check
MF	Master File - the top node of the file structure, the root DF.
NVRAM	Non-volatile memory which can be written or read. Data is retained even when power is removed.
R-APDU	Response APDU
Ref	Reference
Req	Request
Res	Response

2 REFERENCE DOCUMENTS

[Ref-1]		Tier One Card/Terminal Protocols Specification	R00.00.05 01-Jun-1998
[Ref-2]	Working Draft Proposal ISO/IEC 144433 (GemPlus/Innovatron)	Identification cards - Contactless integrated circuit(s) cards - Proximity integrated circuit(s) cards Part 3: Frames and Anticollision for Type B	14.03.1998 a07
[Ref-3]	ISO 7816-3	Identification cards - Integrated circuit(s) cards with contacts	01-Jul-1997
[Ref-4]	ISO 7816-4	Identification cards - Integrated circuit(s) cards with contacts	01-Jul-1997
[Ref-5]	ANSI X3.92	Data Encryption Standard	1981
[Ref-6]	ISO 3309	Information technology - Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures - Frame structure	1993
[Ref-7]	EKMS 308	EKMS Data Tagging and Delivery Standard	25-Oct-2000
[Ref-8]	EKMS 608	ECU-Specific Command.req FDUs	27-May-2003

3 SYNTAX

3.1 BACKUS-NAUR-FORM

The syntax of the FSL is represented in standard Backus-Naur-Form (BNF). BNF is the industry standard language used to represent the syntax of a programming language. A brief description of the BNF language is

Symbol	Meaning
:==	Equivalency
[ ]	0 or 1 may appear
{ }	0 or more may appear
|	One item may appear (this is read as an OR).
..	Inclusive range
Literals	must appear

3.2 BNF FOR FSL

Item	Equivalency	
Script	:==	Statement_List
Statement_List	:==	{ Statement ; }
Statement	:==	Assignment_Statement | Logical_Statement | Directive_Statement | Function_Statement
Assignment_Statement	:==	Variable = Value_Expression
Variable	:==	Identifier | Indexed_Identifier | Sliced_Identifier
Identifier	:==	Letter_or_Underscore { Letter_or_Digit_or_Filler }
Letter_or_Underscore	:==	Letter | _
Letter	:==	A .. Z | a .. z
Letter_or_Digit_or_Filler	:==	Letter | Digit | _ | .
Digit	:==	0 .. 9
Indexed_Identifier	:==	Identifier [Value_Expression]
Value_Expression	:==	Integer_Literal | String_Literal | Variable | Function_Expression | Operation_Expression | (Value_Expression)
Integer_Literal	:==	Decimal_Literal | Hexidecimal_Literal
Decimal_Literal	:==	Digit { Digit }
Hexidecimal_Literal	:==	0x Hexidecimal_Digit { Hexidecimal_Digit }
Hexidecimal_Digit	:==	Digit | A .. F | a .. f
String_Literal	:==	“ { Printable_Letter } “
Printable_Letter	:==	Letter | Digit | ~ | ` | ! | @ | # | $ | % | ^ | & | * | (| . |) | _ | - | + | = | { | } | [|] | | | \ | : | ; | ‘ | < | > | , | ? | / | <tab> | <space>
Sliced_Identifier	:==	Identifier [Value_Expression .. Value_Expression]
Function_Expression	:==	Function_Name ([Argument_Association { , Argument_Association }])
Function_Name	:==	Identifier
Argument_Association	:==	Argument_Identifier = Value_Expression
Argument_Identifier	:==	Identifier
Logical_Statement	:==	if Logical_Expression_List then Statement_List [else Statement_List] endif
Logical_Expression_List	:==	{ Logical_Expression [Logical_Conjunction Logical_Expression] } | (Logical_Expression_List)
Logical_Expression	:==	Value_Expression [Logical_Operator Value_Expression] | (Logical_Expression) | not Logical_Expression
Logical_Conjunction	:==	|| | &&

<table>
<tr><th>Item</th><th colspan="2">Equivalency</th></tr>
<tr><td>Logical_Operator</td><td>:==</td><td>= | **!=** | > | >= | < | <=</td></tr>
<tr><td>Loop_Statement</td><td>:=</td><td>**while** Logical_Expression_List **loop** Statement_List **endloop** | **do** Statement_List **while** Logical_Expression_List</td></tr>
<tr><td>Directive_Statement</td><td>:==</td><td>Directive { Value_Expression }</td></tr>
<tr><td>Function_Statement</td><td>:==</td><td>Function_Expression</td></tr>
<tr><td>Operation_Expression</td><td>:==</td><td>Value_Expression Operator Value_Expression</td></tr>
<tr><td>Operator</td><td>:==</td><td>+ | **-** | ***** | **/** | **and** | **or** | **xor** | **&</strong></td></tr>
</table>

4 SEMANTICS

4.1 GENERAL

The language is case insensitive (upper and lowercase characters are considered equivalent). *MyData* is the same identifier as *MYDATA* as is *mydata*.

4.2 VARIABLES

Variables are dynamically created as they get used. When the FSL interpreter finds a new Identifier on the left side of an assignment statement, the interpreter dynamically allocates the variable and tags it with the known attributes such as the size, and what kind of data was placed into it. All variables are global and exist until an *exit* or *abort* statement is encountered. In accordance with the formal syntax of the FSL for identifiers the following are examples of legal identifiers in the language.

```
MyData
My_Data
_My_Data
My.Data.
thisTHAT_____The__Other.Thing
```

In addition, variables are stored as arrays of bytes and each byte can be individually addressed by using "[ ]". A slicing mechanism is also provided for accessing groups of bytes within the a variable. The zeroth index of any variable is the variable's size in bytes. The following are examples of

```
MyData[ 0 ]    // is the number of bytes stored with MyData.

// The following two variables address the same value.
MyData[ 1 .. MyData[ 0 ] ]
MyData

// The following two variables address the same value.
MyData[ 3 .. 3 ]
MyData[ 3 ]
```

4.3 LITERALS

Literal values are sometimes referred to as hard-coded values. Literals are used to represent known values. The following are valid literals which can be used anywhere a variable is used except on the left hand side of an assignment.

```
MyData = 23 ;
MyData = 0x0123456789ABCDEF ;

MyData = “hello out there!” ;
MyData = “x” ;
```

4.4 Operators

Operators can be used to calculate new values. Most programs require the ability to add, subtract, and, or, etc. Some examples are:

```
MyData = 23 + 2 ;
MyData = 0x0123456789ABCDEF – 57 ;

MyData = MyData + 1;
MyData = YourData xor 0xFFFF ;
```

4.5 Functions

The FSL contains a library of pre-defined functions which can be called directly or as the right side of an assignment statement. They can also be nested inside expressions, however this takes care to make sure the results of the function are compatible with the place they are going to go. Function names are Identifiers as described in section 4.2. Functions take arguments using named notation. In the event that no arguments are necessary, then an empty list of arguments must be used by inserting “()” after the function name. Named notation guarantees that the value passed to the function is used for the intended purpose. Because the parameters are named, they can be specified in any order. Here are some example function calls.

```
MyData = Data.Random.Get( ByteCount=8 ) ;    // this results in the variable
                                             // MyData having 8 bytes of random data.

// this results in Data_With_CRC having the value of the hard-coded
// data + a 2 byte CRC on the end.
Data = 0x00400480263000 ;
Data_With_CRC = Data & Data.CRC( Data=Data ) ;

// this results in an I-block of 0x80263000 being sent to the card.
Data.Transmit( Data=Data_With_CRC ) ;

// this function call waits for data from the card.  TimedOut is set to 1 if
// the data was not received within the specified time.
Data = Data.Receive( BlockWaitTime = 1000, CharWaitTime = 200 ) ;
```

4.6 Logical Statements

Logical functions allow expressions to be evaluated to TRUE or FALSE conditions. When the expression evaluates to TRUE, then the action statement is executed. Note that FALSE is a value of zero and TRUE is non-zero – all bits of the variable or literal are evaluated. For example, a variable with the value 0x0000000000 is FALSE. A variable with the value 0x0000010000000 is TRUE.

```
// Check a boolean to determine if the Factore should be set to 0 or 5.
If TimedOut Then Factor = 5 ; Endif ;
If not IsValid Then Factor = 0 ; Endif ;


// Check two bytes of text to determine if it should be printed.
If Message[ 4 .. 5 ] = “do”
Then print “Message is “ & Message[ 1 .. 3 ] ;
Endif ;

// Check to see if the CRC at the end of data is correct and exit if it is not.
If Data.CRC( Data=Data ) != 0x470F
Then abort ”CRC failed!” ;
Else exit ”CRC successful” ;
Endif ;
```

4.7 Loop Statements

Loop statements allow a list of statements to be executed repetitively while the logical control expression is evaluated as TRUE. The loop is terminated when the control expression is evaluated to be FALSE. Depending on the loop structure, the control expression may be evaluated before or after the loop execution.

```
// Example of a loop statement where the loop control expression is evaluated before
// the loop execution.
// This loop generates and displays 10 samples of 32-bit random data.
Sample = 0;
while ( Sample < 10 )
loop
    RandomData = Data.Random.Get( ByteCount = 4 );
    print RandomData;
    Sample = Sample + 1;
endloop;


// Example of a loop statement where the loop control expression is evaluated after
// the loop execution.
// This loop continues to generate 32-bit random data until a sample of at least
// 0x10000000 is obtained.
do
    RandomData = Data.Random.Get( ByteCount = 4 );
while ( RandomData < 0x10000000 );
print RandomData;
```

5 PREDEFINED VARIABLES

5.1 PREDEFINED VARIABLES TABLE – COMMON

Variable Name	Definition
SUCCESS	
FAIL	Unsuccessful result of executing a function
TRUE	Positive result of executing a functions
FALSE	Negative result of executing a function

5.2 PREDEFINED VARIABLES TABLE – SMART CARD

Variable Name	Meaning when simulating a card.	Meaning when simulating a terminal.
RAPDU.SW (not valid in IAm None mode).		Two byte 7816-4 Status Word value of the last received Response APDU.
CAPDU.CLA (not valid in IAm None mode).		One byte 7816-4 Class value of the last sent Command APDU.
CAPDU.INS (not valid in IAm None mode).		One byte 7816-4 Instruction value of the last sent Command APDU.
CAPDU.P1 (not valid in IAm None mode).		One byte 7816-4 P1 value of the last sent Command APDU.
CAPDU.P2 (not valid in IAm None mode).		One byte 7816-4 P2 value of the last sent Command APDU.
TimedOut (only valid in IAm None mode).		Card response to the terminal command within the wait time. 0 means no time out occurred. 1 means the card did not respond within the wait time.
CommitData (not valid in IAm None mode).		Current value of the Commit data used between a mutual authentication command and the commit command.
AFI (not valid in IAm None mode).		The AFI to use when sending an ISO 14443 REQUEST command. Defaults to 0 (all cards). The script can change this value and it will be used when the next *Card.Reset* or *Session Start* is executed.

5.3 PREDEFINED VARIABLES TABLE – DS-101

Variable Name	Definition
SecondaryDS101Address	The Address of the secondary station. This variable is set after a successful DS101.Connect occurs.
TransmissionConfirmed	Used to compare the result of DS101.FDU.Transmit to verify a successful transmission.
NoConnection	The value that DS101.Disconnect returns if no connection has been made.
ConnectConfirmed	The value of a successful DS101.Connect function call.
AlreadyConnected	The value of what DS101.Connect would return if a successful connection has already been made.
ConnectionPending	The value of what DS101.Connect would return if the connection is pending.
AcceptPending	The value of what DS101.Connect would return if the connection is pending.
LinkFailure	Returned by DS101.FDU.Transmit and DS101.FDU.Receive if the link between the stations has been broken.
NotPrimary	The value of what DS101.Disconnect returns if it is not the primary station.
DisconnectConfirm	The value of what DS101.Disconnect returns if the disconnection was successful.
TransmissionPending	The value of DS101.FDU.Transmit returns if a transmission is pending.

6 FUNCTION LIBRARY

6.1 FUNCTION EXPRESSIONS

The function library contains a set of functions which can be used as *Function_Expression*s in FSL scripts. There are four classifications of functions: Common, Terminal, Card, and None functions. The Common functions are available independent upon the *IAm* directive's value (Terminal, Card, None). The other groups of functions are specific to the *IAm* directive's value. The Terminal and Card values are obvious, the *None* value specifies to the tool that the communication protocols are being bypassed. When bypassing the protocols, the type of simulation is irrelevant, it is up to the script writer to track what is being simulated. The reason the protocol based functions are not available when in the *None* mode, is due to the ambiguities which would be caused between the state machine of the communication protocol with the direct communication.

6.2 SMART CARD RELATED FUNCTIONS

6.2.1 COMMON FUNCTIONS

The Common functions are available independent of the value of the *IAm* directive's value.

Smart Card Related Functions – Common Functions		
FUNCTION/PARAMETER	**DESCRIPTION**	**[Ref-1]**
Value_Expression =	A command APDU with the specified fields.	
APDU.Command.Create	Create a command APDU which does not have any associated data (Lc) or an expected byte count (Le).	
(		
Class = *Value_Expression*	Command APDU class field. Valid range is 0 through 0xFF.	
Instruction = *Value_Expression*	Command APDU instruction field. Valid range is 0 through 0xFF.	
P1 = *Value_Expression*	Command APDU P1 field. Valid range is 0 through 0xFF. Defaults to 0.	
P2 = *Value_Expression*	Command APDU P2 field. Valid range is 0 through 0xFF. Defaults to 0.	
)		

Smart Card Related Functions – Common Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	A command APDU with the specified fields.	
APDU.Command.Create	Create a command APDU which contains associated data (Lc) without an expected byte count (Le).	
(		
Class = *Value_Expression*	Command APDU class field. Valid range is 0 through 0xFF.	
Instruction = *Value_Expression*	Command APDU instruction field. Valid range is 0 through 0xFF.	
P1 = *Value_Expression*	Command APDU P1 field. Valid range is 0 through 0xFF. Defaults to 0.	
P2 = *Value_Expression*	Command APDU P2 field. Valid range is 0 through 0xFF. Defaults to 0.	
Data = *Value_Expression*	Command APDU data field (implies Lc). Valid number of bytes are from 0 to 254.	
)		
Value_Expression =	A command APDU with the specified fields.	
APDU.Command.Create	Create a command APDU which does not have any associated data (Lc) but does have an expected byte count (Le).	
(		
Class = *Value_Expression*	Command APDU class field. Valid range is 0 through 0xFF.	
Instruction = *Value_Expression*	Command APDU instruction field. Valid range is 0 through 0xFF.	
P1 = *Value_Expression*	Command APDU P1 field. Valid range is 0 through 0xFF. Defaults to 0.	
P2 = *Value_Expression*	Command APDU P2 field. Valid range is 0 through 0xFF. Defaults to 0.	
Le = *Value_Expression*	Command APDU Le (expected number of response bytes) field. Valid range is from 0 to 0xFF.	
)		

Smart Card Related Functions – Common Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	A command APDU with the specified fields.	
APDU.Command.Create	Create a command APDU which contains associated data (Lc) and an expected byte count (Le).	
(		
Class = *Value_Expression*	Command APDU class field. Valid range is 0 through 0xFF.	
Instruction = *Value_Expression*	Command APDU instruction field. Valid range is 0 through 0xFF.	
P1 = *Value_Expression*	Command APDU P1 field. Valid range is 0 through 0xFF. Defaults to 0.	
P2 = *Value_Expression*	Command APDU P2 field. Valid range is 0 through 0xFF. Defaults to 0.	
Data = *Value_Expression*	Command APDU data field (implies Lc). Valid number of bytes are from 0 to 254.	
Le = *Value_Expression*	Command APDU Le (expected number of response bytes) field. Valid range is from 0 to 255.	
)		
Value_Expression =	A response APDU with the specified fields.	
APDU.Response.Create	Create a response APDU.	
(		
Data = *Value_Expression*	Response APDU data. Valid number of bytes are from 0 to 254. Defaults to no data.	
StatusWord = *Value_Expression*	Response APDU status word. Valid range is from 0 through 0xFFFF.	
)		
Value_Expression =	A 1 byte value which represents the CLA field of the specified command APDU.	
APDU.Command.CLA	Determine the CLA (class) field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		

Smart Card Related Functions – Common Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	A 1 byte value which represents the INS field of the specified command APDU.	
APDU.Command.INS	Determine the INS (instruction) field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		
Value_Expression =	A 1 byte value which represents the P1 field of the specified command APDU.	
APDU.Command.P1	Determine the P1 (parameter 1) field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		
Value_Expression =	A 1 byte value which represents the P2 field of the specified command APDU.	
APDU.Command.P2	Determine the P2 (parameter 2) field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		
Value_Expression =	A 1 byte value which represents the Lc field of the specified command APDU.	
APDU.Command.Lc	Determine the Lc (length of data) field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		
Value_Expression =	Zero or more bytes which represent the Data field of the specified command APDU.	
APDU.Command.Data	Determine the Data field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		

Smart Card Related Functions – Common Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	A 1 byte value which represents the Le field of the specified command APDU	
APDU.Command.Le	Determine the Le (expected return length) field of the specified command APDU.	
(		
APDU = *Value_Expression*	Command APDU. Valid range is 4 through 254 bytes.	
)		
Value_Expression =	Zero or more bytes which represents the Data of the specified response APDU.	
APDU.Response.Data	Determine the Data field of the specified response APDU.	
(		
APDU = *Value_Expression*	Response APDU. Valid range is 2 through 254 bytes.	
)		
Value_Expression =	A 2 byte value which represents the SW of the specified response APDU.	
APDU.Response.SW	Determine the SW (status word) field of the specified response APDU.	
(		
APDU = *Value_Expression*	Response APDU. Valid range is 2 through 254 bytes.	
)		
Value_Expression =	A two byte CRC value.	
Data.CRC	Calculate a CRC for the specified data.	
(		
Data = *Value_Expression*	Data to calculate the CRC over.	
)		
Value_Expression =	A one byte LRC value.	
Data.LRC	Calculate an LRC for the specified data.	
(		
Data = *Value_Expression*	Data to calculate the LRC over.	
)		

Smart Card Related Functions – Common Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Encrypted data.	
Data.DES.ECB.Encrypt	Encrypt data using 56-bit DES in ECB mode.	
(		
Data = *Value_Expression*	Data to encrypt.	
Key = *Value_Expression*	64 bit DES key.	
)		
Value_Expression =	Decrypted data.	
Data.DES.ECB.Decrypt	Decrypt data using 56-bit DES in ECB mode.	
(		
Data = *Value_Expression*	Data to decrypt.	
Key = *Value_Expression*	64 bit DES key.	
)		
Value_Expression =	Triple encrypted data.	
Data.DES.ECB.Encrypt.Triple	Encrypt data using 56-bit triple DES in ECB mode.	
(		
Data = *Value_Expression*	Data to encrypt.	
Key = *Value_Expression*	128 bit double DES key.	
)		
Value_Expression =	Triple decrypted data.	
Data.DES.ECB.Decrypt.Triple	Decrypt data using 56-bit triple DES in ECB mode.	
(		
Data = *Value_Expression*	Data to decrypt.	
Key = *Value_Expression*	128 bit double DES key.	
)		
Value_Expression =	Encrypted data.	
Data.DES.CBC.Encrypt	Encrypt data using 56-bit DES in CBC mode.	
(		
Data = *Value_Expression*	Data to encrypt.	
Key = *Value_Expression*	64 bit DES key.	
InitialVector = *Value_Expression*	The initial vector to input to the CBC engine. Valid values are any 8 bytes. Defaults to all zeroes.	
)		

Smart Card Related Functions – Common Functions

Function/Parameter	Description	[REF-1]
Value_Expression =	Decrypted data.	
Data.DES.CBC.Decrypt	Decrypt data using 56-bit DES in CBC mode.	
(		
Data = *Value_Expression*	Data to decrypt.	
Key = *Value_Expression*	64 bit DES key.	
InitialVector = *Value_Expression*	The initial vector to input to the CBC engine. Valid values are any 8 bytes. Defaults to all zeroes.	
)		
Value_Expression =	Encrypted data.	
Data.DES.CBC.Encrypt.Triple	Encrypt data using 56-bit triple DES in CBC mode.	
(		
Data = *Value_Expression*	Data to encrypt.	
Key = *Value_Expression*	128 bit double DES key.	
InitialVector = *Value_Expression*	The initial vector to input to the CBC engine. Valid values are any 8 bytes. Defaults to all zeroes.	
)		
Value_Expression =	Decrypted data.	
Data.DES.CBC.Decrypt.Triple	Decrypt data using 56-bit triple DES in CBC mode.	
(		
Data = *Value_Expression*	Data to decrypt.	
Key = *Value_Expression*	128 bit double DES key.	
InitialVector = *Value_Expression*	The initial vector to input to the CBC engine. Valid values are any 8 bytes. Defaults to all zeroes.	
)		
Value_Expression =	A random sequence of bytes.	
Data.Random.Get	Generate random data.	
(		
ByteCount = *Value_Expression*	Number of random bytes to create.	
)		

Smart Card Related Functions – Common Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	The sequence of bytes which comprise the ISO 7816 T=1 block without the EDC.	
ISO7816.T1.Block.Create	Creates a ISO 7816 block (I, R, or S).	
(		
NAD = *Value_Expression*	Block NAD field. Valid range is 0 through 0xFF. Defaults to 0.	
PCB = *Value_Expression*	Block PCB field. Valid range is 0 through 0xFF.	
INF = *Value_Expression*	INF data. Valid number of bytes are from 0 to 254.	
)		

6.2.2 *IAm* Terminal Functions

The Terminal functions are only available when the value of the *IAm* directive is *Terminal*.

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Card.Reset	Reset the card and waits for an ATR when the protocol is T=1 or deselects the card (if selected) and arbitrates the selection of a card when the protocol is ISO 14443.	
(		
)		
Value_Expression =	Response APDU from the card.	
APDU.Transmit	Transmit an APDU to the card and wait for a response from the card.	
(		
APDU = *Value_Expression*	APDU to be sent across the communication channel. Valid number of bytes are 0 to 254.	
)		
Value_Expression =	Current card's historical bytes.	
Card.HistoricalBytes	Get the historical bytes associated with the card which is currently being communicated.	
(		
)		
Value_Expression =	Current card's PUPI.	
Card.PUPI	Get the PUPI associated with the card which is currently being communicated. This functions returns a blank PUPI if the protocol is not ISO 14443.	
(		
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	An 8 byte set of data which the terminal can use in the Authentication.External command.	
Authentication.Challenge.Get	Ask the card for challenge data which the terminal can use to validate itself to the card.	2.7.5
(		
)		
Value_Expression =	Response APDU from the card containing the status word.	
Authentication.External	Authenticate the terminal to the card using the previously obtained challenge data.	2.7.3
(		
Cryptogram = *Value_Expression*	An 8 byte set of data which is sent to the card as a means of demonstrating to the card that the terminal is valid. Valid values are any 8 bytes.	
)		
Value_Expression =	Response APDU from the card containing the 8 byte signature (MAC) followed by the 8 byte random number used to diversify the key, followed by the status word.	
Data.Sign	Have the card sign a set of data.	2.5.6
(		
Data = *Value_Expression*	Data to be signed. Valid values: any sequence of 8 or more bytes.	
KeyID = *Value_Expression*	Which key on the card to use to sign the data with. Valid values are from 0 to 253.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing eight bytes of traceability information associated with the card followed by the status word.	
Card.Traceability.Get	Get the traceability information of the card.	2.6.1
(		
)		
Value_Expression =	Response APDU from the card containing the data read from card followed by the status word.	
Data.Get	Read data on the card during initialization.	2.7.4
(		
StartAddress = *Value_Expression*	A 2 byte address which identifies the first byte to retrieve from the card. Valid values: any 2 bytes.	
ByteCount = *Value_Expression*	The number of bytes to read. Range is 1 through 254.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
Data.Put	Write data to the card during initialization.	2.7.6
(		
StartAddress = *Value_Expression*	A 2 byte address which identifies the first byte to be written within the card. Valid values: any 2 bytes.	
Data = *Value_Expression*	The bytes to write. Range is 0 through 249 bytes.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
Card.State.Transition	Transition the card from the it's current state to the next state (i.e. from initialization to personalization).	2.7.7
(		
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing the status word.	
File.Select	Selects the MF, DF, or EF with the specified FileID which is under the current DF.	2.3.3
(		
FileID = *Value_Expression*	The file identifier of the MF or child DF or child EF to select. Range 1 through 0xFFFE.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Select.MF	Selects the Master File (MF) on the card.	2.3.3
(		
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Select.DF	Selects the DF with the specified FileID which is under the current DF.	2.3.3
(		
FileID = *Value_Expression*	The file identifier of the child DF to select. Range 1 through 0xFFFE.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Select.EF	Selects the EF with the specified FileID which is under the current DF.	2.3.3
(		
FileID = *Value_Expression*	The file identifier of the child EF to select. Range 1 through 0xFFFE.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Select.Parent.DF	Selects the parent DF of the current DF.	2.3.3
(		
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing the status word.	
File.Select.Parent.EF	Selects the parent DF of the current EF.	2.3.3
(		
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Block	Request that the file be blocked from further access.	2.5.1
(		
FileID = *Value_Expression*	The file identifier of the file to block. Defaults to current file. Range 0 (current) through 30.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Unblock	Request that the blocked file be unblocked which will allow access to the file again.	2.5.1
(		
FileID = *Value_Expression*	The file identifier of the file to unblock. Defaults to current file. Range 0 (current) through 30.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing binary data returned from Card (can be less than the number requested if the number requested exceeds the amount in the file); followed by the status word.	
File.Binary.Read	Reads binary data from the transparent EF.	2.3.1
(		
FileID = *Value_Expression*	The file identifier of the file to read. Defaults to current file. Range 0 (current) through 30.	
ByteOffset = *Value_Expression*	The starting byte to read from. Defaults to 0 (first byte). Range 0 (first byte) through 0xFF when the file is not the current or 0 (first byte) through 0x7FFF when the current file is used.	
ByteCount = *Value_Expression*	The number of bytes to read. Range 0 through 252.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Binary.Update	Writes binary data to the transparent EF.	2.3.4
(		
FileID = *Value_Expression*	The file identifier of the file to update. Defaults to current file. Range 0 (current) through 30.	
ByteOffset = *Value_Expression*	The starting byte index to write. Defaults to 0 (first byte). Range 0 (first byte) through 0xFF when the file is not the current or 0 (first byte) through 0x7FFF when the current file is used.	
Data = *Value_Expression*	The data to be written to the transparent EF. Length is derived from this. Valid number of bytes is between 1 and 249.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing the record (or partial record) returned from Card followed by the status word.	
File.Record.Read	Reads a record (or partial record) from the cyclic EF.	2.3.2
(		
FileID = *Value_Expression*	The file identifier of the file to read. Defaults to current file. Range 0 (current) through 30.	
RecordSize = *Value_Expression*	The number of bytes in each record. Defaults to 0 (unspecified). Valid range is 0 (unspecified) to 0xFF. This value determines the Le field of the APDU.	
RecordNumber = *Value_Expression*	The record index of the record to read. Valid range is 0 (current) to 255. Defaults to the current (0).	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing the record(s) returned from Card followed by the status word.	
File.Record.Read	Reads records from the cyclic EF.	2.3.2
(		
FileID = *Value_Expression*	The file identifier of the file to read. Defaults to current file. Range 0 (current) through 30.	
FromRecordNumber = *Value_Expression*	The record index of the record to start reading. Valid range is 0 (current) to 255. Defaults to the current (0).	
)		
Value_Expression =	Response APDU from the card containing the record (or partial record) returned from Card followed by the status word.	
File.Record.Read.First **File.Record.Read.Last** **File.Record.Read.Next** **File.Record.Read.Previous**	Reads a record (or partial record) from the cyclic EF. Record to read is identified by the occurrence of the file identifier.	2.3.2
(		
FileID = *Value_Expression*	The file identifier of the file to read. Defaults to current file. Range 0 (current) through 30.	
RecordSize = *Value_Expression*	The number of bytes in each record. Defaults to 0 (unspecified). Valid range is 0 (unspecified) to 0xFF. This value determines the Le field of the APDU.	
RecordID = *Value_Expression*	The record identifier of the record to read. Valid range is 0 (match any) to 255.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing the status word.	
File.Record.Update	Writes a specific record to the cyclic EF.	2.3.5
(		
FileID = *Value_Expression*	The file identifier of the file to update. Defaults to current file. Range 0 (current) through 30.	
Data = *Value_Expression*	The data to be written to the record. The size is derived from the data. Valid number of bytes is between 1 and 249.	
RecordNumber = *Value_Expression*	The record index of the record to read. Valid range is 0 (current) to 255. Defaults to the current (0).	
)		
Value_Expression =	Response APDU from the card containing the status word.	
File.Record.Update.First **File.Record.Update.Last** **File.Record.Update.Next** **File.Record.Update.Previous**	Writes a record to the cyclic EF. Record written is defined by the current record and the occurrence (first, last, next, previous).	2.3.5
(		
FileID = *Value_Expression*	The file identifier of the file to update. Defaults to current file. Range 0 (current) through 30.	
Data = *Value_Expression*	The data to be written to the record. The size is derived from the data. Valid number of bytes is between 1 and 251.	
)		

Smart Card Related Functions – *IAm* Terminal Functions

Function/Parameter	Description	[REF-1]
Value_Expression =	Response APDU from the card containing the new balance (4 bytes) followed by the status word.	
File.Purse.Credit	Load money into a purse.	2.4.1
(		
FileID = *Value_Expression*	The file identifier of the Purse file to credit. Defaults to current file. Range 0 (current) through 30.	
CurrentBalance = *Value_Expression*	The current balance of thepurse. Range 0 through 2^{31}-1.	
Amount = *Value_Expression*	The amount to be credited to the purse. Range 0 through 2^{31}-1.	
SessionKey = *Value_Expression*	The key to use in preparing the MAC for the card to authenticate this command with. Must be exactly 8 bytes.	
)		
Value_Expression =	Response APDU from the card containing the new balance (4 bytes) followed by the status word.	
File.Purse.Debit	Take money from a purse.	2.4.2
(		
FileID = *Value_Expression*	The file identifier of the Purse file to debit. Defaults to current file. Range 0 (current) through 30.	
Amount = *Value_Expression*	The amount to be debited from the purse. Range 0 Through 2^{31}-1.	
)		
Value_Expression =	Response APDU from the card containing the new balance (4 bytes) followed by the status word.	
File.Purse. Debit.Reverse	Put *took* money back into a purse.	2.4.3
(		
FileID = *Value_Expression*	The file identifier of the Purse file to credit. Defaults to current file. Range 0 (current) through 30.	
Amount = *Value_Expression*	The amount to be credited to the purse. Range 0 Through 2^{31}-1.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card.	
Memory.Test.RAM	Test the RAM memory in the card.	2.7.2
(		
)		
Value_Expression =	Response APDU from the card.	
Memory.Test.Light	Test the RAM memory in the card plus do a minimum set of tests on the NV memory.	2.7.2
(		
)		
Value_Expression =	Response APDU from the card.	
Memory.Test.Medium	Test the RAM memory in the card plus do most tests on the NV memory.	2.7.2
(		
)		
Value_Expression =	Response APDU from the card.	
Memory.Test.Full	Test the RAM memory in the card plus do all tests on the NV memory.	2.7.2
(		
)		
Value_Expression =	Response APDU from the card.	
NVMemory.Check.CRC	Calculate a CRC across an area of non-volatile memory and determine if the CRC is correct.	2.7.1
(		
BeginOffset = *Value_Expression*	The beginning offset into NV memory. Valid values are any 2 bytes.	
EndOffset = *Value_Expression*	The ending offset into NV memory. Note: that the beginning offset to the ending offset must be less than or equal to 256.	
CorrectCRC = *Value_Expression*	The correct CRC value for the memory. Valid values are any 2 bytes.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card.	
NVMemory.Check.LRC	Calculate an LRC across an area of non-volatile memory and determine if the LRC is correct.	2.7.1
(		
BeginOffset = *Value_Expression*	The beginning offset into NV memory. Valid values are any 2 bytes.	
EndOffset = *Value_Expression*	The ending offset into NV memory. Note: that the beginning offset to the ending offset must be less than or equal to 256.	
CorrectLRC = *Value_Expression*	The correct LRC value for the memory. Valid values are any byte.	
)		
Value_Expression =	Response APDU from the card containing the status word.	
Key.Update.Initialize	Initialize the updating of a key on the card.	2.5.4
(		
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing the status word.	
Key.Update	Update the specified key on the card including covering the data prior to sending it the card.	2.5.7
(		
KeyID = *Value_Expression*	The identification number of the key to update. Range 0 through 0xFE.	
KeyVersion = *Value_Expression*	The version of the key. Range 0 through 255.	
Key = *Value_Expression*	The new key value. Valid values are any 8 bytes or any 16 bytes.	
AlgorithmID = *Value_Expression*	The identification number of the algorithm that the key is used. Defaults to single DES. Range 0 through 255 where 0x44 (character 'D') is single DES.	
SecurityMask = *Value_Expression*	The security level mask. Valid values are any byte.	
SessionKeyVersion = *Value_Expression*	The version of the key used to encrypt the key prior to sending it to the card. Range 0 through 255.	
SessionKey = *Value_Expression*	The session key (usually derived from CKEK) used to derive keys for the card. Size is dependent upon the algorithm (8 bytes for DES and 16 bytes for triple DES).	
SessionKeyAlgorithmID = *Value_Expression*	The identification number of the algorithm that the session key is used. Defaults to the key's algorithm ID. Range 0 through 255 where 0x44 (character 'D') is single DES.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing: 8-byte Card key diversifier || 8-byte random number || 1-byte Key Version Number || status word.	
Authentication.Mutual.Initialize	Initialize the card for mutual authentication.	2.5.3
(		
KeyID = *Value_Expression*	The identification number of the key to update. Range 0 through 0xFE.	
FileID = *Value_Expression*	The file identifier which contains the key diversifier. Defaults to the current file (0). Valid range is 0 (current file) through 31.	
DiversifierOffset = *Value_Expression*	The offset into the file which defines the actual diversifier. Defaults to 0. Valid range is 0 through 0xFFFF.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU from the card containing: De-MAC'd 8-byte card random number || status word for response APDU. In addition this command automagically updates the CommitData built-in variable.	
Authentication.Mutual	Mutual authenticate the terminal to the card and the card to the terminal.	2.5.5
(		
TerminalRandomNumber = *Value_Expression*	An 8 byte random sequence for the card to sign and send back.	
CardRandomNumber = *Value_Expression*	An 8 byte signature (MAC) of the Random data sent to the terminal by the card in the Authentication.Mutual.Initialize function.	
Key = *Value_Expression*	The key to use to cover the CardRandomNumber with in order to allow the card to validate us as a terminal. Valid values are any 8 bytes (single DES) or any 16 bytes (single or triple DES).	
AlgorithmID = *Value_Expression*	The identification number of the algorithm that the key is used. Defaults to single DES. Range 0 through 255 where 0x44 (character 'D') is single DES.	
)		

Smart Card Related Functions – *IAm* Terminal Functions		
Function/Parameter	**Description**	**[REF-1]**
Value_Expression =	Response APDU to this command containing a status word.	
Authentication.Commit	Send the commit data to complete the authentication over the commands since the original authentication.	2.5.2
(		
CommitData = *Value_Expression*	The commit data which should be used to calculate the commit MAC to send to the card. Defaults to the global variable CommitData which was updated automagically using the commit data hashing algorithm since the last authentication.mutual command was issued to the card. Valid data is 0 or more bytes of data (any data after 8 bytes is ignored).	
Key = *Value_Expression*	The key to use to calculate the MAC on the commit. Valid values are any 8 bytes (single DES) or any 16 bytes (single or triple DES).	
AlgorithmID = *Value_Expression*	The identification number of the algorithm that the key is used. Defaults to single DES. Range 0 through 255 where 0x44 (character 'D') is single DES.	
)		

6.2.3 *IAM* "NONE" FUNCTIONS (LOW-LEVEL BYPASSING PROTOCOLS)

Smart Card Related Functions – *IAm* None Functions		
FUNCTION/PARAMETER	**DESCRIPTION**	**REF 1**
Data.Transmit	Transmits a set of bytes to the card.	
(		
Data = *Value_Expression*	Data to be sent to the card. No additional data is added to this data (should be a complete block).	
)		
Value_Expression =	Data received from the card. Built-in variable TimedOut will be set to 01 if the card timed-out, otherwise TimedOut will be set to 00.	
Data.Receive	Receive a set of bytes from the card or time out while waiting. This is the entire data stream which came from the card.	
(		
BlockWaitTime = *Value_Expression*	Amount of time to wait for the first byte of the response. Range is 0 through 2^{16}-1 ETU. Zero mean to wait until at least 1 byte is received.	
CharWaitTime = *Value_Expression*	Amount of time to wait for additional bytes of the response. No default. Range is 0 through 2^{16}-1 ETU.	
)		
Card.Reset	Reset the card. Note, the next sequence received will be the ATR sequence when using ISO 7816 T=1.	
(		
)		

6.3 FTT COMMON FUNCTIONS

These functions all available to all Iam settings.

FTT Common Functions	
Function/Parameter	**Description**
Value_Expression =	Results of resetting the FTT device
FTT.Reset	Resets the FTT device. All scripts should begin with this statement following the Iam directive.
(	
)	
Value_Expression =	The hex value of the data read from the file.
FTT.BinaryFile.Read	Opens, reads, and closes the specified binary file.
(	
FileName = *Value_Expression*	The fully qualified file name of the binary file
Size = *Value_Expression*	The number of binary bytes to read from the specified file. If the size of the binary data is unknown, specify a value much larger than the data could possibly be.
)	
Value_Expression =	Results of writing the specified data to the specified binary file.
FTT.BinaryFile.Write	Opens, writes, and closes the specified binary file.
(	
FileName = *Value_Expression*	The fully qualified file name of the binary file
Value = *Value_Expression*	The hex data to be written to the file
)	
Value_Expression =	The name of the computer the FTT application is running on
FTT.ComputerName.Get	Obtains the name of the computer the FTT application is running on
(	
)	

FTT Common Functions	
Function/Parameter	**Description**
Value_Expression =	Byte value (0x00 – 0xff inclusive)
FTT.ConvertToByte	Converts the value to a byte value
(	
Value = *Value_Expression*	The value to convert
)	
Value_Expression =	Word value (0x0000 – 0xffff inclusive)
FTT.ConvertToWord	Converts the value to a word value
(	
Value = *Value_Expression*	The value to convert
)	
Value_Expression =	Dword value (0x00000000 – 0xffffffff inclusive)
FTT.ConvertToDword	Converts the value to a double-word value
(	
Value = *Value_Expression*	The value to convert
)	
Value_Expression =	Hours (0 – 23 inclusive)
FTT.CurrentTime.Hours.Get	Obtains the hours of the current local time
(	
)	
Value_Expression =	Minutes (0 – 59 inclusive)
FTT.CurrentTime.Minutes.Get	Obtains the minutes of the current local time
(	
)	
Value_Expression =	Seconds (0 – 59 inclusive)
FTT.CurrentTime.Seconds.Get	Obtains the seconds of the current local time
(	
)	
Value_Expression =	Tenths of a second (0 – 9 inclusive)
FTT.CurrentTime.Tenths.Get	Obtains the tenths-of-a-second of the current local time
(	
)	
Value_Expression =	Hundredths of a second (0 – 9

FTT Common Functions	
Function/Parameter	**Description**
	inclusive)
FTT.CurrentTime.Hundredths.Get	Obtains the hundredths-of-a-second of the current local time
(	
)	
Value_Expression =	Year (> 2004)
FTT.CurrentTime.Year.Get	Obtains the year of the current local time
(	
)	
Value_Expression =	Day of Year (0 – 366 inclusive)
FTT.CurrentTime.DayOfYear.Get	Obtains the day-of-year of the current local time
(	
)	
Value_Expression =	Results of executing the delay
FTT.Delay	Delays for the specified number of milliseconds
(	
Time = *Value_Expression*	
)	
Value_Expression =	The setting of the specified environment variable
FTT.EnvironmentVariable.Get	Obtains the setting of the specified environment variable
(	
Env = *Value_Expression*	The string name of the environment variable to obtain
)	
Value_Expression =	Results of executing the command
OS.Execute	Executes the specified OS command
(	
Command = *Value_Expression*	
)	
Value_Expression =	Results of displaying the window
FTT.Prompt.User	Presents a prompt dialog to the user. Execution of the script halts until this window is removed.
(	

FTT Common Functions	
Function/Parameter	**Description**
Message = *Value_Expression*	String message that will be shown to the user
)	
Value_Expression =	Choice of the user's selection
FTT.Prompt.User.Choices	Presents a prompt dialog to the user. Execution of the script halts until this window is removed.
(	
Message = *Value_Expression*	String message that will be shown to the user
Choices = *Value_Expression*	String choices that will be shown to the user and that are delimited by the Delimiter parameter below
Default = *Value_Expression*	Defines the choice that is automatically selected from the list of choices in the pull-down list **GUI**: This choice will be the default shown in the pull-down list. **Console**: This choice will have a * next to it in the list.
Delimiter = *Value_Expression*	Defines the string delimiter that separates the list of choices in the Choices parameter above
)	
Value_Expression =	The hex value of the data read from the file.
FTT.TextFile.Read	Opens, reads, and closes the specified text file.
(	
FileName = *Value_Expression*	The fully qualified file name of the text file
Size = *Value_Expression*	The number of character bytes to read from the specified file. If the size of the text is unknown, specify a value much larger than the text could possibly be.
)	
Value_Expression =	Results of writing the specified data to the specified text file.
FTT.TextFile.Write	Opens, writes, and closes the specified binary file.
(	
FileName = *Value_Expression*	The fully qualified file name of the text file

FTT Common Functions	
Function/Parameter	**Description**
Value = *Value_Expression*	The string data to be written to the file
)	
Value_Expression =	Results of displaying the window
FTT.Timed.Prompt.User	Presents a prompt dialog to the user. Execution of the script halts until this window is removed or until the time period has expired.
(	
Message = *Value_Expression*	String message that will be shown to the user
Time = *Value_Expression*	The number of milliseconds to wait for the user to click on the OK button
)	
Value_Expression =	Choice of the user's selection
FTT.Timed.Prompt.User.Choices	Presents a prompt dialog to the user. Execution of the script halts until this window is removed.
(	
Message = *Value_Expression*	String message that will be shown to the user
Choices = *Value_Expression*	String choices that will be shown to the user and that are delimited by the Delimiter parameter below
Default = *Value_Expression*	Defines the choice that is automatically selected from the list of choices in the pull-down list. **GUI**: This choice will be the default shown in the pull-down list. **Console**: This choice will have a * next to it in the list.
Delimiter = *Value_Expression*	Defines the string delimiter that separates the list of choices in the Choices parameter above
Timeout = *Value_Expression*	Amount of time (in milliseconds) to wait for the user to select a choice
)	
Value_Expression =	Hex value of the passed in string
FTT.StringToHex	Converts the string to hex values
(	
String = *Value_Expression*	String to convert
)	
Value_Expression =	String value of the passed in hex value

FTT Common Functions	
Function/Parameter	**Description**
FTT.HexToString	Converts the hex value to a string
(	
Hex = *Value_Expression*	Hex value to convert
)	
Value_Expression =	The serial number of the FTT hardware device
FTT.SerialNumber.Get	Obtains serial number of the FTT device
(	
)	
Value_Expression =	The name of the user the FTT application is running under
FTT.UserName.Get	Obtains the name of the user the FTT application is running under
(	
)	
Value_Expression =	The hardware and firmward version of the FTT hardware device
FTT.Version.Get	Obtains the hardware and firmward versions of the FTT device
(	
)	
Value_Expression =	Results of zeroizing FTT memory
FTT.Zeroize	Zeroizes the memory of the FTT
(	
)	

6.4 DS-101 Common Functions

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	Results of initializing the FTT for DS-101 Frame Layer processing.	
DS101.FrameLayer.Initialize	Initializes the FTT device for DS-101 processing. Must be executed prior to any other DS-101 related functions. This is mutually exclusive from the DS-101 Link Layer initialization.	
(		
Port = *Value_Expression*	The port on the FTT device that will be used for DS-101 communication. Acceptable values are “A” or “B”.	
Address = *Value_Expression*	The address of the frame layer for connection requests	
RetryCount = *Value_Expression*	The address of the frame layer retry limit for transmitting frames	
TransmitBufferSize= *Value_Expression*	The size of the transmit buffer the FTT uses. Should be sized to twice that of the largest FDU transmitted. Default value is 0x800. Range of acceptable values is: 0x200 to 0x1000 inclusive.	
ReceiveBufferSize= *Value_Expression*	The size of the receive buffer the FTT uses. Should be sized to twice that of the largest FDU received. Default value is 0x200 Range of acceptable values is: 0x200 to 0x1000 inclusive.	
BitRate= *Value_Expression*	The bits per second to run the protocol. Default value is 64000 bits per second	
Parity= *Value_Expression*	For port C, the type of parity to use. Default value is no parity. Values can be “odd”, “even”, or “none”.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
FlowControl= *Value_Expression*	For port C, whether FCO and FCI hardware handshaking signals are used. Default value is FALSE.	
)		
Value_Expression =	Result of the transmission request	
DS101.FrameLayer.Transmit	Requests that the specified FDU is transmitted to the connected station	2.2
(		
Data = *Value_Expression*	FDU created by one of the following FDU functions below.	
)		
Value_Expression =	Result of the transmission request	
DS101.FrameLayer.Receive	Retrieves the received frame. The frame must be received within the specified timeout	2.2
(		
Timeout = *Value_Expression*	Amount of time, in milliseconds, that the Frame is to be received. Time is measured from the last command requested (i.e., Transmit or Receive). Infinitie timeout is 0xffffffff.	
)		
Value_Expression =	"TRUE" if a frame has been received; "FALSE" otherwise	
DS101.FrameLayer.Message.Available	Determines if a frame has been received	2.2
(		
)		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	Result of initializing the FTT for DS-101 Link Layer processing	
DS101.Initialize	Initializes the FTT device for DS-101 Link Layer processing. Must be executed prior to any other DS-101 Link Layerrelated functions. This is mutually exclusive with the DS-101 Frame Layer initialization.	N/A
(		
Port = *Value_Expression*	The port on the FTT device that will be used for DS-101 communication. Acceptable values are “A” or “B” or “C”. where C is the RS-232 port on the HD DB44 connector.	
Address = *Value_Expression*	The address of the link layer for connection requests	
RetryCount = *Value_Expression*	The address of the link layer retry limit for transmitting frames	
MaxTxFrames = *Value_Expression*	Specifies the maximum number of frames to be sent in a single transmission. Currently, not implemented	
AcceptConnections = *Value_Expression*	Enables/Disables the link layer from accepting connections. Acceptable values are “TRUE” (primary station) or “FALSE (secondary station).	
Monitor = *Value_Expression*	Enables/Disables for frame layer monitoring. Values are “TRUE” or “FALSE”. Frame Layer monitoring is currently not implemented.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
DTDCompatible= *Value_Expression*	Enables/Disables the DTD-compatible mode. This allows the FTT to communicate to a DTD as a secondary station by transmitting RNRs until a required response is transmitted. If no required response is required, the FTT will transmit an RR and the DTD will issue a DISCONNECT.request. To configure the FTT for the FDUs that need responses, use the command ftt.DTDCompatible.FDUNeedsResponse.Add Only valid for Secondary Stations. Values are "TRUE" or "FALSE". "FALSE" is the default value and does not need to be provided in the function call unless any following parameters are listed	
TransmitBufferSize= *Value_Expression*	The size of the transmit buffer the FTT uses. Should be sized to twice that of the largest FDU transmitted. Default value is 0x800. Range of acceptable values is: 0x200 to 0x1000 inclusive.	
ReceiveBufferSize= *Value_Expression*	The size of the receive buffer the FTT uses. Should be sized to twice that of the largest FDU received. Default value is 0x200 Range of acceptable values is: 0x200 to 0x1000 inclusive.	
BitRate= *Value_Expression*	The bits per second to run the protocol. Default value is 64000 bits per second	
Parity= *Value_Expression*	For port C, the type of parity to use. Default value is no parity. Values can be "odd", "even", or "none".	
FlowControl= *Value_Expression*	For port C, whether FCO and FCI hardware handshaking signals are used. Default value is FALSE.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
T1Connect= *Value_Expression*	The number of milliseconds to use for T1 when attempting to connect to a secondary. Default is 10 milliseconds per EKMS308	
T1= *Value_Expression*	The number of milliseconds to use for T1 when running as a primary. Default is 70 milliseconds per EKMS308	
T2= *Value_Expression*	The number of milliseconds to use for T2 when running as a secondary. Default is 2 seconds per EKMS308	
RRDelay= *Value_Expression*	The minimum number of milliseconds Primary waits before sending RR frame after receiving from secondary. This can be used to give the secondary more time to process data. Default is 0 milliseconds.	
Wakeup = *Value_Expression*	Enable the wakeup signal on Pin C or Port A or B. Wakeup signal is enabled and place in the non wakeup state. As a primary It must be enabled using DS101.Wakeup.Enable to cause the wakeup signal to wake the secondary. DS101.Wakeup.Disable will cause the wakeup signal to not wake the secondary. As a secondary a call to DS101.Wakeup.WaitFor will wait until the primary asserts the wakeup signal. Default is FALSE.	
)		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	Empty string.	
DS101.Wakeup.Enable	When the port was initialized with the wakeup signal, this call causes the wakeup signal to be asserted.	N/A
(		
Port = *Value_Expression*	The port on the FTT device that will be used for DS-101 communication. Acceptable values are “A” or “B”.	
)		
Value_Expression =	Empty string.	
DS101.Wakeup.Disable	When the port was initialized with the wakeup signal, this call causes the wakeup signal to be deasserted.	N/A
(		
Port = *Value_Expression*	The port on the FTT device that will be used for DS-101 communication. Acceptable values are “A” or “B”.	
)		
Value_Expression =	DS-101 Result of the transmission request	
DS101.FDU.Transmit	Requests that the specified FDU is transmitted to the connected station	2.1.1.4
(		
FDU = *Value_Expression*	FDU created by one of the following FDU functions below.	
)		
Value_Expression =	FDU received	
DS101.FDU.Receive	Retrieves the received FDU. The FDU must be received within the specified timeout	2.1.1.6
(		
Timeout = *Value_Expression*	Amount of time, in milliseconds, that the FDU is to be received. Time is measured from the last command requested (i.e., Transmit or Receive)	
)		
Value_Expression =	DS-101 Result of the Disconnect.request	
DS101.Disconnect	Requests that a disconnect be performed on the current connection	2.1.1.7
(		
)		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The FCS over the data	
DS101.FCS.Calculate	Calculate the EKMS 308 FCS over the data	2.2.1.6
(		
Initial Value = *Value_Expression*	The starting value of the FCS (0xFFFF for new calculations)	
Data = *Value_Expression*	The data the FCS is to be calculated over	
)		
Value_Expression =	FCI obtained from the FDU	
FDU.FCI.Get	Obtains the FCI from the passed in FDU	3.1
(		
FDU = *Value_Expression*	The FDU to retrieve the FCI from	
)		
Value_Expression =	The DU Length obtained from the FDU	
FDU.DULength.Get	Obtains the DU Length field from the passed in FDU	3.1
(		
FDU = *Value_Expression*	The FDU to retrieve the DU length from	
)		
Value_Expression =	EOT FDU	
FDU.EndOfTransmissionRequest.Create	Creates an End of Transmission.req FDU	3.3.1
(		
)		
Value_Expression =	Success Acknowledge w/o Data FDU	
FDU.SuccessAcknowledgeWithoutData.Create	Creates a Success Acknowledge without Data FDU	3.3.2
(		
)		
Value_Expression =	Failure Acknowledge w/o Data FDU	
FDU.FailureAcknowledgeWithoutData.Create	Creates a Failure Acknowledge without Data FDU	3.3.3
(		
)		
Value_Expression =	Failure Acknowledge w/o Data FDU	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
FDU.FailureAcknowledgeWithData.Create	Creates a Failure Acknowledge with Data FDU	3.3.4
(		
Data = *Value_Expression*	Error Code/String	
)		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The Data field from the Failure Acknowledge with Data FDU	
FDU.FailureAcknowledgeWithData.Data.Get	Verifies the FDU as a Failure Acknowledge with Data FDU. Obtains the Data field from the FDU	3.3.4
(		
FDU = *Value_Expression*	The received Failure Acknowledge w/o Data FDU	
)		
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
)	The Parity field will be calculated as part of the creation of the Tag.	
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Parity = *Value_Expression*	The parity field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 8 bits.	
)	Since the parity field is included in the list of parameters, it will NOT be calculated	
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Data = *Value_Expression*	The data field of the tag	
)	The parity field will be calculated	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Parity = *Value_Expression*	The parity field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 8 bits.	
Data = *Value_Expression*	The data field of the tag	
)	Since the parity field is included in the list of parameters, it will NOT be calculated	
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Text = *Value_Expression*	The text field of the tag	
)	The parity field will be calculated	
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Parity = *Value_Expression*	The parity field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 8 bits.	
Text = *Value_Expression*	The text field of the tag	
)	Since the parity field is included in the list of parameters, it will NOT be calculated	
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Data = *Value_Expression*	The data field of the tag	
Text = *Value_Expression*	The text field of the tag	
)	The parity field will be calculated	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The DS-100 Tag	
DS100.Tag.Create	Creates a DS-100 Tag	4.2.1
(		
ShortTitle = *Value_Expression*	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	
FormatType = *Value_Expression*	The Format Type field of the tag. Sized to 1 bit.	
TextIndicator = *Value_Expression*	The Text Indicator field of the tag. Sized to two bits.	
Edition = *Value_Expression*	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
RegisterNumber = *Value_Expression*	The Register Number field of the tag. Sized to 36 bits. Binary number.	
SegmentNumber = *Value_Expression*	The Segment Number field of the tag. Sized to 8 bits. Binary number.	
Use = *Value_Expression*	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 4 bits.	
Classification = *Value_Expression*	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Sized to 3 bits.	
FutureExpansion = *Value_Expression*	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits.	
Parity = *Value_Expression*	The parity field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 8 bits.	
Data = *Value_Expression*	The data field of the tag	
Text = *Value_Expression*	The text field of the tag	
)	Since the parity field is included in the list of parameters, it will NOT be calculated	
Value_Expression =	The Short Title field of the tag. Sized to 144 bits (i.e., truncated or padded with NULL characters). This function will convert the Value_Expression to the values defined in section 4.3 of [REF-7]. If the character cannot be converted, a NULL will be inserted.	

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
FDU.DS100Tag.ShortTitle.Get	Decodes the Short Title field from the specified DS100 Tag.	4.2.1.1
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Format Type field of the tag. Least significant bit of 1 byte.	
FDU.DS100Tag.FormatType.Get	Decodes the Format Type field from the specified DS100 Tag.	4.2.1.2
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Format Type field of the tag. Returns string with value “100-1 Tag or Format”, “100-2 Tag Format”.	
FDU.DS100Tag.FormatType.Get AsText	Decodes the Format Type field from the specified DS100 Tag.	4.2.1.2
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Text Indicator field of the tag. Least significant 2 bits of 1 byte.	
FDU.DS100Tag.TextIndicator.Get	Decodes the Short Title field from the specified DS100 Tag.	4.2.1.3
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Text Indicator field of the tag. Returned as “TRUE”, “FALSE”, or “<INVALID>”.	
FDU.DS100Tag.TextIndicator.Get AsText	Decodes the Short Title field from the specified DS100 Tag.	4.2.1.3
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Text Indicator field of the tag. A string of value “TRUE” if TextIndicator is binary 11, otherwise “FALSE”.	
FDU.DS100Tag.TextIndicator.HasText	Decodes the Short Title field from the specified DS100 Tag.	4.2.1.3
(		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Edition field of the tag. Sized to 6 alpha-numeric characters.	
FDU.DS100Tag.Edition.Get	Decodes Edition field from the specified DS100 Tag.	4.2.1.4
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Register Number field of the tag. Sized to 36 bits. Returned as 5 bytes representing binary register number.	
FDU.DS100Tag.RegisterNumber.Get	Decodes the Register Number field from the specified DS100 Tag.	4.2.1.5
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Segment Number field of the tag. Sized to 1 byte.	
FDU.DS100Tag.SegmentNumber.Get	Decodes the Segment Number field from the specified DS100 Tag.	4.2.1.6
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Use field of the tag. Value is not verified to match those listed in [REF-7]. Least significant 4 bits of 1 byte.	
FDU.DS100Tag.Use.Get	Decodes the Use field from the specified DS100 Tag.	4.2.1.7
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The Use field of the tag. Returns string with value “<Not Applicable>”, “FIREFLY/CROSSTALK Key”, “KEK”, “KPK”, “MSK”, “QKEK”, “TEK”, “TSK”, “TrKEK”, or “<RESERVED FOR FUTURE USE>”.	
FDU.DS100Tag.Use.Get AsText	Decodes the Use field from the specified DS100 Tag.	4.2.1.7
(		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The classification field of the tag. Value is not verified to match those listed in [REF-7]. Least significant 3 bits of 1 byte.	
FDU.DS100Tag.Classification.Get	Decodes the Classification field from the specified DS100 Tag.	4.2.1.8
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The classification field of the tag. Returns string with value “<Not Applicable>”, “Unclassified”, “Confidential”, “Secret”, “Top Secret” or “<RESERVED FOR FUTURE USE>”.	
FDU.DS100Tag.Classification.Get AsText	Decodes the Classification field from the specified DS100 Tag.	4.2.1.8
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The future expansion field of the tag. Value is not verified to match thost listed in [REF-7]. Sized to 9 bits returned as 2 bytes.	
FDU.DS100Tag.FutureExpansion.Get	Decodes the Future Expansion field from the specified DS100 Tag.	4.2.1.9
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	The parity field of the tag. Value is not verified. Sized to 1 byte.	
FDU.DS100Tag.Parity.Get	Decodes the Parity field from the specified DS100 Tag.	4.2.1.10
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	Data field of DS100 Tag.	
FDU.DS100Tag.Data.Get	Decodes the Data field from the specified DS100 Tag.	4.2.1.11
(		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	Text field returned as 16 alphanumeric characters.	
FDU.DS100Tag.Text.Get	Decodes the Text field from the specified DS100 Tag.	4.2.1.12
(		
DS100Tag = Value_Expression	The DS100 Tag	
)		
Value_Expression =	TRUE or FALSE	
ftt.DTDCompatible.FDUNeedsResponse.Clear	Specific to DTD Compatiblity mode. Clears the FCI's (EKMS 308) and the Command Codes (EKMS 608) that the FTT uses to determine if RNR or RR is to be transmitted. For example, the AXID.request requires a response (AXID.response).	
(		
)		
Value_Expression =	TRUE or FALSE	
ftt.DTDCompatible.FDUNeedsResponse.Add	Specific to DTD Compatiblity mode. Adds the FCI of the FDU that requires a response. Upon receipt of this FCI, the FTT will transmit RNRs until the script file transmits the response in an I-Frame. This will cause the DTD to disconnect.	
(		
FCI = *Value_Expression*	The FCI of the FDU that requires a response.	
)		

DS-101 Common Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	TRUE or FALSE	
ftt.DTDCompatible.FDUNeedsResponse.Add	Specific to DTD Compatiblity mode. Adds the FCI of the FDU that requires a response. Upon receipt of this Command Code, the FTT will transmit RNRs until the script file transmits the response in an I-Frame. This will cause the DTD to disconnect.	
(		
CommandCode = *Value_Expression*	The Command Code of the EKMS 608 FDU	
)		

The following table presents the FDUs from [REF-7] and [REF-8] and shows which are transmitted/received from the Primary station and which are transmitted/received from the Secondary station.

EKMS	FDU	Primary	Secondary
308 [REF-7]	End of Transmission.request	Tx/Rx	Tx/Rx
	Success Acknowledge w/o Data	Tx/Rx	Tx/Rx
	Failure Acknowledge w/o Data	Tx/Rx	Tx/Rx
	Failure Acknowledge with Data	Tx/Rx	Tx/Rx
	Set Address.request	Tx	Rx
	Set Identifier.request	Tx	Rx
	AXID.req	Tx	Rx
	AXID.res	Rx	Tx
	Field Specifier.request	Tx	Rx
	File Header.request	Tx	Rx
	End of File.request	Tx	Rx
	Key/Data.request	Tx	Rx
	Program/File Data.request	Tx	Rx
	CIK Split.request	Tx	Rx
	DTD Split.Request	Tx	Rx
	TrKEK.request	Tx	Rx
	Command.request	Tx/Rx	Tx/Rx
	Extended Data.request (Fill)	Tx	Rx
	Extended Data.request (Issue)	Tx	Rx
608 [REF-8]	Get Time.request	Tx	Rx
	Get Time.response	Rx	Tx
	Set Time.Request	Tx	Rx
	Extended Data.request (Fill)	Tx	Rx
	Extended Data.request (Issue)	Tx	Rx
	ECU Status Check	Tx	Rx
	Extended Command.request	Tx	Rx

6.4.1 *IAm* DS101Primary Functions

The DS-101 Primary functions are only available when the value of the *IAm* directive is *DS101Primary*.

Iam **DS101Primary Functions**		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	DS-101 Result of the Connect.request	
DS101.Connect	Requests that a connection be made to a secondary station	2.1.1.1
(		
Address = *Value_Expression*	Specifies the address of the secondary address to connect to. 0xff is the broadcast address	
)		
Value_Expression =	Set Address.request FDU	
FDU.SetAddressRequest.Create	Creates a Set Address.request with Data FDU	3.3.5
(		
DS101Address = *Value_Expression*	The Ds-101 Address to assign to the station. Sized to 1 byte.	
)		
Value_Expression =	Set Identifier.request FDU	
FDU.SetIdentifierRequest.Create	Creates a Set Identifier.request with Data FDU	3.3.6
(		
StationID = *Value_Expression*	The Station ID to assigned to a COMSEC equipment. Sized to 14 bytes; left justified; padded with ASCII Null characters if necessary.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	AXID.req FDU	
FDU.AXIDrequest.Create	Creates an AXID.req FDU	3.3.7
(		
Framecount = *Value_Expression*	The number of received frames that the station can accept before an acknowledgement must be sent. There is no check for the values of 1 or 7 per the EKMS spec.	
FixedID = *Value_Expression*	The FixedID of the sender (the FTT)	
StationID = *Value_Expression*	The StationID of the sender (the FTT)	
)		
Value_Expression =	The Frame Count from field from the AXID.res FDU	
FDU.AXIDresponse.Framecount.Get	Verifies the FDU is an AXID.res FDU. Obtains the Framecount field from the AXID.res FDU	3.3.8
(		
FDU = *Value_Expression*	The received AXID.res FDU	
)		
Value_Expression =	The Frame Count from field from the AXID.res FDU	
FDU.AXIDresponse.FixedID.Get	Verifies the FDU is an AXID.res FDU. Obtains the FixedID field from the AXID.res FDU	3.3.8
(		
FDU = *Value_Expression*	The received AXID.res FDU	
)		
Value_Expression =	The Station ID from field from the AXID.res FDU	
FDU.AXIDresponse.StationID.Get	Verifies the FDU is an AXID.res FDU. Obtains the StationID field from the AXID.res FDU	3.3.8
(		
FDU = *Value_Expression*	The received AXID.res FDU	
)		

***Iam* DS101Primary Functions**

Function/Parameter	Description	REF-7
Value_Expression =	The Field Specifier.request FDU	
FDU.FieldSpecifierRequest.Create	Creates the Field Specifier.request FDU	3.3.9
(		
StartByte = *Value_Expression*	The Start Byte field. Sized to 1 byte.	
Length = *Value_Expression*	The Length field. Sized to 2 bytes.	
)		
Value_Expression =	The File Header.request FDU	
FDU.FileHeaderRequest.Fill.Create	Creates the File Header.request FDU (Fill variant)	3.3.10
(		
Path = *Value_Expression*	The path of the file	
Name = *Value_Expression*	The file name	
FileType = *Value_Expression*	The type of file. The value is not verified against those listed in EKMS 308. Sized to 1 byte	
TrKEKCount = *Value_Expression*	The number of KEKs	
TrKEK1 = *Value_Expression*	The first KEK	
TrKEK2 = *Value_Expression*	The second KEK. Default value of 0.	
TrKEK3 = *Value_Expression*	The third KEK. Default value of 0.	
TrKEK4 = *Value_Expression*	The fourth KEK. Default value of 0.	
TrKEK5 = *Value_Expression*	The fifth KEK. Default value of 0.	
)		
Value_Expression =	The File Header.request FDU	
FDU.FileHeaderRequest.Issue.Create	Creates the File Header.request FDU (Issue variant)	3.3.10
(		
Path = *Value_Expression*	The path of the file	
Name = *Value_Expression*	The file name	
FileType = *Value_Expression*	The type of file. The value is not verified against those listed in EKMS 308. Sized to 1 byte	
TrKEKCount = *Value_Expression*	The number of KEKs	
TrKEK1 = *Value_Expression*	The first KEK	
TrKEK2 = *Value_Expression*	The second KEK. Default value of 0.	
TrKEK3 = *Value_Expression*	The third KEK. Default value of 0.	
TrKEK4 = *Value_Expression*	The fourth KEK. Default value of 0.	
TrKEK5 = *Value_Expression*	The fifth KEK. Default value of 0.	
)		

***Iam* DS101Primary Functions**

Function/Parameter	**Description**	**REF-7**
Value_Expression =	The End of File.request FDU	
FDU.EndOfFileRequest.Fill.Create	Creates the End of File.request FDU	3.3.11
(		
)		
Value_Expression =	The End of File.request FDU	
FDU.EndOfFileRequest.Issue.Create	Creates the End of File.request FDU	3.3.11
(		
)		
Value_Expression =	The Key/Data.request FDU	
FDU.KeyDataRequest.Create	Creates the Key/Data.request FDU	3.3.12
(		
KeyData = *Value_Expression*	The Key/Data within the FDU.	
)		
Value_Expression =	The Key/Data.request FDU	
FDU.KeyDataRequest.Create	Creates the Key/Data.request FDU	3.3.12
(		
DS100Tag = *Value_Expression*	The DS-100 Tag.	4.2.1
)		
Value_Expression =	The Program/File Data.request (Fill) FDU	
FDU.ProgramFileDataRequest.Fill.Create	Creates the Program/File Data.request (Fill) FDU	3.3.13
(		
ProgramDataFile = *Value_Expression*	The program or program data file. Sized to no larger than 252 bytes.	
)		
Value_Expression =	The Program/File Data.request FDU	
FDU.ProgramFileDataRequest.Issue.Create	Creates the Program/File Data.request (Issue) FDU	3.3.13
(		
ProgramDataFile = *Value_Expression*	The program or program data file. Sized to no larger than 252 bytes.	
)		
Value_Expression =	The CIK Split.request (Fill) FDU	
FDU.CIKSplitRequest.Fill.Create	Creates the CIK Split.request (Fill) FDU	3.3.14
(		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**REF-7**
CIKSplitData = *Value_Expression*	The CIK data. Sized to 128 bytes.	
)		
Value_Expression =	The CIK Split.request (Issue) FDU	
FDU.CIKSplitRequest.Issue.Create	Creates the CIK Split.request (Issue) FDU	3.3.14
(		
CIKSplitData = *Value_Expression*	The CIK data. Sized to 128 bytes.	
)		
Value_Expression =	The DTD Split.request (Fill) FDU	
FDU.DTDSplitRequest.Fill.Create	Creates the DTD Split.request (Fill) FDU	3.3.15
(		
DTDSplitData = *Value_Expression*	The DTD Split Data. Sized to 64 bytes.	
)		
Value_Expression =	The DTD Split.request (Issue) FDU	
FDU.DTDSplitRequest.Issue.Create	Creates the DTD Split.request (Issue) FDU	3.3.15
(		
DTDSplitData = *Value_Expression*	The DTD Split Data. Sized to 64 bytes.	
)		
Value_Expression =	The TrKEK.request (Fill) FDU	
FDU.TrKEKRequest.Fill.Create	Creates the TrKEK.request (Fill) FDU	3.3.16
(		
TrKEKData = *Value_Expression*	The TRKEK data.	
)		
Value_Expression =	The TrKEK.request (Issue) FDU	
FDU.TrKEKRequest.Issue.Create	Creates the TrKEK.request (Issue) FDU	3.3.16
(		
TrKEKData = *Value_Expression*	The TRKEK data.	
)		
Value_Expression =	The Command.request FDU	
FDU.CommandRequest.Create	Creates the Command.request FDU	3.3.17
(		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**REF-7**
CommandString = *Value_Expression*	The command string.	
)		

***Iam* DS101Primary Functions**

Function/Parameter	**Description**	**REF-7**
Value_Expression =	The Command String from the Command.request FDU	
FDU.CommandRequest.CommandString.Get	Verifies the FDU is a Command.request FDU. Obtains the Command String field.	3.3.17
(		
FDU = *Value_Expression*	The received Command.request FDU.	
)		
Value_Expression =	The Fill Variant of the Extended Data.request FDU	
FDU.ExtendedData.Fill.Create	Creates the Extended Data.request FDU (Fill variant)	3.3.18
(		
XDUControl = *Value_Expression*	The XDU Control bits	
DUData = *Value_Expression*	The DU Data of the FDU	
)		
Value_Expression =	The Fill Variant of the Extended Data.request FDU	
FDU.ExtendedData.Issue.Create	Creates the Extended Data.request FDU (Issue variant)	3.3.18
(		
XDUControl = *Value_Expression*	The XDU Control bits	
DUData = *Value_Expression*	The DU Data of the FDU	
)		
Value_Expression =	The Get Time.request FDU	
FDU.GetTimeRequest.Create	Creates the Get Time.request FDU	3.3.19
(		
)		
Value_Expression =	The Year field of the Get Time.response FDU	
FDU.GetTimeResponse.Year.Get	Verifies the FDU is a Get Time.response FDU Obtains the Year field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		
Value_Expression =	The Day-of-Year field of the Get Time.response FDU	

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**REF-7**
FDU.GetTimeResponse.Day_Of_Year.Get	Verifies the FDU is a Get Time.response FDU Obtains the Day-of-Year field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		
Value_Expression =	The Hour field of the Get Time.response FDU	
FDU.GetTimeResponse.Hour.Get	Verifies the FDU is a Get Time.response FDU Obtains the Hour field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		
Value_Expression =	The Minute field of the Get Time.response FDU	
FDU.GetTimeResponse.Minute.Get	Verifies the FDU is a Get Time.response FDU Obtains the Minute field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		
Value_Expression =	The Second field of the Get Time.response FDU	
FDU.GetTimeResponse.Second.Get	Verifies the FDU is a Get Time.response FDU Obtains the Second field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The Tenths field of the Get Time.response FDU	
FDU.GetTimeResponse.Tenths.Get	Verifies the FDU is a Get Time.response FDU Obtains the Tenths field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		
Value_Expression =	The Hundredths field of the Get Time.response FDU	
FDU.GetTimeResponse.Hundredths.Get	Verifies the FDU is a Get Time.response FDU Obtains the Hundredths field from the Get Time.response FDU	3.3.20
(		
FDU = *Value_Expression*	The received Get Time.response FDU.	
)		
Value_Expression =	The Set Time.request FDU	
FDU.SetTimeRequest.Create	Creates the Set Time.request FDU	3.3.21
(		
Year = *Value_Expression*	The Year	
Day_of_Year = *Value_Expression*	The Day of Year	
Hour = *Value_Expression*	The Hour	
Minute = *Value_Expression*	The Minute	
Second = *Value_Expression*	The Second	
Tenths = *Value_Expression*	The Tenths	
Hundredths = *Value_Expression*	The Hundredths	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Get-Time.request FDU	
FDU.608.GetTimeRequest.Create	Creates the Get-Time.request FDU	2.2
(		
)		
Value_Expression =	The Year field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Year.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Year field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		
Value_Expression =	The Day field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Day_Of_Year.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Day field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		
Value_Expression =	The Hour field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Hour.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Hour field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		
Value_Expression =	The Minute field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Minute.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Minute field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Second field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Second.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Second field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		
Value_Expression =	The Tenths field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Tenths.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Tenths field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		
Value_Expression =	The Hundredths field from the Get-Time.response FDU	
FDU.608.GetTimeResponse.Hundredths.Get	Verifies the FDU is a Get-Time.response FDU. Obtains the Hundredths field from the FDU	2.3
(		
FDU = *Value_Expression*	The received Get-Time.response FDU.	
)		
Value_Expression =	The Set-Time.request FDU	
FDU.608.SetTimeRequest.Create	Creates the Set-Time.request FDU	2.4
(		
Year = *Value_Expression*	The year.	
DayOfYear = *Value_Expression*	The day of year.	
Hour = *Value_Expression*	The hour.	
Minute = *Value_Expression*	The minute.	
Second = *Value_Expression*	The second.	
Tenths = *Value_Expression*	The tenths.	
Hundredths = *Value_Expression*	The hundredths.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.Create	Creates the Extended-Data.request FDU specific for the Transfer Initiate command string that contains Free Text.	2.5
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
FreeText = *Value_Expression*	Readable text description of the file. Sized to 128 characters and padded with ASCII NULL character as necessary.	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.Create	Creates the Extended-Data.request FDU specific for the Transfer Initiate command string that contains no Free Text.	2.5
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
)		

***Iam* DS101Primary Functions**

Function/Parameter	Description	[REF-8]
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DataTransfer.Create	Creates the Extended-Data.request FDU specific for the Data Transfer command string.	2.5
(		
TransferControl = *Value_Expression*	Provides the sequence number that can be used in the reassembly of the data. Value is not verified to be between 0x0001 and 0xfeff.	
Data = *Value_Expression*	The data being transferred. Max size is 249 bytes.	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferComplete.Create	Creates the Extended-Data.request FDU specific for the Transfer Complete command string.	2.5
(		
FileCRC = *Value_Expression*	The overall CRC on the data that has been transferred	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferAbort.Create	Creates the Extended-Data.request FDU specific for the Transfer Abort command string.	2.5
(		
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryRequest.Create	Creates the Extended-Data.request FDU specific for the Directory Request command string.	2.5
(		
PathName = *Value_Expression*	Specifies the directory to obtain the contents of. Sized to 20 bytes; left justififed; padded with ASCII NULL characters as necessary.	
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferInitiate.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Initiate command string.	2.5

Iam **DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
(		
EntryCount = *Value_Expression*	Specifies the total number of entries in all of the selected directories to be transferred. Sized to 2 bytes.	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.Create	Creates the Extended-Data.request FDU specific for the Directory Entry command string that does not contain Free Text.	2.5
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
)		

***Iam* DS101Primary Functions**

Function/Parameter	Description	[REF-8]
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.Create	Creates the Extended-Data.request FDU specific for the Directory Entry command string that contains Free Text.	2.5
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
FreeText = *Value_Expression*	Readable text description of the file. Sized to 128 characters and padded with ASCII NULL character as necessary.	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferComplete.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Complete command string.	2.5
(		
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferAbort.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Abort command string.	2.5
(		
)		

Iam **DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.BulkErase.Create	Creates the Extended-Data.request FDU specific for the Bulk Erase command string.	2.5
(		
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileType = *Value_Expression*	Specifies the types of entries to erase. Value is not verified to follow the values defined in [REF-8]	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.SingleErase.Create	Creates the Extended-Data.request FDU specific for the Single Erase command string.	2.5
(		
PathName = *Value_Expression*	Specifies the path to the used to store the file. Sized to 20 characters.	
FileName = *Value_Expression*	Specifies the specific file name. Sized to 12 ASCII characters; left justified, padded with NULL characters as necessary.	
)		
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.FileRequest.Create	Creates the Extended-Data.request FDU specific for the File Request (Fill) command string.	2.5
(		
PathName = *Value_Expression*	Specifies the path to the used to store the file. Sized to 20 characters.	
FileName = *Value_Expression*	Specifies the specific file name. Sized to 12 ASCII characters; left justified, padded with NULL characters as necessary.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.UploadRequest.Create	Creates the Extended-Data.request FDU specific for the Update Request (Fill) command string.	2.5
(		
DataType = *Value_Expression*	Specifies the type of data to be returned. Sized to 1 byte. Values are not verified to those listed in [REF-8].	
PathName = *Value_Expression*	Specifies the path to the used to store the file. Sized to 20 characters.	
StartName = *Value_Expression*	Specifies the initial file name. Sized to 12 ASCII characters; left justified, padded with NULL characters as necessary.	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.Create	Creates the Extended-Data.request FDU specific for the Transfer Initiate command string that contains Free Text.	2.6
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
DestIdent = *Value_Expression*	Terminal identifier of the final destination of the data	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
FreeText = *Value_Expression*	Readable text description of the file. Sized to 128 characters and padded with ASCII NULL character as necessary.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.Create	Creates the Extended-Data.request FDU specific for the Transfer Initiate command string that does not contain Free Text.	2.6
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
DestIdent = *Value_Expression*	Terminal identifier of the final destination of the data	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DataTransfer.Create	Creates the Extended-Data.request FDU specific for the Data Transfer command string.	2.6
(		
TransferControl = *Value_Expression*	Provides the sequence number that can be used in the reassembly of the data. Value is not verified to be between 0x0001 and 0xfeff.	
Data = *Value_Expression*	The data being transferred. Max size is 249 bytes.	
)		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.TransferComplete.Create	Creates the Extended-Data.request FDU specific for the Transfer Complete command string.	2.6
(		
FileCRC = *Value_Expression*	The overall CRC on the data that has been transferred	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.TransferAbort.Create	Creates the Extended-Data.request FDU specific for the Transfer Abort command string.	2.6
(		
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DirectoryRequest.Create	Creates the Extended-Data.request FDU specific for the Directory Request command string.	2.6
(		
PathName = *Value_Expression*	Specifies the directory to obtain the contents of. Sized to 20 bytes; left justifed; padded with ASCII NULL characters as necessary.	
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferInitiate.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Initiate command string.	2.6
(		
EntryCount = *Value_Expression*	Specifies the total number of entries in all of the selected directories to be transferred. Sized to 2 bytes.	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	

***Iam* DS101Primary Functions**

Function/Parameter	**Description**	**[REF-8]**
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.Create	Creates the Extended-Data.request FDU specific for the Directory Entry command string that does not contain Free Text.	2.6
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.Create	Creates the Extended-Data.request FDU specific for the Directory Entry command string that contains Free Text.	2.6
(		
DataType = *Value_Expression*	The type of data being transferred. Sized to 1 byte. Value is not verified to those defined in [REF-8].	
DataLength = *Value_Expression*	Overal length of the data to be transferred.	
SourceIdent = *Value_Expression*	Terminal identifier of the originator of the data. Sized to 16 bytes.	
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileName = *Value_Expression*	File name that will store the data. Value is not verified to follow the pattern defined in [REF-8].	
FreeText = *Value_Expression*	Readable text description of the file. Sized to 128 characters and padded with ASCII NULL character as necessary.	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferComplete.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Complete command string.	2.6
(		
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferAbort.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Abort command string.	2.6
(		
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.BulkErase.Create	Creates the Extended-Data.request FDU specific for the Bulk Erase command string.	2.6
(		
PathName = *Value_Expression*	Specifies the path to the used to stroe the file. Sized to 20 characters.	
FileType = *Value_Expression*	Specifies the types of entries to erase. Value is not verified to follow the values defined in [REF-8]	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.SingleErase.Create	Creates the Extended-Data.request FDU specific for the Single Erase command string.	2.6
(		
PathName = *Value_Expression*	Specifies the path to the used to store the file. Sized to 20 characters.	
FileName = *Value_Expression*	Specifies the specific file name. Sized to 12 ASCII characters; left justified, padded with NULL characters as necessary.	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.FileRequest.Create	Creates the Extended-Data.request FDU specific for the File Request (Issue) command string.	2.6
(		

***Iam* DS101Primary Functions**		
Function/Parameter	**Description**	**[REF-8]**
PathName = *Value_Expression*	Specifies the path to the used to store the file. Sized to 20 characters.	
FileName = *Value_Expression*	Specifies the specific file name. Sized to 12 ASCII characters; left justified, padded with NULL characters as necessary.	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.UploadRequest.Create	Creates the Extended-Data.request FDU specific for the Update Request (Issue) command string.	2.6
(		
DataType = *Value_Expression*	Specifies the type of data to be returned. Sized to 1 byte. Values are not verified to those listed in [REF-8].	
PathName = *Value_Expression*	Specifies the path to the used to store the file. Sized to 20 characters.	
StartName = *Value_Expression*	Specifies the initial file name. Sized to 12 ASCII characters; left justified, padded with NULL characters as necessary.	
)		
Value_Expression =	The created FDU	
FDU.608.ECUStatusCheck.Create	Creates the Extended-Data.request FDU specific for the ECU Status Check command string.	2.7
(		
)		
Value_Expression =	The created FDU	
FDU.608.ExtendedRequestCommand.Create	Creates the Extended Command.request FDU.	2.8
(		
ECSIN = *Value_Expression*	Specifies the Extended Command Set Identifier Number. Sized to 1 byte.	
ECID = *Value_Expression*	Specifies Extended Command ID. Sized to 1 byte.	
ExtendedCommandString = *Value_Expression*	Specifies the Extended Command String. Sized to be no larger than 65,532 bytes.	
)		

6.4.2 *IAm* DS101Secondary Functions

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	"TRUE" or "FALSE" if a CONNECT.request was received within the specified time period	
DS101.Connect.Accept	Blocks execution of the scrinpt until either a CONNECT.request is received from the Primary station or the time period has expired	2.1.1.3
(		
Timeout = *Value_Expression*	The time, in milliseconds, to wait for the CONNECT.request from the Primary station (0xffffffff is infinite)	
)		
Value_Expression =	"TRUE" or "FALSE" if a DISCONNECT.request was received within the specified time period	
DS101.Disonnect.Accept	Blocks execution of the scrinpt until either a DISCONNECT.request is received from the Primary station or the time period has expired	2.1.1.9
(		
Timeout = *Value_Expression*	The time, in milliseconds, to wait for the DISCONNECT.request from the Primary station (0xffffffff is infinite)	
)		
Value_Expression =	Empty string.	
DS101.Wakeup.WaitFor	When the port was initialized with the wakeup signal, this call will pause until the wakeup signal is asserted by the primary.	N/A
(		
Port = *Value_Expression*	The port on the FTT device that will be used for DS-101 communication. Acceptable values are "A" or "B".	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	TRUE or FALSE.	
DS101.LinkLayer.Message.Available	Determines if an FDU is available for processing. Can be used to poll for FDUs.	N/A
(		
)		
Value_Expression =	The DS-101 Address field from the Set Address.request FDU	
FDU.SetAddressRequest.DS101Address.Get	Verifies the FDU is a Set Address.request FDU. Obtains the DS-101 Address field from the FDU	3.3.5
(		
FDU = *Value_Expression*	The received Set Address.request FDU	
)		
Value_Expression =	The Station ID field from the Set Identifier.request FDU	
FDU.SetIdentifierRequest.StationID.Get	Verifies the FDU is a Set Identifier.request FDU. Obtains the Station ID field from the FDU.	3.3.6
(		
FDU = *Value_Expression*	The received Set Identifier.request FDU	
)		
Value_Expression =	The Frame Count field from the AXID.req FDU	
FDU.AXIDrequest.FrameCount.Get	Verifies the FDU is a AXID.req FDU. Obtains the Frame Count field from the FDU	3.3.7
(		
FDU = *Value_Expression*	The received AXID.req FDU	
)		
Value_Expression =	The Fixed ID field from the AXID.req FDU	
FDU.AXIDrequest.FixedID.Get	Verifies the FDU is a AXID.req FDU. Obtains the Fixed ID field from the FDU	3.3.7
(		
FDU = *Value_Expression*	The received AXID.req FDU	

IAm DS101Secondary Functions

Function/Parameter	Description	REF-7
)		
Value_Expression =	The Station ID field from the AXID.req FDU	
FDU.AXIDrequest.StationID.Get	Verifies the FDU is a AXID.req FDU. Obtains the Station ID field from the FDU	3.3.7
(		
FDU = *Value_Expression*	The received AXID.req FDU	
)		
Value_Expression =	AXID.resp FDU	
FDU.AXIDresponse.Create	Creates an AXID.resp FDU	3.3.8
(		
Framecount = *Value_Expression*	The number of received frames that the station can accept before an acknowledgement must be sent. There is no check for the values of 1 or 7 per the EKMS spec.	
FixedID = *Value_Expression*	The FixedID of the sender (the FTT)	
StationID = *Value_Expression*	The StationID of the sender (the FTT)	
)		
Value_Expression =	The Start Byte field from the Field Specifier.request FDU	
FDU.FieldSpecifierRequest.StartByte.Get	Verifies the FDU is a Field Specifier.request FDU. Obtains the Start Byte field from the FDU.	3.3.9
(		
FDU = *Value_Expression*	The received Field Specifier.request FDU	
)		
Value_Expression =	The Length field from the Field Specifier.request FDU	
FDU.FieldSpecifierRequest.Length.Get	Verifies the FDU is a Field Specifier.request FDU. Obtains the Length field from the FDU.	3.3.9
(		
FDU = *Value_Expression*	The received Field Specifier.request FDU	
)		
Value_Expression =	The Path field from the File Header.request (Fill) FDU	

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	**REF-7**
FDU.FileHeaderRequest.Fill.Path.Get	Verifies the FDU is a File Header.request (Fill) FDU. Obtains the Path field from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
)		
Value_Expression =	The Name field from the File Header.request (Fill) FDU	
FDU.FileHeaderRequest.Fill.Name.Get	Verifies the FDU is a File Header.request (Fill) FDU. Obtains the Name field from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
)		
Value_Expression =	The TrKEK Count field from the File Header.request (Fill) FDU	
FDU.FileHeaderRequest.Fill.TrKEKCount.Get	Verifies the FDU is a File Header.request (Fill) FDU. Obtains the TrKEK Count field from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
)		
Value_Expression =	The specified TrKEK field from the File Header.request (Fill) FDU	
FDU.FileHeaderRequest.Fill.TrKEK.Get	Verifies the FDU is a File Header.request (Fill) FDU. Obtains the specified TrKEK from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
TrKEKNumber = *Value_Expression*	The number of the TrKEK to obtain. Must be between 0 and 5 inclusive. Zero returned otherwise. Zero returned if the specified key was not within the FDU	
)		
Value_Expression =	The Path field from the File Header.request (Issue) FDU	

***IAm* DS101Secondary Functions**

Function/Parameter	Description	REF-7
FDU.FileHeaderRequest.Issue.Path.Get	Verifies the FDU is a File Header.request (Issue) FDU. Obtains the Path field from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
)		
Value_Expression =	The Name field from the File Header.request (Issue) FDU	
FDU.FileHeaderRequest.Issue.Name.Get	Verifies the FDU is a File Header.request (Issue) FDU. Obtains the Name field from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
)		
Value_Expression =	The TrKEK Count field from the File Header.request (Issue) FDU	
FDU.FileHeaderRequest.Issue.TrKEKCount.Get	Verifies the FDU is a File Header.request (Issue) FDU. Obtains the TrKEK Count field from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
)		
Value_Expression =	The specified TrKEK field from the File Header.request (Issue) FDU	
FDU.FileHeaderRequest.Issue.TrKEK.Get	Verifies the FDU is a File Header.request (Issue) FDU. Obtains the specified TrKEK from the FDU.	3.3.12
(		
FDU = *Value_Expression*	The received File Header.request	
TrKEKNumber = *Value_Expression*	The number of the TrKEK to obtain. Must be between 0 and 5 inclusive. Zero returned otherwise. Zero returned if the specified key was not within the FDU	
)		
Value_Expression =	The Key/Data from the Key/Data.request FDU	

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**REF-7**
FDU.KeyDataRequest.KeyData.Get	Verifies the FDU is a Key/Data.request FDU (with the FCI of 0x01b0) Obtains the Key/Data field from the Key/Data.request FDU	3.3.12
(		
FDU = *Value_Expression*	The received Key/Data.request FDU	
)		
Value_Expression =	The DS-100 Tag from the Key/Data.request FDU	
FDU.KeyDataRequest.DS100Tag.Get	Verifies the FDU is a Key/Data.request FDU (FCI of 0x02b0) Obtains the DS-100 Tag from the FDU	3.3.12
(		
FDU = *Value_Expression*	The received Key/Data request.FDU	
)		
Value_Expression =	The DS-100 Tag from the Key/Data.request FDU including the Key Data and Text fields.	
FDU.KeyDataRequest.DS100Tag.GetFull	Verifies the FDU is a Key/Data.request FDU (FCI of 0x02b0) Obtains the DS-100 Tag from the FDU	3.3.12
(		
FDU = *Value_Expression*	The received Key/Data request.FDU	
)		
Value_Expression =	The Key/Data from the Key/Data.request FDU	
FDU.KeyDataRequest.KeyData.WithTag.Get	Verifies the FDU is a Key/Data.request FDU (FCI of 0x02b0) Obtains the Key/Data from the FDU	3.3.12
(		
FDU = *Value_Expression*	The received Key/Data.request FDU	
)		
Value_Expression =	The Text field from the Key/Data.request FDU represented in Hex. Use FTT.HexToString to convert to string if necessary.	
FDU.KeyDataRequest.Text.Get	Verifies the FDU is a Key/Data.request FDU (FCI of 0x02b0). Obtains the Text field from the Key/Data.request FDU.	3.3.12
(		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**REF-7**
FDU = *Value_Expression*	The received Key/Data FDU	
)		
Value_Expression =	The Program Data File field from the FDU	
FDU.ProgramFileDataRequest.Data.Get	Verifies the FDU is a Program/File Data.request Obtains the Data field from the FDU.	3.3.13
(		
FDU = *Value_Expression*	The received Program/File Data FDU.	
)		
Value_Expression =	TRUE or FALSE	
FDU.ProgramFileDataRequest.IsIssue	Verifies the FDU is a Program/File Data.request Returns TRUE/FALSE if the FDU is an Issue FDU.	3.3.13
(		
FDU = *Value_Expression*	The received Program/File Data FDU.	
)		
Value_Expression =	The CIK Split Data from the FDU	
FDU.CIKSplitRequest.Fill.CIKSplitData.Get	Verifies the FDU is a CIK Split.request FDU (fill) Obtains the CIK Split Data field.	3.3.14
(		
FDU = *Value_Expression*	The received CIK Split.request FDU	
)		
Value_Expression =	The CIK Split Data from the FDU	
FDU.CIKSplitRequest.Issue.CIKSplitData.Get	Verifies the FDU is a CIK Split.request FDU (issue) Obtains the CIK Split Data field.	3.3.14
(		
FDU = *Value_Expression*	The received CIK Split.request FDU	
)		
Value_Expression =	The DTD Split Data field from the DTD Split.request FDU	
FDU.DTDSplitRequest.Fill.DTDSplitData.Get	Verifies the FDU is a DTD Split.request FDU (FCI of 0x01d0) Obtains the DTD Split Data field.	3.3.15
(		
FDU = *Value_Expression*	The received DTD Split.request FDU	

IAm DS101Secondary Functions

Function/Parameter	Description	REF-7
)		
Value_Expression =	The DTD Split Data field from the DTD Split.request FDU (FCI of 0x02d0)	
FDU.DTDSplitRequest.Issue.DTDSplitData.Get	Verifies the FDU is a DTD Split.request FDU Obtains the DTD Split Data field.	3.3.15
(		
FDU = *Value_Expression*	The received DTD Split.request FDU	
)		
Value_Expression =	The TrKEK Data from the FDU	
FDU.TrKEKRequest.Fill.TrKEKData.Get	Verifies the FDU is a TrKEK.request FDU (fill variant) Obtains the TrKEK Data field from the FDU.	3.3.16
(		
FDU = *Value_Expression*	The received TrKEK.request FDU.	
)		
Value_Expression =	The TrKEK Data from the FDU	
FDU.TrKEKRequest.Issue.TrKEKData.Get	Verifies the FDU is a TrKEK.request FDU (issue variant) Obtains the TrKEK Data field from the FDU.	3.3.16
(		
FDU = *Value_Expression*	The received TrKEK.request FDU.	
)		
Value_Expression =	TRUE or FALSE	
FDU.TrKEKRequest.IsIssue	Verifies the FDU is a TrKEK.request FDU Returns TRUE/FALSE if the FDU is an Issue FDU.	3.3.16
(		
FDU = *Value_Expression*	The received TrKEK.request FDU.	
)		
Value_Expression =	The Command.request FDU	
FDU.CommandRequest.Create	Creates the Command.request FDU	3.3.17
(		
CommandString = *Value_Expression*	The command string.	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The Command String from the Command.request FDU	
FDU.CommandRequest.CommandString.Get	Verifies the FDU is a Command.request FDU. Obtains the Command String field.	3.3.17
(		
FDU = *Value_Expression*	The received Command.request FDU.	
)		
Value_Expression =	The DU Data field of the FDU	
FDU.ExtendedData.Fill.DUData.Get	Verifies the FDU is a Extended Data.request FDU (Fill variant) Obtains the DU Data field of the FDU	3.3.18
(		
FDU = *Value_Expression*	The received FDU	
)		
Value_Expression =	The XDU Control field of the FDU	
FDU.ExtendedData.Fill.XDUControl.Get	Verifies the FDU is a Extended Data.request FDU (Fill variant) Obtains the XDU Control field of the FDU	3.3.18
(		
FDU = *Value_Expression*	The received FDU	
)		
Value_Expression =	The DU Data field of the FDU	
FDU.ExtendedData.Issue.DUData.Get	Verifies the FDU is a Extended Data.request FDU (Issue variant) Obtains the DU Data field of the FDU	3.3.18
(		
FDU = *Value_Expression*	The received FDU	
)		
Value_Expression =	The XDU Control field of the FDU	
FDU.ExtendedData.Issue.XDUControl.Get	Verifies the FDU is a Extended Data.request FDU (Issue variant) Obtains the XDU Control field of the FDU	3.3.18
(		
FDU = *Value_Expression*	The received FDU	
)		

IAm DS101Secondary Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	The Get Time.response FDU	
FDU.GetTimeResponse.Create	Creates the Get Time.response FDU	3.3.20
(		
Year = *Value_Expression*	The year field	
Day_Of_Year = *Value_Expression*	The day of year	
Hour = *Value_Expression*	The hour	
Minute = *Value_Expression*	The minute	
Second = *Value_Expression*	The second	
Tenths = *Value_Expression*	The tenths	
Hundredths = *Value_Expression*	The hundredths	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The ECSIN from the Extended Command.request FDU	
FDU.608.ExtendedCommandRequest.ECSIN.Get	Verifies the FDU is an Extended Command.request FDU. Obtains the ECSIN field.	2.8
(		
FDU = *Value_Expression*	The received Extended Command.request FDU.	
)		
Value_Expression =	The ECID from the Extended Command.request FDU	
FDU.608.ExtendedCommandRequest.ECID.Get	Verifies the FDU is an Extended Command.request FDU. Obtains the ECID field.	2.8
(		
FDU = *Value_Expression*	The received Extended Command.request FDU.	
)		
Value_Expression =	The ECMC from the Extended Command.request FDU	
FDU.608.ExtendedCommandRequest.ECMC.Get	Verifies the FDU is an Extended Command.request FDU. Obtains the ECMC field.	2.8
(		
FDU = *Value_Expression*	The received Extended Command.request FDU.	
)		
Value_Expression =	The Command String from the Extended Command.request FDU	
FDU.608.ExtendedCommandRequest.CommandString.Get	Verifies the FDU is an Extended Command.request FDU. Obtains the Command String field.	2.8
(		
FDU = *Value_Expression*	The received Extended Command.request FDU.	
)		

IAm DS101Secondary Functions

Function/Parameter	Description	[REF-8]
Value_Expression =	The Get-Time.response FDU	
FDU.608.GetTimeResponse.Create	Creates the Get-Time.response FDU	2.3
(		
Year = *Value_Expression*	The year. Sized to 7 bits.	
DayOfYear = *Value_Expression*	The day of year. Sized to 9 bits.	
Hour = *Value_Expression*	The hour. Sized to 8 bits.	
Minute = *Value_Expression*	The minute. Sized to 8 bits.	
Second = *Value_Expression*	The second. Sized to 8 bits.	
Tenths = *Value_Expression*	The tenths. Sized to 4 bits.	
Hundredths = *Value_Expression*	The hundredths. Sized to 4 bits.	
)		
Value_Expression =	The Year field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Year.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Year field from the FDU	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		
Value_Expression =	The Day field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Day.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Day field from the FDU	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		
Value_Expression =	The Hour field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Hour.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Hour field from the FDU	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		

IAm DS101Secondary Functions

Function/Parameter	Description	[REF-8]
Value_Expression =	The Minute field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Minute.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Minute field from the FDU	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		
Value_Expression =	The Second field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Second.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Second field from the FDU	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		
Value_Expression =	The Tenths field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Tenths.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Tenths field from the FDU.	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		
Value_Expression =	The Hundredths field from the Set-Time.request FDU	
FDU.608.SetTimeRequest.Hundredths.Get	Verifies the FDU is a Set-Time.request FDU. Obtains the Hundredths field from the FDU.	2.4
(		
FDU = *Value_Expression*	The received Set-Time.request FDU.	
)		

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Data Type field form the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.DataType.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate Command String. Obtains the Data Type field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Data Length field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.DataLength.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate Command String. Obtains the Data Length field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Source Ident field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.SourceIdent.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate Command String. Obtains the Source Ident field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	[REF-8]
Value_Expression =	The Path Name field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.PathName.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate Command String. Obtains the Path Name field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The File Name field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.FileName.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate Command String. Obtains the File Name field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Free Text field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.FreeText.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate Command String. Obtains the Free Text field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	[REF-8]
Value_Expression =	The Data field of the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DataTransfer.Data.Get	Verifies the Extended-Data.request FDU for a Data Transfer Command String. Obtains the Data field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended Data.request FDU	
)		
Value_Expression =	The Transfer Control field of the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DataTransfer.TransferControl.Get	Verifies the Extended-Data.request FDU for a Data Transfer Command String. Obtains the Transfer Control field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended Data.request FDU	
)		
Value_Expression =	The File CRC from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.TransferComplete.FileCRC.Get	Verifies the Extended-Data.request FDU for a Data Transfer Complete Command String. Obtains the Data field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended Data.request FDU	
)		
Value_Expression =	The Path Name field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryRequest.PathName.Get	Verifies the Extended-Data.request FDU for a Data Directory Request Command String. Obtains the Path_Name field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Data Type field from the Extended-Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryRequest.DataType.Get	Verifies the Extended-Data.request FDU for a Data Directory Request Command String. Obtains the Path_Name field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Entry Count field from the Extended Data.request (Fill) FDU	
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferInitiate.EntryCount.Get	Verifies the Extended-Data.request FDU for a Directory Transfer Initiate Command String. Obtains the Entry_Count field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Data_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.DataType.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the Data_Type field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Data_Length field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.DataLength.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the Data_Length field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Source_Ident field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.SourceIdent.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the Source_Ident field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The Dest_Ident field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.DestIdent.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the Dest_Ident field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		

IAm **DS101Secondary Functions**

Function/Parameter	Description	[REF-8]
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.PathName.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the Path_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.FileName.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the File_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		
Value_Expression =	The FreeText field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.FreeText.Get	Verifies the Extended-Data.request FDU for a Directory Entry Command String. Obtains the Free_Text field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU	
)		

***IAm* DS101Secondary Functions**

Function/Parameter	Description	[REF-8]
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.BulkErase.PathName.Get	Verifies the Extended-Data.request FDU for a Bulk Erase Command String. Obtains the Path_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The File_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.BulkErase.FileType.Get	Verifies the Extended-Data.request FDU for a Bulk Erase Command String. Obtains the File_Type field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.SingleErase.PathName.Get	Verifies the Extended-Data.request FDU for a Single Erase Command String. Obtains the Path_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

IAm DS101Secondary Functions		
Function/Parameter	**Description**	[REF-8]
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.SingleErase.FileName.Get	Verifies the Extended-Data.request FDU for a Single Erase Command String. Obtains the File_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.FileRequest.PathName.Get	Verifies the Extended-Data.request FDU for a File Request Command String. Obtains the Path_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.FileRequest.FileName.Get	Verifies the Extended-Data.request FDU for a File Request Command String. Obtains the File_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request (Fill) FDU.	
)		

IAm DS101Secondary Functions		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Data_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.UploadRequest.DataType.Get	Verifies the Extended-Data.request FDU for a Upload Request (Fill) Command String. Obtains the Data_Type field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.UploadRequest.PathName.Get	Verifies the Extended-Data.request FDU for a Upload Request (Fill) Command String. Obtains the Path_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Start_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Fill.UploadRequest.StartName.Get	Verifies the Extended-Data.request FDU for a Upload Request (Fill) Command String. Obtains the Start_Name field from the FDU.	2.5
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	[REF-8]
Value_Expression =	The Data_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.DataType.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the Data_Type field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Data_Length field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.DataLength.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the Data_Length field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Source_Ident field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.SourceIdent.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the Source_Ident field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Dest_Ident field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.DestIdent.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the Dest_Ident field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.PathName.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.FileName.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the File_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Free_Text field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.FreeText.Get	Verifies the Extended-Data.request FDU for a Transfer Initiate (Issue) Command String. Obtains the Free_Text field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Data field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DataTransfer.Data.Get	Verifies the Extended-Data.request FDU for a Data Transfer (Issue) Command String. Obtains the Data field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Transfer Control field of the Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DataTransfer.TransferControl.Get	Verifies the Extended-Data.request FDU for a Data Transfer Command String. Obtains the Transfer Control field from the FDU	2.5
(		
FDU = *Value_Expression*	The received Extended Data.request FDU	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The File_CRC field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.TransferComplete.FileCRC.Get	Verifies the Extended-Data.request FDU for a Transfer Complete (Issue) Command String. Obtains the File_CRC field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryRequest.PathName.Get	Verifies the Extended-Data.request FDU for a Directory Request (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Data_type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryRequest.DataType.Get	Verifies the Extended-Data.request FDU for a Directory Request (Issue) Command String. Obtains the Data_Type field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

***IAm* DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Entry_Count field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferInitiate.EntryCount.Get	Verifies the Extended-Data.request FDU for a Directory Transfer Initiate (Issue) Command String. Obtains the Entry_Count field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Data_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.DataType.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the Data_Type field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Data_Length field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.DataLength.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the Data_Length field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

IAm **DS101Secondary Functions**		
Function/Parameter	**Description**	**[REF-8]**
Value_Expression =	The Source_Ident field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.SourceIdent.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the Source_Ident field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Dest_Ident field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.DestIdent.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the Dest_Ident field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.PathName.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

***IAm* DS101Secondary Functions**

Function/Parameter	Description	[REF-8]
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.FileName.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the File_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Free_Text field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.FreeText.Get	Verifies the Extended-Data.request FDU for a Directory Entry (Issue) Command String. Obtains the Free_Text field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Extended-Data.request (Issue) FDU	
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferAbort.Create	Creates the Extended-Data.request FDU specific for the Directory Transfer Abort command string.	2.6
(		
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.BulkErase.PathName.Get	Verifies the Extended-Data.request FDU for a Bulk Erase (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

IAm DS101Secondary Functions

Function/Parameter	**Description**	[REF-8]
Value_Expression =	The File_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.BulkErase.FileType.Get	Verifies the Extended-Data.request FDU for a Bulk Erase (Issue) Command String. Obtains the File_Type field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.SingleErase.PathName.Get	Verifies the Extended-Data.request FDU for a Single Erase (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.SingleErase.FileName.Get	Verifies the Extended-Data.request FDU for a Single Erase (Issue) Command String. Obtains the File_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

***IAm* DS101Secondary Functions**

Function/Parameter	**Description**	[REF-8]
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.FileRequest.PathName.Get	Verifies the Extended-Data.request FDU for a File Request (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The File_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.FileRequest.FileName.Get	Verifies the Extended-Data.request FDU for a File Request (Issue) Command String. Obtains the File_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Data_Type field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.UploadRequest.DataType.Get	Verifies the Extended-Data.request FDU for a Upload Request (Issue) Command String. Obtains the Data_Type field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

***IAm* DS101Secondary Functions**

Function/Parameter	**Description**	[REF-8]
Value_Expression =	The Path_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.UploadRequest.PathName.Get	Verifies the Extended-Data.request FDU for a Upload Request (Issue) Command String. Obtains the Path_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		
Value_Expression =	The Start_Name field from the Extended-Data.request FDU.	
FDU.608.ExtendedDataRequest.Issue.UploadRequest.StartName.Get	Verifies the Extended-Data.request FDU for a Upload Request (Issue) Command String. Obtains the Start_Name field from the FDU.	2.6
(		
FDU = *Value_Expression*	The received Extended-Data.request FDU.	
)		

6.4.3 *IAm* DS101Monitoring Functions

***IAm* DS101Monitor Functions**	
Function/Parameter	**Description**
Value_Expression =	Results of initializing the FTT for DS-101 monitoring
DS101.FrameLayer.Monitor	Puts the FTT in a mode to monitor all data seen on the differential line.
(	
Port = *Value_Expression*	The port on the FTT device that will be used for DS-101 monitoring. Acceptable values are “A” or “B”. Port “C” can not be monitored.
ConnectAtoB = *Value_Expression*	Whether to physically connect port A and B together thus allowing a Primary to be connected to one port and a Secondary to the other port without the need of a Y cable. Default is FALSE.
)	
Value_Expression =	Results of causing a physcial connection between the two ports on the FTT.
DS101.ConnectPhysical	Physically connects the two ports on the FTT internally.
(	
Port = *Value_Expression*	The port on the FTT that will be connected. Acceptable values are “A” or “B”. Port “C” can not be monitored.
)	
Value_Expression =	Results of causing a physcial disconnection between the two ports on the FTT.
DS101.DisconnectPhysical	Physically disconnects the two ports on the FTT internally.
(	
Port = *Value_Expression*	The port on the FTT that will be connected. Acceptable values are “A” or “B”. Port “C” can not be monitored.
)	

6.5 DS-102 Functions

DS-102 Functions		
Function/Parameter	**Description**	**REF-7**
Value_Expression =	Result of the transmission request	
DS102.Transmit	Transmit the number of bits out the DS-102 port.	
(		
Data = *Value_Expression*	The key data to transmit. Only the first BitCount bits will be transmitted from Most significant to least significant.	
BitCount = *Value_Expression*	Number of bits to transmit.	
InvertClock = *Value_Expression*	Invert the normal polarity of the clock. This is useful in debugging to see if receiver has the correct polarity. Default is FALSE.	
ClockPeriodUseconds = *Value_Expression*	The number of microseconds for the period of the clock. One divided by this number is the clock frequency. Default is 300 microseconds.	
ClockSkewUseconds = *Value_Expression*	The amount of microseconds to skew the clock. This is useful in testing a DS-102 design to determine if an irregular clock works. The skew is applied over four clocks by moving each edge of the clock by this amount.	
MaxWaitMsec= *Value_Expression*	The maximum number of milliseconds to wait for data to be received. Default is 15 seconds.	
Sense= *Value_Expression*	The state to put the sense line during transmission. Default is FALSE.	
CTS= *Value_Expression*	Wait for CTS to go low before transmitting. Default is FALSE.	
InvertCTS= *Value_Expression*	Invert the normal polarity of CTS. This is useful in debugging new DS-102 circuits which might have accidentally inverted the CTS line. Default is FALSE.	
CTSUnsetFirst= *Value_Expression*	Before sampling CTS, make sure it is in the inactive state prior to it going to the active state. Default is FALSE.	
ClockWhileWaitForCTS= *Value_Expression*	While waiting for CTS output a clock	

DS-102 Functions		
Function/Parameter	**Description**	**REF-7**
	signal. Default is FALSE.	
)		
Value_Expression =	Received data.	
DS102.Receive	Requests that the specified key data is transmitted to the connected receiver.	
(		
MaxBitCount = *Value_Expression*	Maximum number of bits to receive before returning.	
MaxWaitMsec = *Value_Expression*	The maximum number of milliseconds to wait for data to be received. Default is 15 seconds.	
InvertClock = *Value_Expression*	Invert the Clock signal. Useful for debugging new circuits which might have inverted clock. Default is FALSE.	
WaitForClock = *Value_Expression*	Do not assert CTS until a clock is received. Default is FALSE.	
CTS = *Value_Expression*	Assert CTS per EKMS 308. Data starts on the next clock edge. Default is FALSE.	
InvertCTS = *Value_Expression*	Invert the CTS signal. Useful for debugging new circuits which might have inverted CTS. Default is FALSE.	
CTSusec = *Value_Expression*	Assert CTS for this number of micro seconds. Default is 0 which indicates to follow EKMS 308.	
ClockLowToCTSusec = *Value_Expression*	Wait this number of microseconds after the received clock goes low before asserting CTS. Default is 0 which indicates to follow EKMS 308 timing.	
ClockLowToCTSHighusec = *Value_Expression*	Wait this number of microseconds after the clock goes low to deassert CTS. Default is 0 which indicates to follow EKMS 308 timing.	
ClockLossEnd= *Value_Expression*	Whether to measure for the clock to have stopped in order to group bits together. This overrides the PollingIntervalMsec if the clock slows	

DS-102 Functions		
Function/Parameter	**Description**	**REF-7**
	below 10 Hertz (100 milliseconds). Default is TRUE.	
)		
Value_Expression =	The number of bits received on the last DS102.Receive call.	
DS102.ReceivedBitCount	Retrieves the number of bits received during the last reception of DS-102 data.	
(		
)		
Value_Expression =	The return value is insignficant.	
DS102.Monitor	Monitor the DS-102 signals and report data received in groups of bits.	
(		
MaxBits= *Value_Expression*	The maximum number of bits to receive in one group. Default is 1024 bits.	
InvertClock= *Value_Expression*	TRUE means to assume the input clock is inverted. Useful for testing of devices which may have their clock inverted due to design error. Default is FALSE.	
PollingIntervalMsec= *Value_Expression*	The amount of milliseconds to wait for a group of bits to be received together. Default is 60 seconds.	
ClockLossEnd= *Value_Expression*	Whether to measure for the clock to have stopped in order to group bits together. This overrides the PollingIntervalMsec if the clock slows below 10 Hertz (100 milliseconds). Default is TRUE.	
)		

7 DIRECTIVES

7.1 DIRECTIVE TABLE

Directive	Meaning
Include	Includes the specified file as though the entire of the contents were in the current file.
Filename	*Value_Expression* which evaluates to a filename. Path is assumed to be in the current directory.
Print	Output the following values to the output window.
string sequence	{ *Value_Expression* } which each evaluate to a string or integer.
Exit	Script was successful and output the following values to the output window.
string sequence	{ *Value_Expression* } which each evaluate to a string or integer.
Abort	Script failed and output the following values to the output window.
string sequence	{ *Value_Expression* } which each evaluate to a string or integer.
IAm	Establish whether the script is written as a terminal simulator or a card simulator. A script can only contain one IAm directive and it should appear at the beginning of the script.
Terminal	Terminal simulator with access to APDU level communications. This is the default.
Card	Card simulator with access to APDU level communications.
None	Has byte-level access to the communication port. Can behave as a terminal or a card. High-level functions are unavailable due to conflicts in the high-level protocol with direct communication access. This mode is intended to be used for testing low-level protocols.
DS101Primary	Causes the FTT to act as the primary DS-101 Station
DS101Secondary	Causes the FTT to act as the secondary DS-101 Station
DS101Monitor	Monitors DS-101 frame layer traffic between a primary and a secondary DS-101 stations
Session	Establish or close a session. This has the effect of making sure that both the terminal and card are synchronized together. These are smart-card only directives.
Start	Begin a new session. This starts a new arbitration sequence for ISO 14443 protocols. This causes the card to be reset for a ISO 7816 T=1 session.
End	End the current session. This deselects the card for ISO 14443. This has no effect on a T=1 session.

8 EXAMPLE SCRIPTS

8.1 SMART CARD RELATED SCRIPTS

8.1.1 KEYS.FTT

```
Master_Key = 0x123456789ABCDEF00FEDCBA987654321 ;
Master_Key_File_ID = 30 ;
Master_Key_ID = 1 ;
```

8.1.2 MUTUAL_AUTHENTICATION.FTT

```
// Mutual Authenticate Macro:
// Mutually authenticate the terminal with the card and vica versa.
//
include “keys.FTT” ;  // get the key definitions

Rapdu = authentication.mutual.initialize
          ( DiversifierOffset = 1,
            FileID = Master_Key_File_ID,
            KeyID = Master_Key_ID );
Authentication_Data = APDU.Response.Data( APDU = Rapdu );
If ( RAPDU.SW != 0x9000 )
then abort “mutual authentication init failed with “ & RAPDU.SW ;
endif ;

Diversifier = Authentication_Data[ 1 .. 8 ] ;
RN_C = Authentication_Data[ 9 .. 16 ] ;
Key_Version = Authentication_Data[ 1 ] ;

diversified_key = data.des.ecb.encrypt( Data = diversifier,
                                        Key = Master_Key );
session_key = data.des.ecb.encrypt( Data = RN_C,
                                    key = diversified_key );
print “Session key => “ & session_key ;


RN_T = Data.Random.Get( ByteCount = 8 );
Authentication_data = APDU.Response.Data( APDU = Authentication.Mutual
                                                   (TerminalRandomNumber = RN_T,
                                                     CardRandomNumber = RN_C,
                                                     Key = Session_Key,
                                                     AlgorithmID = “T” ) );
if RAPDU.SW != 0x9000
```

```
then abort “mutual authentication failed with “ & RAPDU.SW ;
endif ;
// Note that the CommitData was initialized by the mutual authentication.
```

8.1.3 COMMIT.FTT

```
// Commit macro:
// Use the CommitData buffer and session key to commit to the card.
//
include “keys.ftt”  ;  // get the key definitions


authentication.commit( Key = session_key );
If RAPDU.SW != 0x9000
Then abort “Commit failed with  “ & RAPDU.SW ;
Endif ;
```

8.1.4 TESTCASE_01.FTT

```
// Category: Test case 01
// Description: Verify that a mutual authenticate can not be executed
//              more than once after the initialization of mutual authenticate.
//
IAm terminal ;                 // we are simulating the terminal

session start ;                // start the session
print “Executing Test Case 01:”;

include “mutual_authenticate.ftt”;  // do a mutual authenticate with the card

Authentication_data = Authentication.Mutual( TerminalRandomData = 0x12345678,
                                            CardRandomDataSignature = RN_C,
                                            Key = 0xABCDABCD );
If RAPDU.SW != SW_CommandOutOfSequence
Then abort “Second authentication failed with “ & RAPDU.SW ;
Endif ;


session end ;                  // end the session


Exit “Test Case 01: Passed”;
```

8.1.5 TESTCASE_02.FTT

```
// Category: Test case 02
// Description: Mutual authentication with commit.
// Mutually authenticate.
// Update several files.
// Commit changes.
//
IAm terminal ;                // we are simulating the terminal

AFI = 0x01 ;       // in case this is 14443 set the AFI for selecting a card
session start ;                // start the session
print “Executing Test Case 02:” ;

include “mutual_authenticate.ftt” ;  // do a mutual authenticate with the card


// Note that CommitData is updated by this command.
File.Binary.Update( FileID = 5, Data=0x44332211 ) ;
If RAPDU.SW != 0x9000 Then abort “Update Binary failed with “ & RAPDU.SW ; Endif ;

// Note that CommitData is updated by this command.
File.Binary.Update( FileID = 6, ByteOffset=7 , Data=0x665544332211 ) ;
If RAPDU.SW != 0x9000 Then abort “Update Binary failed with “& RAPDU.SW ; Endif ;

// Note that CommitData is updated by this command.
File.Binary.Update( FileID = 22, ByteOffset=8 , Data=0x11 ) ;
If RAPDU.SW != SW_SecurityNotSatisfied
Then abort “Update binary should have failed on security but SW was “ & RAPDU.SW ;
Endif ;


include “commit.ftt” ;        // commit the commands to the card.

session end ;                 // end the session

Exit “Test Case 02: Passed” ;
```

8.2 DS-101 Related Scripts

8.2.1 VerifyAXIDResponse.inc

```
//
// The following need to be set prior to including this macro:
// --------------------------------------------------------------------
// VerifiedFrameCount..........1 byte
// VerifiedDULength............2 bytes
// VerifiedStationID..........14 bytes
//
// ReceivedFDU.................The AXID.resp FDU
//
//
//
// The following variables are set after including this macro:
// -----------------------------------------------------------
// FrameCount.........The received Frame Count from the AXID.resp FDU
// FixedID............The received Fixed ID from the AXID.resp FDU
// StationID..........The Station ID from the AXID.resp FDU
//
//
//
//
// Verify all variables are set
if (VerifiedFrameCount = Uninitialized666) then
    abort "Script Error: VerifiedFrameCount is not set!!";
endif;

if (VerifiedFixedID = Uninitialized666) then
    abort "Script Error: VerifiedFixedID is not set!!";
endif;

if (VerifiedStationID = Uninitialized666) then
    abort "Script Error: VerifiedStationID is not set!!";
endif;

if (ReceivedFDU = Uninitialized666) then
    abort "Script Error: ReceivedFDU is not set!!";
endif;



// Verify the common FDU fields
VerifiedFCI      = 0x0060;
VerifiedDULength = 0x0011;

include "VerifyFDU.inc";


// Get the fields of the AXID.resp
FrameCount = fdu.axidresponse.framecount.get
```

```
             (
                fdu = ReceivedFDU
             );
print "FrameCount => " & FrameCount;


FixedID = fdu.axidresponse.fixedid.get
          (
             fdu = ReceivedFDU
          );
print "FixedID => " & FixedID;


StationID = fdu.axidresponse.stationid.get
            (
               fdu = ReceivedFDU
            );
print "StationID => " & StationID;



// Verify the fields of the AXID.resp
if (FrameCount != VerifiedFrameCount) then
    abort "Invalid Frame Count!!";
endif;


if (FixedID != VerifiedFixedID) then
    abort "Invalid Fixed ID!!";
endif;


if (StationID != VerifiedStationID) then
    abort "Invalid Station ID!!";
endif;
```

8.2.2 AXID.FTT

```
// Category: AXID.req/AXID.res Scenario
// Description: Verifies a normal AXID Request/Response scenario

IAm DS101Primary;


// Initialize the DS-101 Interface
Result = ds101.initialize
        (
            port              = "a",
            address           = 0x55,
            retrycount        = 0x03,
            maxtxframes       = 0x01,
            acceptconnections = "TRUE",
            monitor           = "FALSE"
          );

if (Result != "Success") then
    abort "DS-101 Initialization failed!!";
endif;


// CONNECT.request

// ****************************************************************************
// Notify user to make the connection
// ****************************************************************************
Result = ftt.prompt.user
         (
            message = "Press button above DS-101 connection (be sure LED is on)"
         );



// ****************************************************************************
// CONNECT.request
// ****************************************************************************
Result = ds101.connect
         (
            Address = DS101_ConnectAddress
         );


if (Result != ConnectConfirmed) then
    print "Result => " & Result;
    abort "CONNECT.request failed!!";
endif;



print "Secondary DS-101 Address => " & SecondaryDS101Address;
```

```
// Formulate and transmit an AXID.req FDU
AXIDreqFDU = fdu.axidrequest.create
             (
                 framecount = 0x01,
                 fixedid    = 0x55, // FTT's address
                 stationid  = "FTT Station ID" // FTT's station ID
             );

Result = ds101.fdu.transmit
         (
            fdu = AXIDreqFDU
         );

print "Result => " & Result;

if (Result != TransmissionConfirmed) then
    abort "Failure in transmitting AXID.req FDU!!";
endif;



// Receive and verify the AXID.resp
AXIDResponseFDU = ds101.fdu.receive
                  (
                      timeout = Timeout
                  );



VerifiedFrameCount = 0x01;
VerifiedFixedID    = 0x50;
VerifiedStationID  = "Verified S ID ";
ReceivedFDU        = AXIDResponseFDU;

include "VerifyAXIDresponse.inc";



// DISCONNECT.request
// ****************************************************************************
// DISCONNECT.request
// ****************************************************************************
Result = ds101.disconnect();

if (Result != DisconnectConfirm) then
    abort "DISCONNECT.request failed!!";
endif;



// ****************************************************************************
// Notify user to break the connection
// ****************************************************************************
Result = ftt.prompt.user
         (
            message = "Press button above DS-101 connection (be sure LED is off)"
         );
```

```
print "Test passed";
```

8.2.3 DS-101 MONITORING

```
iam ds101monitor;

// ****************************************************************************
// Initialize the FTT
// ****************************************************************************
Result = ftt.reset();


// ****************************************************************************
// Notify FTT to monitor DS-101 Frame Layer traffic
// ****************************************************************************
Result = ds101.framelayer.monitor
         (
            port = "a"
        );


// ****************************************************************************
// Notify user to make the connection
// ****************************************************************************
Result = ftt.prompt.user
         (
            message = "Press button above DS-101 connection (be sure LED is on)"
         );


// ****************************************************************************
// Monitor DS-101 Frame Layer Traffic forever
// ****************************************************************************
Result = ftt.delay(time = 0xffffffff);
```

9 INDEX

APDU.Command.CLA 21
APDU.Command.Create 19, 20, 21
APDU.Command.Data 22
APDU.Command.INS 22
APDU.Command.Lc 22
APDU.Command.Le 23
APDU.Command.P1 22
APDU.Command.P2 22
APDU.Response.Create 21
APDU.Response.Data 23
APDU.Response.SW 23
APDU.Transmit 27
Authentication.Challenge.Get 28
Authentication.Commit 42
Authentication.External 28
Authentication.Mutual 41
Authentication.Mutual.Initialize 40
Card.HistoricalBytes 27
Card.PUPI 27
Card.Reset 27, 43
Card.State.Transition 29
Card.Traceability.Get 29
Data.CRC 23
Data.DES.CBC.Decrypt 25
Data.DES.CBC.Decrypt.Triple 25
Data.DES.CBC.Encrypt 24
Data.DES.CBC.Encrypt.Triple 25
Data.DES.ECB.Decrypt 24
Data.DES.ECB.Decrypt.Triple 24
Data.DES.ECB.Encrypt 24
Data.DES.ECB.Encrypt.Triple 24
Data.Get 29
Data.LRC 23
Data.Put 29
Data.Random.Get 25
Data.Receive 43
Data.Sign 28
Data.Transmit 43
DS100.Tag.Create 63, 64, 66, 67, 68, 70
DS101.Connect 59
DS101.Connect.Accept 89
DS101.ConnectPhysical 121

DS101.Disconnect 55

DS101.Disconnect.Accept 89

DS101.DisconnectPhysical 121

DS101.FCS.Calculate 55

DS101.FDU.Receive 55

DS101.FDU.Transmit 55

DS101.FrameLayer.Initialize 50

DS101.FrameLayer.Message.Available 51

DS101.FrameLayer.Monitor 121

DS101.FrameLayer.Receive 50

DS101.FrameLayer.Transmit 50

DS101.Initialize 52

DS102.Monitor 125

DS102.Receive 124

DS102.ReceivedBitCount 125

DS102.Transmit 123

FDU. CIKSplitRequest. CIKSplitData.Get 95

FDU. CIKSplitRequest.Fill.Create 71

FDU. CIKSplitRequest.Issue.Create 71

FDU. CommandRequest.CommandString.Get 73, 96

FDU. CommandRequest.Create 72, 96

FDU. DTDSplitRequest.Fill.Create 71

FDU. DTDSplitRequest.Fill.DTDSplitData.Get 95

FDU. DTDSplitRequest.Issue.Create 72

FDU. DTDSplitRequest.Issue.DTDSplitData.Get 95

FDU. ExtendedData.Fill.Create 73

FDU. ExtendedData.Fill.DUData.Get 97

FDU. ExtendedData.Fill.XDUControl.Get 97

FDU. ExtendedData.Issue.Create 73

FDU. ExtendedData.Issue.DUData.Get 97

FDU. FailureAcknowledgeWithData.Create 56

FDU. FailureAcknowledgeWithData.Data.Get 56

FDU. FailureAcknowledgeWithoutData.Create 56

FDU. FieldSpecifierRequest.Create 61

FDU. ProgramFileDataRequest.IsIssue 95

FDU. TrKEKRequest. IsIssue 96

FDU. TrKEKRequest.Fill.Create 72

FDU. TrKEKRequest.Fill.TrKEKData.Get 96

FDU. TrKEKRequest.Issue.Create 72

FDU.608.ECUStatusCheck.Create 88

FDU.608.ExtendedCommandRequest.CommandString.Get 99

FDU.608.ExtendedCommandRequest.ECID.Get 99

FDU.608.ExtendedCommandRequest.ECMD.Get 99

FDU.608.ExtendedCommandRequest.ECSIN.Get 99

FDU.608.ExtendedDataRequest.Fill.BulkErase.Create...........................82
FDU.608.ExtendedDataRequest.Fill.BulkErase.FileType.Get................108
FDU.608.ExtendedDataRequest.Fill.BulkErase.PathName.Get 108
FDU.608.ExtendedDataRequest.Fill.DataTransfer.Create......................79
FDU.608.ExtendedDataRequest.Fill.DataTransfer.Data.Get..................104
FDU.608.ExtendedDataRequest.Fill.DataTransfer.TransferControl.Get 104
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.Create...................80, 81
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.DataLength.Get ..106
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.DataType.Get......105
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.DestIdent.Get......106
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.FileName.Get.......107
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.FreeText.Get 107
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.PathName.Get.....107
FDU.608.ExtendedDataRequest.Fill.DirectoryEntry.SourceIdent.Get..106
FDU.608.ExtendedDataRequest.Fill.DirectoryRequest.Create..............79
FDU.608.ExtendedDataRequest.Fill.DirectoryRequest.DataType.Get ..105
FDU.608.ExtendedDataRequest.Fill.DirectoryRequest.PathName.Get .104
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferAbort.Create ...81
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferComplete.Create 81
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferInitiate.Create .79
FDU.608.ExtendedDataRequest.Fill.DirectoryTransferInitiate.EntryCount.Get............. 105
FDU.608.ExtendedDataRequest.Fill.FileRequest.Create.........................82
FDU.608.ExtendedDataRequest.Fill.FileRequest.FileName.Get............109
FDU.608.ExtendedDataRequest.Fill.FileRequest.PathName.Get 109
FDU.608.ExtendedDataRequest.Fill.SingleErase.Create.........................82
FDU.608.ExtendedDataRequest.Fill.SingleErase.FileName.Get............109
FDU.608.ExtendedDataRequest.Fill.SingleErase.PathName.Get 108
FDU.608.ExtendedDataRequest.Fill.TransferAbort.Create....................79
FDU.608.ExtendedDataRequest.Fill.TransferComplete.Create 79
FDU.608.ExtendedDataRequest.Fill.TransferComplete.FileCRC.Get...104
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.Create 78
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.DataLength.Get .102
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.DataType.Get.....102
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.FileName.Get 103
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.FreeText.Get 103
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.PathName.Get....103
FDU.608.ExtendedDataRequest.Fill.TransferInitiate.SourceIdent.Get.102
FDU.608.ExtendedDataRequest.Fill.UploadRequest.Create...................83
FDU.608.ExtendedDataRequest.Fill.UploadRequest.DataType.Get 110
FDU.608.ExtendedDataRequest.Fill.UploadRequest.PathName.Get.....110
FDU.608.ExtendedDataRequest.Fill.UploadRequest.StartName.Get 110
FDU.608.ExtendedDataRequest.Issue.BulkErase.Create........................87
FDU.608.ExtendedDataRequest.Issue.BulkErase.FileType.Get 118
FDU.608.ExtendedDataRequest.Issue.BulkErase.PathName.Get 117

FDU.608.ExtendedDataRequest.Issue.DataTransfer.Create...................84
FDU.608.ExtendedDataRequest.Issue.DataTransfer.Data.Get..............113
FDU.608.ExtendedDataRequest.Issue.DataTransfer.TransferControl.Get........................ 113
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.Create................86
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.DataLength.Get............................. 115
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.DataType.Get ...115
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.DestIdent.Get ...116
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.FileName.Get....117
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.FreeText.Get.....117
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.PathName.Get ..116
FDU.608.ExtendedDataRequest.Issue.DirectoryEntry.SourceIdent.Get 116
FDU.608.ExtendedDataRequest.Issue.DirectoryRequest.Create 85
FDU.608.ExtendedDataRequest.Issue.DirectoryRequest.DataType.Get 114
FDU.608.ExtendedDataRequest.Issue.DirectoryRequest.PathName.Get 114
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferAbort.Create 87, 117
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferComplete.Create 87
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferInitiate.Create 85
FDU.608.ExtendedDataRequest.Issue.DirectoryTransferInitiate.EntryCount.Get........... 115
FDU.608.ExtendedDataRequest.Issue.FileRequest.Create......................87
FDU.608.ExtendedDataRequest.Issue.FileRequest.FileName.Get..........119
FDU.608.ExtendedDataRequest.Issue.FileRequest.PathName.Get 119
FDU.608.ExtendedDataRequest.Issue.SingleErase.Create......................87
FDU.608.ExtendedDataRequest.Issue.SingleErase.FileName.Get..........118
FDU.608.ExtendedDataRequest.Issue.SingleErase.PathName.Get 118
FDU.608.ExtendedDataRequest.Issue.TransferAbort.Create.................85
FDU.608.ExtendedDataRequest.Issue.TransferComplete.Create...........85
FDU.608.ExtendedDataRequest.Issue.TransferComplete.FileCRC.Get114
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.Create 83, 84
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.DataLength.Get 111
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.DataType.Get ..111
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.DestIdent.Get ..112
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.FileName.Get ..112
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.FreeText.Get ...113
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.PathName.Get.112
FDU.608.ExtendedDataRequest.Issue.TransferInitiate.SourceIdent.Get........................... 111
FDU.608.ExtendedDataRequest.Issue.UploadRequest.Create................88
FDU.608.ExtendedDataRequest.Issue.UploadRequest.DataType.Get ...119
FDU.608.ExtendedDataRequest.Issue.UploadRequest.PathName.Get ..120
FDU.608.ExtendedDataRequest.Issue.UploadRequest.StartName.Get..120
FDU.608.ExtendedRequestCommand.Create...88
FDU.608.GetTimeRequest.Create..76
FDU.608.GetTimeResponse.Create..100
FDU.608.GetTimeResponse.Day_Of_Year..Get 76
FDU.608.GetTimeResponse.Hour.Get...76

FDU.608.GetTimeResponse.Hundredths.Get 77
FDU.608.GetTimeResponse.Minute.Get 76
FDU.608.GetTimeResponse.Second.Get 77
FDU.608.GetTimeResponse.Tenths.Get 77
FDU.608.GetTimeResponse.Year.Get 76
FDU.608.SetTimeRequest.Create 77
FDU.608.SetTimeRequest.Day.Get 100
FDU.608.SetTimeRequest.Hour.Get 100
FDU.608.SetTimeRequest.Hundredths.Get 101
FDU.608.SetTimeRequest.Minute.Get 101
FDU.608.SetTimeRequest.Second.Get 101
FDU.608.SetTimeRequest.Tenths.Get 101
FDU.608.SetTimeRequest.Year.Get 100
FDU.AXIDrequest.Create 60
FDU.AXIDrequest.FixedID.Get 90
FDU.AXIDrequest.FrameCount.Get 90
FDU.AXIDrequest.StationID.Get 90
FDU.AXIDresponse.Create 91
FDU.AXIDresponse.FixedID.Get 60
FDU.AXIDresponse.Framecount.Get 60
FDU.AXIDresponse.StationID.Get 60
FDU.DULength.Get 56
FDU.EndOfFile.Create 62
FDU.EndOfTransmissionRequest.Create 56
FDU.ExtendedData.Issue.XDUControl.Get 97
FDU.FCI.Get 55
FDU.FieldSpecifierRequest.Length.Get 91
FDU.FieldSpecifierRequest.StartByte.Get 91
FDU.FileHeaderRequest. Fill.TrKEK.Get 92
FDU.FileHeaderRequest. TrKEKCount.Get 92
FDU.FileHeaderRequest.Fill.Create 61
FDU.FileHeaderRequest.Fill.Name.Get 92
FDU.FileHeaderRequest.Fill.Path.Get 91
FDU.FileHeaderRequest.Issue.Create 61
FDU.FileHeaderRequest.Issue.Name.Get 93
FDU.FileHeaderRequest.Issue.Path.Get 92
FDU.FileHeaderRequest.Issue.TrKEK.Get 93
FDU.FileHeaderRequest.Issue.TrKEKCount.Get 93
FDU.GetTimeRequest.Create 73
FDU.GetTimeResponse. Hundredths.Get 75
FDU.GetTimeResponse.Create 98
FDU.GetTimeResponse.Day_Of_Year.Get 74
FDU.GetTimeResponse.Hour.Get 74
FDU.GetTimeResponse.Minute.Get 74

FDU.GetTimeResponse.Second.Get...74
FDU.GetTimeResponse.Tenths.Get 75
FDU.GetTimeResponse.Year.Get...73
FDU.KeyDataRequest.Create 62, 70
FDU.KeyDataRequest.DS100Tag.Get...94
FDU.KeyDataRequest.KeyData.Get 93
FDU.KeyDataRequest.KeyData.WithTag.Get 94
FDU.KeyDataRequest.Text.Get...94
FDU.ProgramFileDataRequest.Create 71
FDU.ProgramFileDataRequest.Data.Get 94
FDU.SetAddressRequest.Create..59
FDU.SetAddressRequest.DS101Address 89
FDU.SetIdentifierRequest.Create...59
FDU.SetIdentifierRequest.StationID.Get 90
FDU.SetTimeRequest.Create...75
FDU.SuccessAcknowledgeWithoutData.Create.......................................56
FDU.TrKEKRequest.IssueTrKEKData.Get...96
File.Binary.Read 32
File.Binary.Update...32
File.Block 31
File.Purse. Debit.Reverse...36
File.Purse.Credit 36
File.Purse.Debit...36
File.Record.Read...33, 34
File.Record.Read.First..34
File.Record.Read.Last 34
File.Record.Read.Next...34
File.Record.Read.Previous..34
File.Record.Update 35
File.Record.Update.First...35
File.Record.Update.Last..35
File.Record.Update.Next 35
File.Record.Update.Previous 35
File.Select...30
File.Select.DF...30
File.Select.EF 30
File.Select.MF 30
File.Select.Parent.DF 30
File.Select.Parent.EF..31
File.Unblock..31
FTT.BinaryFile.Read...44
FTT.BinaryFile.Write...44
FTT.ComputerName.Get...44
FTT.ConvertToByte 45

FTT.ConvertToDword 45

FTT.ConvertToWord 45

FTT.CurrentTime.DayOfYear.Get 46

FTT.CurrentTime.Hours.Get 45

FTT.CurrentTime.Hundredths.Get 46

FTT.CurrentTime.Minutes.Get 45

FTT.CurrentTime.Seconds.Get 45

FTT.CurrentTime.Tenths.Get 45

FTT.CurrentTime.Year.Get 46

FTT.Delay 46

ftt.DTDCompatible.FDUNeedsResponse.Add 57

ftt.DTDCompatible.FDUNeedsResponse.Clear 57

FTT.EnvironmentVariable.Get 46

FTT.HexToString 49

FTT.Prompt.User 46

FTT.Prompt.User.Choices 47

FTT.Reset 44

FTT.SerialNumber.Get 49

FTT.StringToHex 48

FTT.TextFile.Read 47

FTT.TextFile.Write 47

FTT.Timed.Prompt.User 48

FTT.Timed.Prompt.User.Choices 48

FTT.UserName.Get 49

FTT.Version.Get 49

FTT.Zeroize 49

ISO7816.T1.Block.Create 26

Key.Update 39

Key.Update.Initialize 38

Memory.Test.Full 37

Memory.Test.Light 37

Memory.Test.Medium 37

Memory.Test.RAM 37

NVMemory.Check.CRC 37

NVMemory.Check.LRC 38

OS.Execute 46