



Script Reference

*Digita™ Operating Environment
version 1.5*

Part Number 6-2120-0001-01

Version 1.5
December 7, 1999



© 1998 FlashPoint Technology, Inc. All Rights Reserved.

Digita, Digita Desktop and the Digita logo are trademarks or registered trademarks of FlashPoint Technology, Inc. in the U.S. and other countries. All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

All information contained herein is the property of FlashPoint Technology, Inc., or its licensors, and is protected by copyright law, international treaties, trade secrets, and may be covered by U.S. and foreign patents. Any reproduction or dissemination of any portion of this document, of the software, or other works derived from it is strictly forbidden unless prior written permission is obtained from FlashPoint Technology, Inc.

The information contained herein is subject to change and may change without notice.

Table of Contents

About This Book xiii

1 Script Grammar 1

Script Command Conventions	1
Statements	1
Identifiers	1
Script Language File Names	2
Variable Types	2
Operators	3
Data Type Assumptions	3
Script Command Format	4
Assignment Statements	5
Command Statements	6
Comment Statements	7
Conditional Statements	8
Declarative Statements	9
Exit Statements	9
Marker Statements	9
mylabel:	9
Goto Statements	9
Label Statements	10
Menu Statements	10
Mode Statements	10
Name Statements	11
Error Codes	11

2 Script Command Reference 13

Script Command Summary 13

Script Commands 16

- Alert 16
- Display 16
- DisplayClear 17
- DisplayLine 17
- EndCapture 18
- EraseFile 18
- FileClose 19
- FileOpen 19
- FindString 20
- GetCameraDefault 20
- GetCameraState 21
- GetCameraStatus 22
- GetCapabilitiesCount 26
- GetCapabilitiesListItem 26
- GetCapabilitiesRange 27
- GetCapabilitiesValue 27
- GetCapabilityType 28
- GetClock 28
- GetDateString 29
- GetError 29
- GetFileCount 30
- GetFileInfo 31
- GetFileTag 32
- GetImageSpecifications 32
- GetMarkedImage 34
- GetMarkedImageCount 35
- GetNewFileCount 35
- GetNewFileInfo 36
- GetOption 37
- GetPowerMode 37
- GetProductInfo 38
- GetScriptName 38
- GetStorageStatus 39
- GetString 40
- GetSystemFileCount 40
- GetSystemFileName 41
- GetTimeString 41
- MakeFolder 42
- MarkImage 42
- MarkAllImages 42
- NumberToString 43
- Read 43
- ReadLine 44
- RestoreCameraDefault 44
- RunApp 45
- RunScript 45

SerialClose	45
SerialOpen	46
SerialReceive	46
SerialSend	46
SerialSendReceive	47
SetCameraDefault	48
SetCameraState	49
SetCaptureMode	49
SetClock	50
SetDelimiter	50
SetOption	51
SetPowerMode	51
SetTitle	52
SetUserFileTag	52
StartCapture	52
SubString	54
StringToNumber	55
UnMarkImage	56
UnMarkAllImages	56
Wait	56
WaitForShutter	57
Write	57
WriteLine	58

A Product Information Parameters 59

cacn – Application Copyright Notice	59
carv – Control Application Version	59
cccn – CCS Copyright Notice	60
ccsv – CCS Subsystem Version	60
fwcn – Digita Copyright Notice	60
fwrv – Digita Firmware Version	61
hprs – Host PRAM Size	61
hwrv – Hardware Version	61
ipcn – IPM Copyright Notice	62
iprn – IPM Subsystem Version	62
pcnt – Picture Count Taken	62
ptid – Product Name	63
rams – Application RAM Size	63
scnt – Strobe Pictures Taken	64
sern – Camera Serial Number	64
ucrv – Universal Command Set Version	64
vdiid – Vendor Name	65

B Camera Capabilities Parameters 67

aagc – Analog Signal Gain Control	67
acpd – AC Power Down Timeout	68
acse – AC Sleep Timeout Enable	68
actc – AC Sleep Timeout	68
aelk – AE Lock Mode	69
aflk – AF Lock Mode	69
aper – Aperture (F Number)	70
bccl – Burst Capture Color Mode	70
bcpn – Burst Capture Compression	71
bsiz – Burst Capture Size	71
btpd – Battery Power Down Timeout	72
btse – Battery Sleep Timeout Enable	72
bttc – Battery Sleep Timeout	72
camf – Camera Folder Name	73
cmne – User-defined Camera Name	73
cpnt – User Copyright Notice	74
cpus – Processor Speed	74
cput – Processor Type	74
crst – Image Counter Mode	75
deft – Short Date Format	75
dfmt – Date Display Format	75
drdn – User RAMDisk Name	76
ebrs – Enable Built-in Scripts	76

exsa – Expansion Slots Available	76
fbti – Burst Time Interval	77
fdst – Focus Distance	77
flty – File Type	78
fmod – Focus Mode	78
fmod – Focus Method	79
fwrd – Digital Release Date	79
hint – Hint Mode	80
icts – Image Capture Tone	80
ihtc – Image Head Timeout	81
iirv – Image Info Version	81
irev – Instant Review Duration	82
isam – Image Sharpness Amount	82
isra – Image Sharpness Radius	83
isth – Image Sharpness Threshold	83
mcap – Media Type Capture Mode	83
mcro – Macro Mode	84
mhbs – Maximum Host Buffer Size	84
mtdy – Self Timer Delay	84
oerv – Digital OE Version	85
pasw – User Password	85
pbbr – Burst Playback Interval	86
pbgr – Group Playback Interval	86
pbrt – Print Brightness	86
pbtr – Timelapse Playback Interval	87
pchc – Patch Count	87
pcon – Print Contrast	87
pcpy – Print Copy Count	88
pgms – Program Shift	88
play – Print Layout	88
port – Print Paper Orientation	89
posh – Print Horizontal Shift	89
posv – Print Vertical Shift	89
pqly – Print Quality	90
pres – Print Resize	90
psam – Print Sharpness	90
psat – Print Saturation	91
psel – Printer Selection	91
psiz – Print Paper Size	91
psra – Print Sharpness Radius	92
psth – Print Sharpness Threshold	92
ptyp – Print Paper Type	92
qbti – Quarter Burst Time Interval	93

rdsz – RAM Disk Size	93
rgnc – Region Code	93
rmod – Auto Rotate Image	94
saim – Script Activity Mask	94
sccl – Still Capture Color Mode	94
scmp – Strobe Compensation	95
scpn – Still Capture Compression	95
shut – Shutter Speed	96
smod – Strobe Mode	96
spre – Strobe Precount	97
sscs – Slideshow Category String	97
ssct – Slideshow Content	97
ssdr – Slide Show Frame Interval	98
ssis – Slide Show Image Sound	98
ssiz – Still Capture Size	99
sslp – Slide Show Loop Indicator	99
ssnd – System Sound Selector	100
ssvl – System Sound Volume	100
stim – Self Timer Enable	101
tccl – Timelapse Capture Color Mode	101
tcnt – Timelapse Capture Count	101
tcpn – Timelapse Capture Compression	102
tfmt – Time Format	102
tldy – Timelapse Capture Interval	103
tsiz – Timelapse Capture Size	103
ucpt – User Copyright Notice	104
vmod – Video Output Mode	104
wbcb – White Balance Blue	105
wbcg – White Balance Green	105
wbcr – White Balance Red	106
wena – Watermark Enable	106
wmod – White Balance Mode	107
wpic – Watermark Logo File Name	107
wpop – Watermark Logo Operation Mode	108
wpxp – Watermark Logo X Position	108
wpyp – Watermark Logo Y Position	109
wsbc – Watermark Text Background Color	109
wsfc – Watermark Text Foreground Color	110
wsop – Watermark Text Operation Mode	110
wstr– Watermark Text String	111
wsts– Watermark Text Font Size	111
wsexp – Watermark Text X Position	112
wsyp – Watermark Text Y Position	112

wdbc – Watermark Time Background Color	113
wdfc – Watermark Time Foreground Color	113
wtop – Watermark Date Background Mode	114
wtds – Watermark Date Font Size	114
wtxp – Watermark Time X Position	115
wtyp – Watermark Time Y Position	115
xcmp – Exposure Compensation	116
xmod – Exposure Mode	116
xmtd – Exposure Method	117
zmod – Zoom Mode	117
zpos – Zoom Position	118
zspd – Zoom Speed	118

C Image File Tags 119

aagc – Analog Gain Value	119
aper – Aperture (F Number)	119
bsfd – Image Group Folder Name	120
ccsv – Camera Firmware Version	120
clos – Closest Subject Distance	120
cmpn – Image Compression Level	121
cmrc – Camera Region Code	121
cmtc – Image Compression Type	121
cngn – Image Natural Group Name	122
cpgn – Image Group Folder Name	122
date – Image Capture Date	122
eval – Exposure Value	123
exht – Extracted Image Height	123
exwd – Extracted Image Width	123
fars – Farthest Subject Distance	124
fdst – Focus Distance	124
flty – Image File Type	124
fmod – Focus Mode	125
fntd – Focus Method	125
fnum – F Number	125
fwrv – Digita Firmware Version	126
grct – Image Group Sequence Number	126
grd1-4 – Image Group Data 1-4	126
grfd – Capture Group Folder Name	127
grtg – Image Group Tag	127
hint – Image Hint Mode	127
hwrv – Hardware Version	128
icts – Image Capture Tone	128
iirv – Image Info Version	128

imcn – Image File Type	129
imfg – Image Flag	129
imht – Image Height	129
imis – Image Info Size	130
imiv – Image Info Version	130
impn – Image Color Depth	130
imwd – Image Width	131
isam – Image Sharpness Amount	131
isra – Image Sharpness Radius	131
isth – Image Sharpness Threshold	132
lkmd – AE/AF Lock Mode	132
mcap – Media Type Capture Mode	132
ortn – Capture Orientation	133
pcnt – Picture Taken Count	133
pgms – Program Shift	133
ptid – Product Name	134
rgnc – Region Code	134
rmod – Auto Image Rotation Mode	134
scmp – Strobe Compensation	135
scom – Sound Compression Type	135
sdsr – Sound Sampling Rate	135
sdss – Sound Sampling Size	136
sern – Serial Number	136
shpn – Sharpening Level	136
shut – Shutter Speed	137
smod – Strobe Mode	137
spre – Strobe Precount	137
time – Image Capture Time	138
ucnr – User Caption Region Code	138
ucnt – User Caption Text	138
ucpt – User Copyright Notice	139
ucrc – User Copyright Region Code	139
uc01-uc16 – User Categories 01-16	139
udt1-udt8 – User Data 1-8	140
uptd – User Protected	140
ur01-ur16 – User Categories 01-16 Region Code	140
usr1-usr4 – User Strings 1-4 Region Code	141
ust1-ust4 – User Strings 1-4	141
utag – User Tag	141
vdid – Vendor ID	142
vd01-vd04 – Vendor Data 1-4	142
wbcb – White Balance Blue	142
wbcg – White Balance Green	143

wbcr – White Balance Red	143
wmod – White Balance Mode	143
wmrc – Watermark Region Code	144
wstr – Watermark Text String	144
xcmp – Exposure Compensation	144
xmod – Exposure Mode	145
xmtd – Exposure Method	145
zmod – Zoom Mode	145
zpos – Zoom Position	146

D Date and Time Formats 147

E Bitmask Table 149

F Japanese Characters 151

Index 157

About This Book

Purpose

This manual provides a complete reference for the Digita™ Script language.

Intended Audience

This manual is intended for users who have some prior experience with a scripting or programming language. It does not teach basic programming techniques.

Related Manuals

The *Digita Script Guide* provides an introduction to the Digita Script language.

Text Conventions

code	sample code
<i>italics</i>	references to other manuals

This chapter provides a detailed description of the Digita Script Language grammar and conventions. Script items with Digita OS version requirements have been noted. Refer to the *Digita Script Guide* for an overview and introduction to scripts.

Script Command Conventions

This section describes the command conventions for the Digita Script command set.

Statements

Statements in the Digita Script language must conform to the following guidelines:

- Statements can start at any position of a line.
- End of line is specified with a linefeed, carriage return, or a combination of linefeed and carriage return.
- Single-line comments start with a #, until reaching an end of line.
- Multi-line comments start with /* and end with */.(Digita Script V 1.5 compatible only)
- Single statements are always on one line, and continuation statements are not supported.
- Elements in a statement are separated with blanks or horizontal tabs.

Identifiers

Digita Script language identifiers are a series of characters that specify a value or other language element. The following rules apply to identifiers in the Digita Script language.

- The first character of an identifier must be an upper or lower case letter of the alphabet, either A through Z or a through z.
- Upper and lower case letters are not interpreted as the same. For example, you could not use eGad and EGad interchangeably in a script, since they would be recognized as two distinctly different identifiers.
- Identifiers may be any length, but will be unique only up to the first 31 bytes/characters.

Script Language File Names

Script file naming conventions are as follows:

<code><name>.CSM</code>	Digita Script file names conform to the DOS file format (8.3). Names can include letters, digits, or any of the following special characters: \$ % ' - _ @ { } ~ ' ! # ()
<code><name></code>	A string of up to eight alphanumeric characters. Use upper case characters. The first character cannot be a number.
<code>.CSM</code>	indicates that the script is a general “camera script module” loaded by the camera application.

An example script file name is MYTEST_1.CSM

Variable Types

You can declare a data type for your script data elements by using the declare keyword and one of the data type specifiers listed in Table 1. The following example declares uStatus as a u type variable.

Example:

```
declare u :uStatus
```

Table 1. Constant Types

Data Type	Description
u	An integer value between 0 and $(2^{32}-1)$ (~4G). Hexadecimal values begin with “0x” or “0X”.
i	A signed integer between -2^{31} and $(2^{31}-1)$. Hexadecimal values begin with “0x” or “0X”.
f	A fixed point decimal number between -32768.0000 and +32767.9999.
s	A sequence of characters (maximum length=31) surrounded by double quotes (e.g., “abcde12345”). Double quotes may be used within the sequence if first preceded by the backslash character ‘\’ (e.g., “abc\”123”). To use a backslash character within a string, precede it with an initial backslash (e.g., “abc\\123”).
t	Same as, and interchangeable with, the s data type, but with a maximum string length of 255 characters. ^a
n	A DOS file name surrounded by double quotes. Maximum file name length of eight characters followed by “.” and a 3-character extension, for example, “TEST_01.CSM”
b	An unsigned 32-bit binary integer where each bit can either be true(1) or false(0). Can be expressed as a boolean (0b), hexadecimal (0X or 0x) or decimal value. Example: By converting the hex value “0xFF03” to its boolean value “0b1111111100000011”, it can be seen that the first eight bits are true, the next six bits are false, and the last two bits true.

a. The t data type can only be used in scripts for products that support Digita Script V1.5 or greater.

Operators

The operators supported by the Digita Script language are listed in Table 2.

Table 2. Digita Script Language Operators

Operator Type	Valid Data Types	Definition
arithmetic	u, i, f, b	+ (plus), - (minus), * (multiply by), / (divide by) Multiplication and division have precedence over addition and subtraction.
modular arithmetic	u, i	% (modulus) ^a
relational	u, i, f, b (s, t, n for == and != only)	> (greater than), >= (greater than or equal to), < (less than), <= (less than or equal to), == (equal to), != (not equal to)
bitwise	u, i, b	& (AND), (inclusive OR), ^ (exclusive OR) and one special operator, ~ (one's complement). The ~ always appears in front of the first operand of the bitwise statement or expression. ~ has precedence over &, and ^
assignment	u, i, f, b, s, t ^b , n	= -Sets the value of the item on the left of the "=" sign to the value of the item on the right

- The modulus operator (%) can only be used in scripts for products that support Digita Script V1.5 or greater.
- The t data type can only be used in scripts for products that support Digita Script V1.5 or greater.

Data Type Assumptions

The following assumptions are made about data types when scripts are interpreted.

- 5 to 12 character strings ending with a period and 3 character extension are assumed to be type n unless in string assignment statements.
 - Uninitialized variable data types u, i, f, and b are assumed to have a value of 0.
 - Uninitialized variable data types s, t and n are assumed to be NULL strings.
 - "n" type data will not be validated to ensure conformance to the DOS 8.3 format during assignment.
- "f" type data may not be stored as accurately as it was specified in the script program because of truncation (15-bit integer and 16-bit fraction). The maximum number of decimal places stored within this type will be four.

Script Command Format

The format used to describe script commands in this document is as follows:

CommandName (input-parameters, output-parameters)

- All data is sent 32-bit word aligned. That is, 32-bit data is always aligned on 32-bit word boundaries, and 16-bit data is always aligned within a 32-bit data field.
- All BitFlag tables in this document list the bits in order, with the most significant bit first.

The following statements are supported by the Digita Script language:

```
assignment statements
command statements
comment statements
conditional statements
declarative statements
exit statements
marker statements
goto statements
label statements
menu statements
mode statements
name statements
```

Syntax Notes:

```
[.....] the contents are optional
|       means OR
```

Assignment Statements

The syntax for assignment statements is shown here.

```
identifier = operation
```

operation The operation can be a string operation, an arithmetic operation, or a bitwise operation. These are described in more detail below.

STRING OPERATION

The syntax for string assignment statements is shown here:

```
identifier = identifier | string-literal
```

Examples:

```
str1 = "xmod"
```

```
str2 = str3
```

ARITHMETIC OPERATION

Arithmetic operations apply to u, i, and f data types only. The syntax for arithmetic assignment statements is shown here.

```
identifier = [-]identifier|constant [arithmetic-operator identifier|constant]
```

Example:

```
num1 = b + c
```

MODULUS OPERATOR – Digita Script V1.5 or greater only

Modular arithmetic operations apply to the u and i data types only.

Example:

```
uvar = 5
```

```
ivar = 2
```

```
uvar =uvar % ivar
```

MULTIPLE LEVELS OF PARENTHESIS – Digita Script V1.5 or greater only

Equations can be enclosed within parenthesis.

Examples:

```
uvar = (ivar - 23) / (uvar + 5)
```

```
uvar = uvar * (fvar * (25 * (23.5 - 5.1)))
```

```
SetOption (1, "Which one?", bvar || (bvar2 & bvar3))
```

BITWISE OPERATION

Bitwise assignment statements apply to u, i, and b data types only. The syntax for bitwise assignment statements is shown here.

```
identifier = [~]identifier|constant [bitwise-operator identifier|constant]
```

Example:

```
bit1 = ~bit2 & bit3
```

RANDOM NUMBERS – Digita Script V1.5 or greater only

Applies to u and i data types only. Generates values from 0 to 32,767.

Example:

```
declare i: ivar  
Random (ivar)
```

Command Statements

Command statements set up and execute the Digita Script language commands. The syntax of the command statement is as follows:

```
command-name (parameter-list)
```

command-name script language command name

parameter-list parameters defined for this command

Here is a command statement example:

```
GetCameraState ("wmod", wbal)
```

The following guidelines are used to handle the input and output parameters:

- There are only checks on the accuracy of the input parameter data type as defined in the script language reference. No type checking is done on returned (output) data.
- The variable parameter types must be the same as defined in the script language functions. No type conversion is performed for sending data.
- No data type checking is done if the sent parameters are beyond the command parameters' definition; the data is just sent out based on whatever is defined in the script program.
- If a parameter is a variable, the data type must be the same as the required command parameter type; no type conversion is performed.
- Data may be truncated if the storage area for the result variable is less than the actual data being returned by the command (e.g., data of type string being stored in a variable of type i (signed int)).

Comment Statements

The comment statement allows the script writer to add explanatory comments to the script. Generally, it is a good idea to use comments throughout a script, especially if the script is complex and subject to change. The format is shown here:

```
# comment-text
```

Example:

```
# Scripts are easier to maintain when the writer adds comments.
```

End of line Comments – Digita Script V1.5 or greater only.

Executable code can now be followed on the same line by ‘#’ style comments.

Example:

```
if count < 85 # check count
count = count + 1 # increment
goto LOOP # LOOP again
end
```

Multiple line format – Digita Script V1.5 or greater only.

Multiple line comments are "C-style" comments, i.e. an initial /* followed by */. The standard format for multi-line comments is as follows:

```
/*comment text*/
```

Example:

```
/*Scripts are easier
to maintain when the
writer adds comments.*/
```

Single and multi-line comment styles can be intermixed. There are, however, some cases in which only one comment style is recognized.

Some permissible forms are:

```
/*
DisplayLine ("one") # */
```

Note here that */ is detected even though it follows #. The first /* renders # to a regular character.

```
# DisplayLine ("two") /*
# DisplayLine ("two") */
```

In this case, /* and */ aren't detected because of #.

```
/ *
DisplayLine ("help")
* /
```

```
DisplayLine ("me") / * on the same line * /
```

```
/ *
DisplayLine ("hello")
# enclosed within comments
* /
```

Note Nested comments are not supported.

Conditional Statements

The syntax for conditional statements is as follows:

```
if conditional-expression
    arithmetic-statements | bitwise-statement | string-statement |
    jump-statement | function-statements | UI-statement
end
```

where

```
conditional-expression    [~] identifier | constant
                           [relational-operator | bitwise-operator
                           identifier | constant]
```

All relational operators apply to u, i, f and b data types.

Only the relational operators “==” and “!=” apply to s, t and n data types.

Bitwise operators apply to u, i, and b data types.

For more information on operators, refer to Table 2 on page 3.

Table 3 shows the supported data types and operators that apply to each.

Table 3. Data types and associated operators

Data type	Applicable operators
u	>, >=, <, <=, ==, !=, ~, &, , ^, % ^a
i	>, >=, <, <=, ==, !=, ~, &, , ^, % ^a
f	>, >=, <, <=, ==, !=
s	==, !=
t	==, !=
n	==, !=
b	>, >=, <, <=, ==, !=, ~, &, , ^

a. The modulus operator (%) can only be used in scripts for products that support Digita Script V1.5 or greater.

Here are some conditional statement examples:

```
if a > 100
    a=100
end

if str1 == "xmod"
    blue = 0.345
end

if ~b & c
    green = 0.567
end
```

Declarative Statements

The declarative statement declares the data type for parameters and other elements used in the Digita Script language. The syntax of declarative statements is as follows:

```
declare declaration-specifier:variable-list

declaration-specifier    u | i | f | s | t | n | b
variable-list           variable, variable, .....
```

Here are some declarative statement examples:

```
declare    u: num1, num2
declare    s: str1, str2
```

Declarations may be defined anywhere within a script, but since this is an interpreted language, variables must be defined before use.

Exit Statements

Exit statements cause the script program to terminate. The syntax of the exit statement is as follows:

```
exitscript
```

Exit statements can be placed anywhere within a script. If the end of the script is reached without an explicit `exitscript` statement, an implicit `exitscript` is executed.

Marker Statements

Markers create a reference by identifying a specific location in your script. Markers can be used with `goto` statements to create loops. The syntax of a marker is shown here:

```
marker :
```

Here is a label marker definition example:

```
mylabel:
```

Goto Statements

`Goto` statements move script execution from the current line to the next statement on the line identified by a marker. The marker can be a forward or backward reference. The syntax of `goto` statements is as follows:

```
goto marker
```

marker: label identifying the program location to which the script execution should jump.

Here is a `goto` statement example:

```
goto mylabel
```

The label search begins at the current line and progresses forward in the script. If the requested label is not encountered, the script is rewound and the search continues from the beginning of the script.

Label Statements

The label statement defines the script name that appears in the menu. The statement format is shown here:

```
label "string"
```

string any character string; total length must be less than or equal to 31 characters.

Here is a label statement example:

```
label "Set White Balance Mode"
```

In the event that multiple label statements are encountered, the first valid label statement that conforms to the syntax defined above will be used to name the script within the specified menu.

Menu Statements

The menu to which this script belongs. The format is shown here:

```
menu "string"
```

string any character string; total length must be less than or equal to 31 characters.

If the menu does not currently exist, it is created, and a default icon is assigned to this menu. Scripts can always be added to existing menus, and different scripts can be added to the same menu.

Here is a menu statement example:

```
menu "Photo Scripts"
```

In the event that multiple menu statements are encountered, the first valid menu statement that conforms to the syntax defined above will be used to place the script within the specified mode/menu.

Mode Statements

The mode statement defines the mode for this script. The name of the script will be displayed on its designated menu only when the Digita enabled device is in this mode. Currently, the supported values are 0 (capture), 1 (review), and 2 (play). The format is shown here:

```
mode n
```

n numerical value of the mode for this script. The meaning of different mode values is established by the camera manufacturer.

Here is a mode statement example:

```
mode 0
```

In the event that multiple mode statements are encountered, the first valid mode statement that conforms to the syntax defined above will be used to place the script within the specified mode/menu.

Name Statements

The name statement is a 31 character description of the script. The format is shown here:

```
name "string"
```

string any character string; total length must be less than or equal to 31 characters.

Here is a name statement example:

```
name "sample script name"
```

In the event that multiple name statements are encountered, the first valid name statement that conforms to the syntax defined above will be used to name the script.

Note The mode, menu, label, and name statements are parsed during the boot sequence of the Digita™ operating environment. It is recommended that these statements be defined as early as possible within the script to avoid impacting the performance of the boot sequence.

Error Codes

This section describes the error codes that may be passed back in response to a command. The error codes shown in Table 4 are common to all script commands.

Table 4. Error Codes

Error Code	Explanation
0	No error. The operation was successful.
1	Illegal command or command not supported.
2	Protocol error.
3	Camera application failed to respond in time.
4	Memory errors, flash read/write errors, bad file, image corrupted, or operating system errors.
5	Illegal parameter value, too many parameters, too few parameters, or bad format of parameter.
6	The specified file system has insufficient free space to complete the requested operation.
7	Specified file does not exist.
8	The image does not contain the section requested (e.g., thumbnail).
9	The file specified is an invalid or unknown type.
10	The drive specified is unknown to the system.
11	The drive specified is not mounted or contains no media.
12	The system is currently busy and cannot process the command.
13	Battery low.
negative number	Illegal script grammar or operation.

Script Command Reference

This chapter describes the Digita Script command set in detail. Refer to the *Digita Script Guide* for an introduction to scripts and examples of scripts.

Note Refer to “Script Command Format” on page 4 for details on the syntax conventions used in this document.

Script Command Summary

The following is a summary of the script command set provided by the Digita operating environment. The table groups the commands by function.

Table 5. Script Commands Grouped by Purpose

Command	Description	Location
<i>Product and Image Information Commands</i>		
GetProductInfo	Requests specific information about the product by parameter.	page 38
GetImageSpecifications	Returns hardware-related image information, including zone organization.	page 32
<i>Status Commands</i>		
GetCameraStatus	Returns system status, capture status, and vendor status.	page 22
GetError	Clears error flag and returns last error and its description.	page 29
<i>Option List Commands</i>		
SetOption	Displays an option list to the user.	page 51
GetOption	Prompts the user for an option selection.	page 37
<i>Camera Capabilities And State Commands</i>		
GetCapabilityType	Returns capability type associated with the specified camera parameter.	page 28
GetCapabilitiesRange	Returns the min-max range information associated with the specified camera parameter.	page 27
GetCapabilitiesCount	Returns the number of capability list items defined for the specified camera parameter.	page 26
GetCapabilitiesListItem	Returns a name-value pair for the specified camera parameter's list index.	page 26
GetCapabilitiesValue	Returns the capability value defined for the specified camera parameter.	page 27
GetCameraState	Requests a current camera parameter setting.	page 21
SetCameraState	Updates a current camera parameter setting.	page 49
GetCameraDefault	Requests the user or factory default value of a camera parameter setting.	page 20
SetCameraDefault	Updates the user default value for a camera parameter setting.	page 48

Table 5. Script Commands Grouped by Purpose

Command	Description	Location
RestoreCameraDefault	Restores either the user defaults or the factory defaults.	page 44
<i>Power and Capture Commands</i>		
GetPowerMode	Determines the power level available to the camera.	page 37
SetPowerMode	Powers off the camera.	page 51
SetCaptureMode	Controls the type of capture sequence: still, group, or timelapse.	page 49
StartCapture	Starts the capture process.	page 52
EndCapture	Stops the capture process.	page 18
<i>File Commands</i>		
GetFileCount	Returns the total number of captured images resident on the disk.	page 30
GetFileInfo	Accesses the list of all captured image files currently available.	page 31
GetNewFileCount	Returns the number of new images resident on the disk.	page 35
GetNewFileInfo	Accesses the list of all newly created image files currently available.	page 36
EraseFile	Delete image files or other files.	page 18
GetStorageStatus	Determines the status of storage available to the camera.	page 39
GetFileTag	Returns the value of the specified file tag.	page 32
SetUserFileTag	Sets the value of the specified user tag.	page 52
MakeFolder	Creates a new directory	page 42
<i>Date and Time Commands</i>		
GetClock	Gets the current date and time.	page 28
SetClock	Sets the current clock value for date and time.	page 50
GetString	Returns the date as a formatted string.	page 29
GetTimeString	Returns the time as a formatted string.	page 41
<i>Camera Script Execution Commands</i>		
GetSystemFileCount	Returns the number of system files resident in the SYSTEM folder.	page 40
GetSystemFileName	Returns a single system file name based on an index.	page 41
RunApp	Restarts the camera using the selected application.	page 45
RunScript	Runs a script file that is listed in the SystemFilesList.	page 45
GetScriptName	Retrieves the long name of a script.	page 38
<i>Serial Processing Commands</i>		
SerialOpen	Opens the serial channel at the selected baud rate.	page 46
SerialSendReceive	Sets up for the next data exchange through the serial port.	page 47
SerialSend	Sends the indicated data and sets up the next receive cycle.	page 46
SerialReceive	Waits to receive an expected transmission.	page 46
SerialClose	Releases the serial port.	page 45

Table 5. Script Commands Grouped by Purpose

Command	Description	Location
<i>File and Read/Write Commands</i>		
FileOpen	Opens a file in the specified directory with the specified filename.	page 19
FileClose	Closes a file.	page 19
Read	Reads data from a text file.	page 43
ReadLine	Reads one line of data from a text file.	page 44
Set Delimiter	Sets the delimiter used by the Read and ReadLine commands.	page 50
Write	Appends the specified data to the current pointer location in an open file (no carriage return added at end).	page 57
WriteLine	Appends the specified data to the current pointer location in an open file and adds a carriage return to the end of the data.	page 58
<i>String Manipulation Commands</i>		
FindString	Places the pointer at the start of a substring within a source string.	page 20
NumberToString	Converts a numerical value to a string value.	page 43
StringToNumber	Converts a string to a numerical value.	page 55
Substring	Extracts and returns a substring from a source string.	page 54
<i>Wait And Display Commands</i>		
Wait	Specifies the number of milliseconds for the script program to wait.	page 56
Display	Places feedback text in the LCD (liquid crystal display).	page 16
DisplayLine	Places feedback text in the LCD and appends a carriage return.	page 17
DisplayClear	Clears the LCD.	page 17
GetString	Requests an input text string from the user.	page 40
WaitForShutter	Returns camera system control to the user while waiting for the shutter button to be pressed. Valid only in "capture" mode.	page 57
Alert	Places an alert on the LCD.	page 16
<i>Image Commands</i>		
GetMarkedImageCount	Returns the number of images currently marked.	page 35
GetMarkedImage	Retrieves an image file.	page 34
MarkImage	Marks a specified image.	page 42
MarkAllImages	Marks all the images on the CompactFlash card.	page 42
UnMarkImage	Unmarks a specified image.	page 56
UnMarkAllImages	Unmarks all the images on the CompactFlash card.	page 56

Script Commands

This section provides details of the Digita Script commands. The commands are listed in alphabetical order.

Each command is described in terms of syntax and parameter definitions. In cases where parameter descriptions apply only to one command, they are included in this chapter with the command description. However, many parameter descriptions apply to several commands. To make this information easy for you to access, these descriptions are tabulated in Appendix A, "Product Information Parameters" and Appendix B, "Camera Capabilities Parameters".

All commands described below return a status indicating success or failure of the command execution. If the command fails and data is expected to be returned, there is no guarantee that the data fields represent valid values. It is recommended that all scripts issuing commands should check the return status before continuing execution.

The list of possible command results may be found in Table 4 on page 11.

Note In the syntax descriptions for the commands, inputs are underlined.

Alert

This command allows the script to place an alert on the LCD display. The alert command displays the prompt message to the user via a full screen display and appends the softkey prompts to "Continue" and "Exit". The user selects to either "Continue" or "Exit" the script via the softkeys.

SYNTAX

Alert (Prompt)

PARAMETER DEFINITION

Parameter Name	Type	Description
Prompt	s	The text to be placed in the alert on the display. The data can include one or more strings and variables, separated by commas.

EXAMPLE

In this example, 'count' and 'total' are previously defined variables:

```
Alert ("Processed ", count, " out of ", total, " images.")
```

Although string variables are limited to 31 characters, there is no limitation on the prompt string length when formatted as displayed in the example. All text and variable data will be appended and displayed within the alert dialog. If the text is too long to fit on a single line, the text will be wrapped based on word boundaries.

Display

This command allows the script to place feedback text in the LCD (liquid crystal display) during long operations. For simple scripts, this is not necessary, since the elapsed execution time is so short. However, if a long process is required in a script, it is a good idea to give the user some feedback on progress.

The text display dialog is a scrolling display in which each line of text is scrolled up the display as the display is filled. Depending on font styles and sizes, up to five lines of text may be displayed at a time before the scrolling of the text begins to accommodate additional text.

SYNTAX

```
Display (DisplayString)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
DisplayString	s	The feedback information to be displayed on the LCD. The data can include one or more strings and variables, separated by commas.

EXAMPLE

```
declare u:count, total
Display ("Processing image ", count, " of ", total, ".")
```

Although string variables are limited to 31 characters, there is no limitation on the display string length when formatted as displayed in the example. All text and variable data will be appended and displayed within the display dialog. If the text is too long to fit on a single line, the text will be wrapped based on word boundaries.

Note Fixed variables are not displayed as nn.nn, but as signed long data that is not useful for most purposes. To convert this information to usable information, divide it by 65,536 to get the actual number.

DisplayClear

This command clears any script text messages currently displayed on the LCD.

SYNTAX

```
DisplayClear ()
```

EXAMPLE

```
DisplayClear ()
```

DisplayLine

This command allows the script to place feedback text in the LCD during long operations. This is the same as the Display command except a carriage return is appended to the string.

The text display dialog is a scrolling display in which each line of text is scrolled up the display as the display is filled. Depending on font styles and sizes, up to five lines of text may be displayed at a time before the scrolling of the text begins to accommodate additional text.

SYNTAX

```
DisplayLine (DisplayString)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
DisplayString	s, t	The feedback information to be displayed on the LCD. The data can include one or more strings and variables, separated by commas.

EXAMPLE

```
declare i: status
...
DisplayLine ("Image Processing completed: ", status)
```

Although string variables are limited to 31 characters, there is no limitation on the display string length when formatted as displayed in the example. All text and variable data will be appended and displayed within the display dialog. If the text is too long to fit on a single line, the text will be wrapped based on word boundaries.

EndCapture

This command stops the capture process initiated by the `StartCapture` command. It may be used to terminate a group sequence. Although this command is primarily used to end a group capture sequence, it is good practice to issue this command when any type of capture sequence completes, as internal camera states will be restored to the condition that existed before the directed capture.

The parameter `GName` is returned if a group capture is being performed; otherwise, an empty string is returned.

SYNTAX

```
EndCapture (GName)
EndCapture ()
```

PARAMETER DEFINITION

Parameter Name	Type	Description
GName	n	The optional group name, stored in tag <code>grfd</code> . It is in the form of a DOS name but will have no extension, since it is also the folder name in which the images are stored.

EXAMPLE

```
declare n: GName
EndCapture (GName)
```

EraseFile

This command allows the user to delete image files or any other type of file on the internal storage or the removable media.

SYNTAX

```
EraseFile (path, filename)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
path	s	The path received by means of the <code>GetFileInfo</code> or <code>GetNewFileInfo</code> commands. All path names are relative to the root directory.
filename	n	The DOS file name received by means of the <code>GetFileInfo</code> or <code>GetNewFileInfo</code> commands.

EXAMPLE

```
EraseFile ("CAMERA01/", "IM000001.JPG")
```

FileClose

This command closes the open file pointed to by `FileId`.

SYNTAX

```
FileClose (FileId)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
FileId	u	This is the FileId returned with the FileOpen command. It identifies the file to be closed.

EXAMPLE

```
declare u: fileId
FileOpen (... , fileId)
...
FileClose (fileId)
```

FileOpen

This command opens a file with the specified `FileName`. If the file does not exist, it is created. Currently a maximum of 16 files can be opened at one time. Note that you can only write to text (TXT) files. Writing to a .JPG or .CAM file using this command will corrupt the file.

Note Because the `Read()` and `ReadLine()` commands have been added at Digita Script V 1.5, any write is targeted at the current file pointer (writes were previously targeted at the EOF). It is therefore possible to open a file, read some data, then write to the file thereby destroying the remaining data.

SYNTAX

```
FileOpen (DriveNo, Path/FileName, FileMode, FileId)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DriveNo	u	The drive on which the file will reside. Currently 1 is used for internal storage and 2 is used for removable media.
FileName	s	DOS name that specifies the name of the file, including the directory path for the file.
FileMode	u	Specifies the file mode. 1-Append 2-Open at the beginning of the file. Used for <code>Read()</code> or <code>ReadLine()</code> . ^a 3-Deletes an existing file, then creates a new empty file of the same name. ^a
FileId	u	The FileId that is returned for use in the other file logging commands.

^a File modes 2 and 3 are only applicable for scripts written for products compatible with Digita Script V1.5 or greater.

EXAMPLE

```
declare u: fileId
FileOpen (2, "CAMERA01/SAMPLE.TXT", 1, fileId)
```

FindString

Note This command is only recognized by products compatible with Digita Script V1.5 or greater. This command finds a substring 'find' within the string 'source', beginning at 'start'. The pointer is left at the substring's start point and the integer value of the start point is returned to 'location'. A value of -1 is returned if the substring is not found.

SYNTAX

```
Substring(s|t:source, u:start, s|t:find, i:location)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
source	s, t	Holds the original string.
start	u	Start position of the substring.
find	s, t	Name of the substring.
location	i	Start location of the substring.

EXAMPLE

```
declare t: source
declare s: find
declare u: start, end
declare i: location
source = "washington"
find="ton"
start=0
FindString(source, start, find, location)
FindString(source, 0, "ton", location)
```

Return value for this example: 8

GetCameraDefault

This command requests the user or factory default value of the camera parameter setting. The user default value is the value that will be saved when the camera is powered down. This allows user configurable information to be available when the camera is power cycled. The factory defaults are values that may be used to reset the camera back to factory-defined values. These defaults are not modifiable by the user.

Note The alternate command, GetParameterDefault, can be used in scripts written for Digita Script V 1.5 compatible devices.

SYNTAX

```
GetCameraDefault (DefaultSource, PName, PNameValue)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DefaultSource	u	Specifies which default to use. A value of 0 indicates that the factory default should be used, while a value of 1 indicates that the user default should be used.
PName	s	The product information parameter requested. "Camera Capabilities Parameters" starting on page 67 lists all the parameters currently available.
PNameValue	varies ^a	Contains the item requested in the PName parameter. Refer to "Camera Capabilities Parameters" starting on page 67 for detailed information.

- a. The value returned is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type. In the event that the variable's buffer space is less than the returned data, the data will be truncated to the size of the buffer space.

EXAMPLE

```
declare u: aelk
GetCameraDefault (0, "aelk", aelk)
```

GetCameraState

This command requests the current setting for a particular camera parameter. The current camera values may or may not be the same as the user default values. The current camera settings are temporary settings that are not saved during a power-cycle.

Note The alternate command, `GetParameterState`, can be used in scripts written for Digita Script V 1.5 compatible devices.

SYNTAX

```
GetCameraState (PName, PNameValue)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
PName	s	Specifies the camera parameter requested. "Camera Capabilities Parameters" starting on page 67 lists all the parameters currently available.
PNameValue	varies ^a	Contains the item requested in the PName parameter. Refer to "Camera Capabilities Parameters" starting on page 67 for detailed information.

- a. The value returned is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type. In the event that the variable's buffer space is less than the returned data, the data will be truncated to the size of the buffer space.

EXAMPLE

```
declare u: hint
GetCameraState ("hint", hint)
```

GetCameraStatus

This command returns three bit fields, SystemStatus, CaptureStatus and VendorStatus, which contain data about the current state of the camera.

Note The alternate command, GetParameterStatus, can be used in scripts written for Digita Script V 1.5 compatible devices.

SYNTAX

```
GetCameraStatus (SystemStatus, CaptureStatus, VendorStatus)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
SystemStatus	b	System status bit flag; see below for details.
CaptureStatus	b	Capture status bit flag; see below for details.
VendorStatus	b	Vendor status bit flag; see below for details.

The first bit field, `SystemStatus`, contains 32 separate logical flags. If a camera cannot report some of these conditions, the related bit is cleared and the condition reported as false. The bit positions do not change on this account, and all functions must be represented. The following flags are currently defined, and they are listed in order, with the most significant bit first:

Position	Flag	Details
1	ed1a	Expansion disk 1 is installed and available. This is the most significant bit.
2	ed2a	Expansion disk 2 is installed and available.
3	ramda	RAM disk is installed and available.
4	ipip	Image Processing In Progress. If this bit is set, images are still being processed.
5	memf	When the CompactFlash card is full this flag is <code>true</code> (set to 1). The image processing chain stops until the image data has been removed by means of the <code>GetFileInfo/GetNewFileInfo</code> and <code>EraseFile</code> commands. The flag is <code>false</code> when memory is not full.
6	pwra	Power alert indicates that the camera has entered a low-power state because of low batteries.
7	flsc	File list status change indicates that the file list from <code>GetFileInfo/GetNewFileInfo</code> has changed because of a successful capture. When this flag is set, you should call <code>GetNewFileInfo</code> or <code>GetFileInfo</code> to get a list of the changes. Either of these commands will clear this bit. This bit is set when processing of an image file has been successfully completed.
8	trun	Timer running indicates that the delay timer or time lapse/burst timer function is counting. <code>StartCapture</code> commands are not accepted at this time.
9	stst	Self test is in progress.
10	stsc	Self test is complete.
11	cerr	Camera error indicates that an error has occurred in the camera. Call <code>GetError</code> to determine the error and to clear this bit.
12	mcro	The macro flag indicates that the macro position is engaged. This can occur in response to a <code>SetCameraState</code> command or when you press the macro button on the camera.
13	sbs1	Shutter button in the S1 position indicates that you have pressed the shutter button half way down. This bit reflects the real-time status of the shutter button.
14	sbs2	Shutter button in the S2 position indicates that you have pressed the shutter button all the way down. This bit reflect the real-time status of the shutter button.
15	zoom	Zooming indicates that you are currently zooming the camera manually. This bit reflects the real-time status of the zoom buttons.
16	extp	External power indicates that external power is being applied to the camera.
17	edch	Expansion disk changed indicates that the expansion disk has been changed. This bit is cleared after a <code>GetFileInfo</code> or a <code>GetNewFileInfo</code> command has been executed.
18-32	rsvd	The rest of these bits are reserved

The second bit field, `CaptureStatus`, contains 32 separate logical flags. The following flags are currently defined. If a camera cannot report some of these conditions, the related bit is cleared and the condition reported as false. The bit positions do not change on this account, and all functions must be represented. The following flags are listed in order, with the most significant bit first:

Position	Flag	Details
1	camr	Camera ready indicates that the camera subsystem is ready to capture a picture, and that any focus and exposure processing is complete. This is the most significant bit.*
2	strc	Strobe charging indicates that the strobe subsystem is charging.
3	strr	Strobe ready indicates that the strobe is charged and ready to fire.
4	shkw	Shake warning indicates that the current (non-strobe) exposure requires a long exposure time and the camera should be held by a tripod or equivalent restraint.*
5	focl	Focus locked is active when Focus Lock mode is selected and the shutter button is pressed into the S1 or S2 position. Under these conditions, the flag indicates when the automatic focus (AF) function has been completed. *
6	expl	Exposure locked is active when Exposure Lock mode is selected and the shutter button is pressed into the S1 or S2 position. Under these conditions, the flag indicates when the automatic exposure (AE) function has been completed. *
7	oexp	Overexposure indicates that with the current camera settings you will get overexposure. *
8	uexp	Underexposure indicates that with the current camera settings you will get underexposure. *
9	stfa	Subject too far away indicates that the AF system has determined the subject is too far away for the strobe to be effective. *
10	stcl	Subject too close indicates that the AF system has determined that the subject is too close for the strobe to be effective or for the lens to focus on the subject properly. The focus system cannot distinguish between a subject that is too close or just on the edge of being too close, so if you focus on a subject at the minimum distance, this bit will be set. *
11-32	rsvd	The rest of these bits are reserved. *

* Not currently accessible through scripting.

The third bit field, `VendorStatus`, contains 32 separate logical flags. These bit flags are defined by each individual vendor. The following flags are currently defined. If a camera cannot report some of these conditions, the related bit is cleared and the condition reported as false. The bit positions do not change on this account, and all functions must be represented. The following flags are listed in order, with the most significant bit first:

Position	Flag	Details
1	hed1	If this bit is set, the camera cannot access expansion disk 1. *
2	hed2	If this bit is set, the camera cannot access expansion disk 2. *
3	hrmd	If this bit is set, the camera cannot access the RAM disk. *
4	stdc	Status display change indicates the camera status display has changed. *
5	srun	Script running indicates that a custom script function is running. *
6	serr	Script error indicates that an error occurred during a script. *
7	tnav	Thumbnail available indicates that a new thumbnail image is available for transfer. *
8	lors	When set, the left orientation sensor indicates when the camera is rotated into the left portrait position. This bit is active only during the S1 or S2 capture process, that is, when you are pressing the shutter button. *
9	rors	When set, the right orientation sensor indicates when the camera is rotated into the right portrait position. This bit is active only during the S1 or S2 capture process, that is, when you are pressing the shutter button. *
10	udos	When set, the upside down orientation sensor indicates when the camera is turned upside down. Most cameras do not support this function. *
11-32	rsvd	The rest of these bits are reserved. *

* Not currently accessible through scripting.

EXAMPLE

```
declare b: systemstat, capturestat, vendstat
GetCameraStatus (systemstat, capturestat, vendstat)
```

GetCapabilitiesCount

This command returns the number of capability list items defined for the specified camera parameter. If the camera parameter is not found or the parameter is not of a list type, an error is returned. This command is usually used in conjunction with the `GetCapabilitiesListItem` command.

SYNTAX

```
GetCapabilitiesCount (parameter-tag, list-count, default-value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
parameter-tag	s	Camera parameter whose capability list count is requested.
list-count	u	The number of items defined within the list. This value will be zero if the parameter-tag is invalid or if the capability type for the parameter is not a list.
default-value	u	The factory default list item. This value should be guaranteed to be one of the values within the list. This value is not the default list index but the value within the name-value pairing.

EXAMPLE

```
declare u: wCount, wValue
GetCapabilitiesCount ("wmod", wCount, wValue)
```

GetCapabilitiesListItem

This command returns a name-value pair for the specified camera parameter's list index. This command is usually used in conjunction with the `GetCapabilitiesCount` command.

SYNTAX

```
GetCapabilitiesListItem (parameter-tag, list-index, item-desc, item-value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
parameter-tag	s	Camera parameter whose capability list count is requested.
list-index	u	The index for the capability list item to be returned to the script. Note that the index starts with 0.
item-desc	s	The description (name) of the item as defined within the capability.
item-value	u	Value that represents the item. No value should be defined twice within the same list.

EXAMPLE

```
declare s: wDesc
declare u: wValue
GetCapabilitiesListItem ("wmod", 3, wDesc, wValue)
```

GetCapabilitiesRange

This command returns the min-max range information associated with a specified camera parameter. The script writer is responsible for ensuring that the camera capability supplied is of the correct type; otherwise, an error status is returned.

SYNTAX

```
GetCapabilitiesRange (parameter-tag, min-value, max-value, default-value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
parameter-tag	s	Camera parameter whose capability range detail is requested.
min-value	i,u ^a	The minimum value defined within the range.
max-value	i,u	The maximum value defined within the range.
default-value	i ^b ,u	The factory default value.

- a. Note that min-value, max-value and default-value must all be of the same type for the command to function correctly. That is, all three must be of type i or all three must be of type u.
- b. The value returned is based on the capability range type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter range type, and the last three parameters (min, max, and default) are required to be the same data type.

EXAMPLE

```
declare i: min, max, default
GetCapabilitiesRange ("wbc", min, max, default)
```

GetCapabilitiesValue

This command returns the capability value defined for the specified camera parameter. An error is returned if the camera parameter is not found or is not a string value type.

SYNTAX

```
GetCapabilitiesValue (parameter-tag, capability-value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
parameter-tag	s	Camera parameter whose capability list item is requested.
capability-value	s	Value as defined within the system. This string may be up to 31 characters in length.

EXAMPLE

```
declare s: cvalue
GetCapabilitiesValue ("cmne", cvalue)
```

GetCapabilityType

This command returns the capability type associated with the specified camera parameter. Depending on the outcome of the call to this command, additional capability information may be obtained by invoking one of the other specific capability commands.

SYNTAX

```
GetCapabilityType (parameter-tag, capability-type)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
parameter-tag	s	Camera parameter whose capability type is requested.
capability-type	u	The type of the camera capability. The possible values returned are: 1 = name-value list capability type 2 = min-max capability type 3 = simple value type

EXAMPLE

```
declare u: captype
GetCapabilityType ("wbc", captype)
```

GetClock

This command gets the current date and time. All zeros are returned if the clock is not set or not running. Otherwise, the command gets the current state of the clock.

SYNTAX

```
GetClock (date, time)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
date	u	The hexadecimal encoding of MM/DD/YY. For example, for 3/15/95 the value is 0x031595.
time	u	The hexadecimal encoding of HH/MM/SS. For example, for 9:15:30 PM the value is 0x211530. Note that hours are in the 24-hour format.

EXAMPLE

```
declare u: date, time
GetClock (date, time)
```

GetString

This command returns a string with the date formatted as specified in Table 7 on page 147 and Table 9 on page 148. The format is controlled by the parameters `rgnc` (Region Code) and `dfmt` (Date Format).

SYNTAX

```
GetString (DateString)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
DateString	s	Contains the formatted date information.

EXAMPLE

```
declare s: datestring
GetString (datestring)
```

GetError

This command clears the error flag when the `GetCameraStatus` command returns a `cerr` flag. The command also returns the last error that occurred along with the description of the error.

The status of this command will be valid only if the device supports error logging.

SYNTAX

```
GetError (date, time, error-code, error-description)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
date	s	Table 6 on page 147, Table 7 on page 147, Table 8 on page 147 and Table 9 on page 148 provide detailed information about time and date format. The format of the date is not modified by the region code.
time	s	Hours are in the 24-hour format. The format of the time is not modified by the region code.
error-code	i	The error code.
error-description	s	The NULL-terminated error description string.

EXAMPLE

```
declare s: errdesc, date, time
declare i: errcode
GetError (date, time, errcode, errdesc)
```

GetFileCount

This command returns the total number of captured images resident on the disk.

SYNTAX

```
GetFileCount (file-count)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
file-count	u	The total number of images.

EXAMPLE

```
declare u: filecount  
GetFileCount (filecount)
```

GetFileInfo

This command allows a script to access the list of all captured files available in the current camera. GetFileInfo can be used in response to the FileListStatusChange flag (flsc) reported by the GetCameraStatus command. This flag is set when new files are created as the result of a StartCapture command. GetFileInfo will clear this flag.

SYNTAX

```
GetFileInfo (file-index, path, name, length, status)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
file-index	u	Iterator value. This value should be between zero and (file-count - 1).
path	s	Path for this image, including "/".
name	n	DOS name of the image file. File names are IMnnnnnn.ext, where nnnnnn is the image number and ext is one of the file extensions.
length	u	Length of the file in bytes; this is 1 if the length is unknown.
status	b	File status BitFlags (see separate table below).

File status BitFlags, listed in order with most significant bit first:

Name	Description
ptyp	Picture type. The value 1 indicates full size; the value 0 indicates thumbnail.
piav	Partial image available. The value 1 is true; 0 is false.
ipcm	Image processing complete. The value 1 is true; 0 is false.
piwm	Picture in working memory. The value 1 is true; 0 is false.
prmd	Picture in internal storage. The value 1 is true; 0 is false.
ped1	Picture in expansion disk 1. The value 1 is true; 0 is false.
ped2	Picture in expansion disk 2. The value 1 is true; 0 is false.

EXAMPLE

```
declare s: path
declare n: name
declare u: length
declare b: status
GetFileInfo (3, path, name, length, status)
```

GetFileTag

This command returns the value of the specified file tag.

SYNTAX

```
GetFileTag (path, file-name, tag, return-value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
path	s	The path of the file received by means of the <code>GetFileInfo</code> or <code>GetNewFileInfo</code> command.
file-name	n	The DOS name of the file received by means of the <code>GetFileInfo</code> or <code>GetNewFileInfo</code> command.
tag	s	Specifies which file tag is requested. "Camera Capabilities Parameters" starting on page 67 lists all the parameters currently available.
return-value	varies ^a	Contains the item requested in the tag parameter. Refer to "Camera Capabilities Parameters" starting on page 67 for more details.

- a. The value returned is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type. In the event that the variable's buffer space is less than the returned data, the data will be truncated to the size of the buffer space.

EXAMPLE

```
declare u: wmrc
GetFileTag ("CAMERA01/", "IM000001.JPG", "wmrc", wmrc)
```

GetImageSpecifications

This command gets the hardware-related image information. This includes information about the camera's **zone** organization, and it is needed to set up the Camera Control Manager and other application-related data structures. Zone organization varies from camera to camera, so the information returned is product specific in terms of zone definition. All other data must be supported by all Digita cameras.

If type 0 is specified for zones, this implies a shorter `ImageSpecificationsList`, and no data is provided. Zones can be implemented as software or hardware and that factor is not indicated in this section.

SYNTAX

```
GetImageSpecifications (spec-type, spec-param, spec-data)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
spec-type	u	An unsigned value indicating the specification whose parameter is to be retrieved (1 = CCD, 2 = Thumbnail, 3 = Screennail, 4 = Focus Zone, 5 = Exposure Zone).
spec-param	s	A four-character value defining the parameter to retrieve from the image specification. The parameters for each of the specification types are defined below.
spec-data	u	The unsigned variable in which the requested data will be returned.

ImageSpecificationList contains the following data, all of which are unsigned integer (UInteger) values. Following are the lists of supported specifications and their corresponding parameters:

1 = CCD (charge-coupled device) specifications

The following CCD information is available through this specification type:

ptrn	Defines the CCD pattern (e.g., RGB triplet or Bayer pattern)
hpix	CCD pixels horizontal - Total number of pixels on the CCD in the horizontal direction.
vpix	CCD pixels vertical - Total number of pixels on the CCD in the vertical direction.
rhpx	CCD ring pixels horizontal - Number of pixels clipped from the horizontal edge.
rvpx	CCD ring pixels vertical - Number of pixels clipped from the vertical edge.
bcol	Number of bad columns - Total number of bad columns in the CCD at manufacture.
bpix	Number of bad pixels - Total number of bad pixels in the CCD at manufacture.

2 = Thumbnail specifications

The following thumbnail information is available through this specification type:

ttyp	Thumbnail Type (a value of 4 indicates YCC 422)
hpix	Thumbnail pixels horizontal
vpix	Thumbnail pixels vertical
fsiz	Thumbnail file size

3 = Scrennail specifications

The following scrennail information is available through this specification type:

styp	Scrennail type (a value of 0 indicates no scrennail and a value of 1 indicates JPEG compressed)
hpix	Scrennail pixels horizontal
vpix	Scrennail pixels vertical

4 = Focus zone specifications

The following focus zone information is available through this specification type:

ztyp	Zone type (a value of 0 = none, 1 = rectilinear array, 2 and up indicate other types of arrays)
hcnt	Number of horizontal zones
vcnt	Number of vertical zones
horg	Origin of horizontal zones
vorg	Origin of vertical zones
hsiz	Horizontal zone size
vsiz	Vertical zone size

5 = Exposure zone specification

The following exposure zone information is available through this specification type:

ztyp	Zone type (a value of 0 = none, 1 = rectilinear array, 2 and up indicate other types of arrays)
hcnt	Number of horizontal zones
vcnt	Number of vertical zones
horg	Origin of horizontal zones
vorg	Origin of vertical zones
hsiz	Horizontal zone size
vsiz	Vertical zone size

GetMarkedImage

This command allows the user to iterate through the number of marked images to retrieve the image file. The image file can then be used in calls to get and set image file tag information. If no images are currently marked and if the iterator value indicates the first item (0), then the current image file is returned; otherwise, the image corresponding to the iterator value is returned. The files are not guaranteed to be ordered.

SYNTAX

```
GetMarkedImage (iterator, image-file-path, image-file-name)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
iterator	u	The index of the next image file to retrieve. The value is an unsigned integer between zero and (image-count - 1).
image-file-path	s	The DOS pathname of the image relative to the root directory.
image-file-name	n	The DOS filename of the image. This file name does not include the path of the file.

EXAMPLE

```
declare s: file_path
declare n: file_name
GetMarkedImage (1, file_path, file_name)
```

GetMarkedImageCount

This command returns the number of images currently marked. If no images are currently marked, then this command will return a value of 1, and the current image will be regarded as marked. Since images can only be marked in review mode, scripts using this command should not be run in other modes, such as capture.

SYNTAX

```
GetMarkedImageCount (image-count)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
image-count	u	The number of marked images.

EXAMPLE

```
declare u: count
GetMarkedImageCount (count)
```

GetNewFileCount

This command is used in response to the `FileListStatusChange (flsc)` flag reported by the `GetCameraStatus` command. This flag is set when new files are created from the result of a `StartCapture` command. This command returns the number of new images resident on the disk.

SYNTAX

```
GetNewFileCount (new-file-count)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
new-file-count	u	The number of new image files found on the disk.

EXAMPLE

```
declare u: count
GetNewFileCount (count)
```

GetNewFileInfo

This command is used in response to the `FileListStatusChange (flsc)` flag reported by the `GetCameraStatus` command. This flag is set when new files are created from the result of a `StartCapture` command. This command returns the indicated new file name and clears the flag.

You can use `GetNewFileCount` to determine the number of new image files and then iterate through the list with `GetNewFileInfo` to retrieve all new file names.

SYNTAX

```
GetNewFileInfo (file-index, path, name, length, status)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
file-index	u	Iterator value. This value should be between zero and (new-file-count - 1).
path	s	Path for this image, including "/".
name	n	DOS name of the image file. File names are IMnnnnnn.ext, where nnnnnn is the image number and ext is one of the file extensions.
length	u	Length of the file in bytes; this is 1 if the length is unknown.
status	b	File status BitFlags (see separate table below).

File status `BitFlags`, listed in order with most significant bit first:

Name	Description
ptyp	Picture type. The value 1 indicates full size; the value 0 indicates thumbnail.
piav	Partial image available. The value 1 is true; 0 is false.
ipcm	Image processing complete. The value 1 is true; 0 is false.
piwm	Picture in working memory. The value 1 is true; 0 is false.
prmd	Picture in internal storage. The value 1 is true; 0 is false.
ped1	Picture in expansion disk 1. The value 1 is true; 0 is false.
ped2	Picture in expansion disk 2. The value 1 is true; 0 is false.

EXAMPLE

```
declare s: path
declare n: name
declare u: length
declare b: status
GetNewFileInfo (3, path, name, length, status)
```

GetOption

This command is used in conjunction with the `SetOption` command. `GetOption` prompts the user for input based on the option list defined by the number of `SetOption` commands performed. The user has the ability to select from the list or to exit from the script at this point. Once the user selects an option, the option's identifier value is returned to the script, allowing the script to continue operation.

SYNTAX

```
GetOption (option-id)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
option-id	u	Numeric value that identifies the selected option defined within the option list created with <code>SetOption</code> .

EXAMPLE

```
declare u: choice
GetOption (choice)
```

GetPowerMode

This command retrieves the camera's current power level.

SYNTAX

```
GetPowerMode (PowerState)
```

PARAMETER DEFINITION

PowerState Value contains unsigned integers from 1 to 5 that indicate the power state of the camera. Some devices will not have power state 4. The values contained are as follows:

- 5 = Full operation. Allows full operation of all camera functions.
- 4 = Reduced capture. This state applies in devices that have the option of low charge for flash. Image capture can occur with flash, but the time required to charge the flash is longer than in full operation mode.
- 3 = Minimal capture. This state disallows the use of the strobe entirely, and may include other power reduction techniques within the camera operation, such as running only one motor at a time, and so forth. This mode trades performance for power reduction.
- 2 = Processing and Viewing only. This state does not allow the use of the image capture subsystem at all. You therefore cannot take any pictures. This state will allow the LCD to be on and allows any image to complete it's processing.

1 = Processing only. This state only allows any image processing to be completed. The LCD is not on. If power falls below state 1, the serial port is powered down, terminating the connection.

EXAMPLE

```
declare u: power-state
GetPowerMode (power-state)
```

GetProductInfo

This command requests specific information about the product by parameter. The information includes such things as the name of the camera vendor, the camera product name, hardware revision, and so forth. The parameters are described in “Product Information Parameters” starting on page 59.

SYNTAX

```
GetProductInfo (PName, PNameValue)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
PName	s	Specifies the product information parameter requested. “Product Information Parameters” starting on page 59 lists all the ProductInfo parameters currently available.
PNameValue	varies ^a	The value of the product information parameter requested. Refer to “Product Information Parameters” starting on page 59 for more detailed information.

- a. The value returned is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type.

EXAMPLE

```
declare u: strobe-count
GetProductInfo ("scnt", strobe-count)
```

GetScriptName

This command retrieves the long name of a script, which is defined within the script itself by means of the name statement. This command will return the text string that the script supplies.

SYNTAX

```
GetScriptName (DOSName, NameString)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DOSName	n	This is the DOS Name of a currently installed script (.CSM file) from the <code>GetSystemFileInfo</code> command. If a name is given that is not currently available, an error is reported.
NameString	s	This is the response to DOSName. The string contains the definition from the name statement. If no name is found in the script, an empty (NULL) string is returned.

EXAMPLE

```
declare s: script-name
GetScriptName ("TEST_01.CSM", script-name)
```

GetStorageStatus

This command determines the status of storage available to the camera. The amount of storage remaining is an estimate that may vary due to JPEG compression.

SYNTAX

```
GetStorageStatus (StorageSelector, TakenCount, AvailableCount, RawCount)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
StorageSelector	u	This is an optional parameter. If no StorageSelector is specified, this command returns the current number of captured files in the camera (in all camera folders), how many more images can be captured (at the current compression and capture mode settings), and how many images you can capture without waiting for image processing and compression to be completed. Other values of StorageSelector are: 0 = Return value for the internal drive 1 = Return value for the removable memory card (1 only) 2 = Return value for the second memory card. An error is reported if the product does not support two memory card slots.
TakenCount	u	Indicates how many captured files exist in the specified storage.
AvailableCount	u	The approximate number of compressed files that can fit in the free storage within the specified storage area. This number is an estimate that may vary due to JPEG compression.
RawCount	i	Indicates how many unprocessed files can fit in the available space within the specified storage area. If the camera does not store the raw image data this value is set to -1.

EXAMPLE

```
declare u: taken, available
declare i: raw-count
GetStorageStatus (1, taken, available, raw-count)
```

GetString

This command allows the script to request an input text string from the user.

SYNTAX

```
GetString (Prompt, StringVar)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
Prompt	s	The text that will be displayed as a prompt on the color LCD.
StringVar	s	The feedback information from the user. The user string is currently limited to 31 characters. The display will signal an audible alarm and inhibit the additional entry of text once the limit is reached.

EXAMPLE

```
declare s: feedback
GetString ("Enter desired value:", feedback)
```

GetSystemFileCount

This command returns the total number of system files in the `SYSTEM` folder plus the number of ROM-based scripts that exist on the camera's internal hard drive. ROM-based scripts are system-level scripts that have been built directly into the camera's operating environment.

SYNTAX

```
GetSystemFileCount (system-file-count)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
system-file-count	u	The number of resident system files.

EXAMPLE

```
declare u: system-files
GetSystemFileCount (system-files)
```

GetSystemFileName

This command returns a single system file name based on the supplied iterator value. These files must be located in the `SYSTEM` folder or be ROM-based files (scripts) in order to be recognized by this command. ROM-based scripts are system-level scripts that have been built directly into the camera's operating environment.

SYNTAX

```
GetSystemFileName (file-index, filename)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
file-index	u	The iterator value, which should be between 0 and (system-file-count - 1).
filename	n	The DOS name of a system file as defined within the <code>SYSTEM</code> folder.

EXAMPLE

```
declare n: file-name
GetSystemFileName (3, filename)
```

GetTimeString

This command returns a string with the time formatted as specified in Table 6 on page 147 and Table 8 on page 147. The format is controlled by the parameters `rgnc` (Region Code) and `tfmt` (Time Format).

SYNTAX

```
GetTimeString (TimeString)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
TimeString	s	The formatted time information

EXAMPLE

```
declare s: time
GetTimeString (time)
```

MakeFolder

Note This command is only recognized by products compatible with Digita Script V1.1 or greater. This command creates a new directory on the file directory based on the supplied path. The specified path can only create one new directory; i.e., the enter path. The last item in the path must already exist. Thus, in order to create a directory three levels down on a blank disk, three MakeFolder statements must be used. An error will occur if an attempt is made to create more than one directory.

SYNTAX

```
MakeFolder(DriveNo, Path)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DriveNo	u	The drive on which the file will reside. Currently 1 is used for internal storage and 2 is used for removable media.
PName	s, t	The pathname of the directory relative to the root directory. Must include a leading "/".

EXAMPLE

```
MakeFolder (2, "/system/folder1")
```

MarkImage

Note This command is only recognized by products compatible with Digita Script V1.1 or greater. This command marks an image "filename" currently stored in the specified drive and located on the specified path.

SYNTAX

```
MarkImage(DriveNo, Path, filename)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DriveNo	u	The drive on which the file will reside. Currently 1 is used for internal storage and 2 is used for removable media.
Path	s,t	Directory path to the desired file.
FileName	n	DOS name that specifies the name of the file.

MarkAllImages

Note This command is only recognized by products compatible with Digita Script V1.1 or greater. This command marks all of the images on the CompactFlash card.

SYNTAX

```
MarkAllImages(DriveNo)
```

NumberToString

Note This command is only recognized by products compatible with Digita Script V1.5 or greater.
Converts a numerical value to a string.

SYNTAX

```
NumberToString(Number, String)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
Number	i, u, t, b	Number to be converted to a string value.
String	s, t	String variable to store the converted number.

EXAMPLE

```
declare i: ivar
declare s: svar
declare f: fvar
declare b: bvar

ivar = 3
fvar = 1.2
bvar = 0x12
NumberToString(ivar, svar)
NumberToString(fvar, svar)
NumberToString(bvar, svar)
```

Read

Note This command is only recognized by products compatible with Digita Script V1.5 or greater.
Reads data from a text file according to the delimiter type set in SetDelimiter(). The default delimiter is whitespace (space and tab). The format of Read() is similar to Write(), i.e. a file identifier following by variables. The file pointer is left after the last data field. This command reports an error of 4 at the end of the file.

- FileOpen must be executed to get the file identifier (FileId) for the Read function.

SYNTAX

```
Read(FileId, Variable)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
FileId	u	Identifier returned from the file open command.
Variable	varies	Variable to which the delimited data is being written

EXAMPLE (THE DATA FILE FOR THIS DATA IS OF THE FORM: 13, 15, 2.34, 1, ROGER)

```
declare i: ivar
declare u: uvar
declare f: fvar
declare s: svar
SetDelimiter (comma)
Read (FileId, ivar, uvar, fvar, bvar, svar)
```

ReadLine

Note This command is only recognized by products compatible with Digita Script V1.5 or greater. Reads one line of data from a text file according to the delimiter set in Set Delimiter(). If there are insufficient variables to store the data, the remainder of the line is not stored. If there are more variables than data in the line, the extra variables are untouched.

- FileOpen must be executed to get the file identifier (FileId) for the Read function.

SYNTAX

```
Read(FileId, VariableType)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
FileId	u	Identifier returned from the file open command.
Variable Type	varies	Variable to which the delimited data is being written

RestoreCameraDefault

This command restores the current parameter value to either the user default or the factory default. If PName is NULL, all camera parameters are restored to the specified defaults.

Note The alternate command, RestoreParameterDefault, can be used in scripts written for Digita Script V 1.5 compatible devices.

SYNTAX

```
RestoreCameraDefault (DefaultSource, PName)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DefaultSource	u	Specifies which source to use to update the current state value. A value of 0 indicates that factory defaults should be used, while a value of 1 indicates that user defaults should be used
PName	s	This specifies the parameter to be restored. If this is NULL (empty string), all parameters will be restored to either the user defaults or the factory defaults. "Camera Capabilities Parameters" starting on page 67 lists all the parameters currently available.

EXAMPLE

```
RestoreCameraDefault (0, "")
```

RunApp

This command restarts the camera using the selected .CAM file until the camera is restarted again. If the name sent is DEFAULT.CAM, the camera is restarted and returns to use the original application. An error is returned if the AppName is not found or if an incorrect extension is used. If this command is successful, there will be no response sent back to the script.

SYNTAX

```
RunApp ( AppName )
```

PARAMETER DEFINITION

Parameter Name	Type	Description
AppName	n	This is the DOSName of the file to be run. It should be a .CAM file name returned by the GetSystemFilesInfo command, or the name "DEFAULT.CAM" (to restart with the original application).

EXAMPLE

```
RunApp ( "USERAPP.CAM" )
```

RunScript

This command runs a specified script file. The script file must be stored in the SYSTEM folder. If this command is successful, the current script will be terminated and the specified script executed.

Note A script cannot run itself. Also, if RunScript cannot find the specified script, it returns status error 7 ("Specified file does not exist"). If an empty name is specified, RunScript returns status error 9 ("The file specified is an invalid or unknown type").

SYNTAX

```
RunScript ( script-name )
```

PARAMETER DEFINITION

Parameter Name	Type	Description
script-name	n	This is the DOSName of the file to be run. It should be a .CSM file name returned by the GetSystemFileName command.

EXAMPLE

```
RunScript ( "TEST_02.CSM" )
```

SerialClose

This command releases the serial port. The port must be open or an error results.

SYNTAX

```
SerialClose ( )  
example  
SerialClose ( )
```

SerialOpen

This command opens the serial channel at the selected baud rate. If the serial channel is in use, an error is returned. This command allows scripts to control or communicate with external devices via the serial port.

SYNTAX

```
SerialOpen (BaudRate)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
BaudRate	u	Indicates the asynchronous serial baud rate. The value is rounded down to the closest value supported by the hardware. Legal values are 115200, 57600, 38400, 28800, 19200, 14400, and 9600 bps.

EXAMPLE

```
SerialOpen (28800)
```

SerialReceive

This command waits to receive an expected transmission. The serial port must be opened by the `SerialOpen` command before this command is issued; otherwise, an error results. This command allows scripts to control or communicate with external devices through the serial port.

The command will block (not return data) until the serial port receives the required amount of data, as specified in a previous `SerialSend` command.

Note If the requested amount of data is not received, the command returns with an error after the timeout period has elapsed.

SYNTAX

```
SerialReceive (ReceiveData)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
ReceiveData	varies	This is the data requested and received. The length of this data is specified in the <code>ReceiveSize</code> parameter.

EXAMPLE

```
declare s: receive-data
SerialReceive (receive-data)
```

SerialSend

This command sends the indicated data and sets up the next receive cycle. The serial port must be opened by the `SerialOpen` command before this command is issued; otherwise, an error results.

This command allows scripts to control or communicate with external devices through the serial port.

SYNTAX

```
SerialSend (SendSize, ReceiveSize, SendData)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
SendSize	u	Specifies the size in bytes of the data that will be sent. This can be 0, in which case the SendData parameter is not required.
ReceiveSize	u	Specifies the size in bytes of the data that the serial port expects to receive. This parameter also can be 0, in which case the command returns immediately without ReceiveData. If a receive cycle is set up where ReceiveSize is greater than 0, the serial hardware is set up to receive this data. This command returns immediately after the SendData parameter is sent.
SendData	varies	This is the data to be transmitted at the set baud rate. The length of the data is specified in the SendSize parameter.

EXAMPLE

```
SerialSend (6, 6, "test01")
```

SerialSendReceive

This command sets up for the next data exchange through the serial port. The port must be opened by the `SerialOpen` command before this command is issued; otherwise, an error results. This command allows scripts to control or communicate with external devices by means of the serial port.

The command will block (not return data) until the serial port receives the required amount of data.

Note If the requested amount of data is not received, the command returns with an error after the timeout period has elapsed.

SYNTAX

```
SerialSendReceive (SendSize, ReceiveSize, SendData, ReceiveData)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
SendSize	u	Specifies the size in bytes of the data to be sent. This can be 0, in which case the SendData parameter is not required.
ReceiveSize	u	Specifies the size in bytes of the data that the serial port expects to receive. This parameter also can be 0, in which case the command returns immediately without ReceiveData.
SendData	varies	This is the data to be transmitted at the set baud rate. The length of the data is specified in the SendSize parameter.
ReceiveData	varies	This is the data requested and received. The length of this data is specified in the ReceiveSize parameter.

EXAMPLE

```
declare s: receive-data
SerialSendReceive (6, 6, "test02", receive-data)
```

SetCameraDefault

This command updates the user default value for a camera parameter setting. For range-type camera parameters, if a value set by the SetCameraDefault command is outside the allowable range specified by the GetCapabilitiesRange command, an error is returned and no action is taken. For list-type camera parameters, if a value set by SetCameraDefaults is not included in the list, an error is returned and no action is taken.

Note The alternate command, SetParameterDefault, can be used in scripts written for Digita Script V 1.5 compatible devices.

SYNTAX

```
SetCameraDefault (UpdateSource, PName, DefaultValue)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
UpdateSource	u	Specifies which source to use to update the user default value. A value of 0 indicates that the factory defaults should be used, a value of 1 indicates that the current values should be used, and a value of 2 indicates that the value in the DefaultValue parameter should be used.
PName	s	This specifies the parameter for which the default is to be set. "Camera Capabilities Parameters" starting on page 67 lists all the parameters currently available
DefaultValue	varies ^a	Default value to use in the update, if UpdateSource is set to 2.

- a. The data type is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type. In the event that the DefaultValue's buffer size is greater than the parameter's expected data, the DefaultValue's data will be truncated to the size of the parameter's expected buffer size.

EXAMPLE

```
SetCameraDefault (2, "aelk", 1)
```

SetCameraState

This command updates a current camera parameter setting. For range-type camera parameters, if a value set by the `SetCameraState` command is outside the allowable range specified by the `GetCapabilitiesRange` command, an error is returned and no action is taken. For list-type camera parameters, if a value set by `SetCameraState` is not included in the list, an error is returned and no action is taken.

Note The alternate command, `SetParameterState`, can be used in scripts written for Digita Script V 1.5 compatible devices.

SYNTAX

```
SetCameraState (PName, value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
PName	s	This is a camera PName. Appendix B, "Camera Capabilities Parameters" lists the available camera parameters.
value	varies ^a	The value of PName.

- a. The data type is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type. In the event that the `DefaultValue`'s buffer size is greater than the parameter's expected data, the `DefaultValue`'s data will be truncated to the size of the parameter's expected buffer size.

EXAMPLE

```
SetCameraState ("aek", 1)
```

SUPPLEMENTARY INFORMATION

Camera parameters that are not relevant for specific exposure or focus modes are ignored, but saved for possible use when a different mode is selected. For example, a value for aperture can be set, but is ignored in Auto mode and used in Aperture Priority mode.

SetCaptureMode

This command controls the type of capture sequence. This command has three selectors that enable one of three types of capture sequence: still, group and timelapse. This command must be called before a `StartCapture` command can be made.

SYNTAX

```
SetCaptureMode (mode)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
mode	u	The capture mode to be set. Possible values are still, group and tlap (time lapse).

EXAMPLE

```
SetCaptureMode (still)
```

SetClock

This command sets the current clock value from the information supplied. The format of the data is not modified by the region code.

SetClock does not “filter” dates in an attempt to verify whether the date/time parameters are valid. It is the responsibility of the script to intercept any invalid dates or times which the user might attempt to enter (such as trying to set the date to Feb. 31). If an invalid date or time is sent to the camera, the result is undefined.

SYNTAX

```
SetClock (date, time)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
date	u	The hexadecimal encoding of MM/DD/YY. For example, for 3/15/95 the value is 0x031595.
time	u	The hexadecimal encoding of HH/MM/SS. For example, for 9:15:30 PM the value is 0x211530. Note that hours are in the 24-hour format.

EXAMPLE

```
SetClock (0x101597, 0x083000)
```

SetDelimiter

Note This command is only recognized by products compatible with Digita Script V1.5 or greater.

This command is always used in conjunction with a Read command unless the default delimiter(whitespace) is desired. Four delimiter choices are available to separate the data fields: comma, whitespace, tab and none. If “whitespace” is selected, the spaces and tabs between data fields are considered to be a single delimiter. The “none” delimiter is the same as reading to the end of the line. The End-of-line (return character) always represents a delimiter regardless of the delimiter set. Delimiters are not returned with the data.

SYNTAX

```
SetDelimiter (comma|whitespace|tab|none)
```

EXAMPLE

```
SetDelimiter(comma)
Read(FileId, iVar, uVar, fVar, bVar, sVar. tVar)
```

SetOption

This command allows the script writer to display an option list to the user via the color LCD. The script writer issues a `SetOption` command for each item in the list, specifying a numeric identifier, the displayed label, and whether or not the option is the default; the defaulted option is displayed with a check mark next to the label.

SYNTAX

```
SetOption (identifier, label-text, is-default)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
identifier	u	A numeric value between 0x00000000 and 0xFFFFFFFF that uniquely identifies the option within the list. Values need not be sequential. Value is returned via the <code>GetOption</code> command for the selected option.
label-text	s	The label as it should appear to the user via the color LCD. The label may be truncated if it is too large to fit in the label region.
is-default	b	Optional argument; defaults to zero if omitted. Boolean value that indicates if the option is the default. Only one option can be defined as the default. If more than one option is set as the default, only the last item set will be displayed as the default value.

EXAMPLE

```
SetOption (1, "Label Text for Option 1", 0)
SetOption (2, "Label Text for Option 2", 0)
SetOption (3, "Label Text for Option 3", 0)
SetOption (4, "Label Text for Option 4 (Default)", 1)
```

SetPowerMode

This command will power off the camera. When this command is issued, the camera will power down. There will be no response from the camera.

SYNTAX

```
SetPowerMode ()
```

EXAMPLE

```
SetPowerMode ()
```

SetTitle

Note This command is only recognized by products compatible with Digita Script V1.5 or greater. This command changes the current string in the title bar to 'title'. The title bar display update is separate from the main window. SetTitle() may be called multiple times in a single script.

SYNTAX

```
SetTitle(s|t: title)
```

EXAMPLE

```
SetTitle("NewTitle")
```

SetUserFileTag

This command sets the value of the specified user tag. This applies to the tags that start with 'u' from "Image File Tags" starting on page 119.

SYNTAX

```
SetUserFileTag (path, file-name, tag, value)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
path	n	The path of the file received by means of the GetFileInfo or GetNewFileInfo command.
file-name	s	The DOSName of the file received by means of the GetFileInfo or GetNewFileInfo command.
tag	s	The tag of the item the user wants to set. Refer to "Image File Tags" starting on page 119.
value	varies ^a	The value of the item the user wants to set. Refer to "Image File Tags" starting on page 119.

a. The data type is based on the capability type of the parameter. It is up to the script writer to ensure that the variable is of the correct type. No error will be reported if the variable type does not match the parameter type.

EXAMPLE

```
SetUserFileTag ("CAMERA01/", "IM000001.JPG", "ur01", 1)
```

StartCapture

This command starts the capture process. The camera returns the response right away when performing a still or group capture. Timelapse capture sequences will only return a response when the capture sequence is complete or when the user selects the "Stop Script" option via the softkey button.

The script can determine when still image processing is complete by checking the Image Processing In Progress (ipip) flag in the GetCameraStatus command. If no parameters are passed with this command, a normal single still image is taken.

Note Be sure to use SetCaptureMode to define the capture type before invoking StartCapture.

Note It is important to check the return status from this command before attempting to process the new image file, as this command may return a status indicating that the camera is busy. This may occur in the event that all of the camera's capture buffers are currently in use.

The status variable should be cleared before issuing this command, as a successful invocation of this command will not update the return value; the return value may contain an invalid status when the command executes successfully (see examples below).

SYNTAX

```
StartCapture ()
StartCapture (group-name, start-count, auto-count)
StartCapture (image-count, timer-delay, capture-size)
```

PARAMETER DEFINITIONS

GROUP CAPTURE:

Parameter Name	Type	Description
group-name	s	A string of up to eight characters that identifies the group of the captured images. If the group already exists, the images are added to the group. If the group does not exist, a new group is created.
start-count	u	An unsigned integer stored in the group sequence number tag (grct) for the first image in the group. Normally, this selector is 1.
auto-count	b	Boolean value; if true, automatic counting occurs for any additional images in this group. If false, the group sequence number tag (grct) for all images is the same. This selector is normally true.

EXAMPLE FOR GROUP CAPTURE

```
declare i:status

SetCaptureMode (group)
StartCapture ("group1", 1, 1)

LOOP:
  status = 0
  status = StartCapture ()
  if status != 0
    Wait (5000)
    goto LOOP
end
```

TIME LAPSE CAPTURE:

Parameter Name	Type	Description
image-count	u	An unsigned integer that specifies the number of images to be captured.
timer-delay	u	An unsigned integer that specifies either the initial delay (for a single capture) or the time between captures for a time lapse sequence. This value is set in hundredths of a second increments (one second increments for pre V 1.5 releases) and indicates the time lapse between the issuance of StartCapture commands. If the time is set to a value that is below the time it takes to actually complete a capture, the capture rate is as fast as the camera can go. A value of zero is illegal.
capture-size	u	An integer that indicates the size reduction during capture. The range is 1 to 8. Legal values are: 1 = full size (no reduction) 2 = 1/4 size 4 = 1/16 size 8 = 1/64 size.

EXAMPLE FOR TIME LAPSE CAPTURE

- Digita Script V1.5 compatible

```
StartCapture (2, 6000, 1)
```

Note There are certain side effects to the time lapse capture mode, including the following:

- If size is reduced, watermarks are suppressed.
- If flash is ON or AUTO (and light level is low), the capture always waits for the flash to charge, even if this slows down the capture rate.
- Thumbnails are not created for reduced-size images (except for the first one in a series).
- Statistics are not generated for reduced-size images.
- Once the sequence starts, the script will be suspended until the timelapse sequence completes or the user stops the script execution via the “Stop Script” softkey.
- This command does not verify that sufficient file space is available before starting the capture sequence.

SUPPLEMENTARY INFORMATION

Once a capture is complete, the FileListStatusChange flag (flsc) is set for the GetCameraStatus command. The name of the file can then be retrieved with the GetNewFileInfo command.

SubString

Note This command is only recognized by products compatible with Digita Script V1.5 or greater.

Extracts and returns a substring from source string starting at ‘start’ for ‘count’ characters. Substring is stored in ‘destination’. First character of string starts at index 0. If ‘start’ is less than

the 'source' string, but 'start' + 'count' is longer than the 'source' string, then the substring extracted will be from 'start' to the end of the source string.

SYNTAX

```
Substring(source, start, count, destination)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
source	s, t	Holds the original string.
start	u	Start position. Number of characters from beginning of the data file
count	u	Number of characters to be extracted.
destination	s, t	Destination for the extracted characters.

EXAMPLE

```
declare s: source, output
declare u: start, end
source = "washington"
start = 0
finish = 5
Substring (source, start, finish, output)
Substring (source, 0, 15, output)
```

StringToNumber

Note This command is only recognized by products compatible with Digita Script V1.5 or greater. Converts a string to a numerical value.

SYNTAX

```
StringToNumber(s:String, Number:i,u,f,b)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
String	s	Describes the number to be converted. Can be either a literal string or a string variable. The input string consists of numerics(0-9), hex(0x,0X,a-f,A-F), positive sign(+), negative sign(-) and decimal point(.). If used, + or - must precede the number. The + sign is optional for positive numbers. The decimal point is also optional, even if conversion to a fixed point number is desired.
Number	varies	Variable to store the converted number.

EXAMPLE

```
declare s: svar
declare i: ivar
declare u: uvar
declare f: fvar

svar="1"
StringToNumber(svar, ivar)
StringToNumber(svar, uvar)
StringToNumber(svar, fvar)
```

UnMarkImage

Note This command is only recognized by products compatible with Digita Script V1.1 or greater. This command unmarks an image “filename” currently stored in the specified drive and located on the specified path.

SYNTAX

```
UnMarkImage(DriveNo, Path, filename)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
DriveNo	u	The drive on which the file will reside. Currently 1 is used for internal storage and 2 is used for removable media.
Path	s,t	Directory path to the desired file.
FileName	n	DOS name that specifies the name of the file

UnMarkAllImages

Note This command is only recognized by products compatible with Digita Script V1.1 or greater. This command unmarks all of the images on the CompactFlash card.

SYNTAX

```
UnMarkAllImages()
```

Wait

This command allows you to specify how many milliseconds you want the script program to wait before starting the next statement. If you wait until the timer expires, the script continues running. The user can exit the script during this command by selecting the appropriate softkey as defined by the color LCD.

SYNTAX

```
Wait (Milliseconds)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
Milliseconds	u	The number of milliseconds that you want the system to wait to start the next command.

EXAMPLE

```
Wait (5000)
```

WaitForShutter

This command allows the script to give control of the system back for user operation while the script is waiting for the shutter button to be pressed. This command is only valid when the camera is in 'Capture' mode.

SYNTAX

```
WaitForShutter (Prompt)
```

PARAMETER DEFINITION

Parameter Name	Type	Description
Prompt	s	This contains a quoted string or string variable to be used as a prompt.

EXAMPLE

```
WaitForShutter ("Take photo of kitchen")
```

Write

This command appends the specified data to an open file pointed to by `FileId`. No carriage return is added to the end of the data.

SYNTAX

```
Write (FileId, Data)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
FileId	u	This is the FileId returned with the FileOpen command. It identifies the file to which data will be written.
Data	s	This is the data written to the opened file. The data can include one or more strings and variables, separated by commas.

EXAMPLE

```
declare u: fileId
FileOpen (... , fileId)
Write (fileId, "Processing image ", count, " of ", total, ".")
```

Although variables of type string are limited to 31 characters, there is no limitation on the prompt string length when formatted as displayed in the example. All text/variable data will be appended and written out to the file.

WriteLine

This command appends the specified data to an open file pointed to by `FileId`. This command also adds a carriage return to the end of the data.

SYNTAX

```
WriteLine (FileId, Data)
```

PARAMETER DEFINITIONS

Parameter Name	Type	Description
FileId	u	This is the FileId returned with the FileOpen command. It identifies the file to which data will be written.
Data	s	This is the data written to the opened file. The data can include one or more strings and variables, separated by commas.

EXAMPLE

```
declare u: fileId
declare i: status
FileOpen (... , fileId)
...
WriteLine (fileId, "Image Processing completed: ", status)
```

Although variables of type string are limited to 31 characters, there is no limitation on the prompt string length when formatted as displayed in the example. All text/variable data will be appended and written out to the file.

Appendix A Product Information Parameters

The following tables define the product information (`ProductInfo`) parameters supplied in the `PName` parameter. These parameters are accessed through the `GetProductInfo` script command.

Note Specific products are listed only when their details or defaults differ from the current Digita Script standard.

cacn – Application Copyright Notice

Definition: Copyright notice for the control application
 Data Type: String
 Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	.	(c) 1998 FlashPoint Technology	ROM

carv – Control Application Version

Details: Defines the version for the currently running Application. Defined by the Application itself.
 Data Type: UInt
 Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CCDD. Values are BCD, from 0 to 9, a to f. Leading zeros within group (AA, for example) are suppressed. For example, version 2.0.3.a would appear as 0x0200030a.	varies by product	ROM

cccn – CCS Copyright Notice

Definition: The copyright notice for the current Camera Control Subsystem.

Data Type: String

Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script		varies by product	ROM

ccsv – CCS Subsystem Version

Definition: The version of the current Camera Control Subsystem.

Data Type: UInt

Access: Read only

Associated File Tag:ccsv

Product/OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CCDD. Values are BCD, from 0 to 9, a to f. Leading zeros within group (AA, for example) are suppressed. For example, version 2.0.3.a would appear as 0x0200030a.	varies by product	ROM

fwcn – Digita Copyright Notice

Definition: Specifies copyright for the Digita firmware

Data Type: String

Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script		(c) 1999 FlashPoint Technology, Inc.	ROM

fwrv – Digita Firmware Version

Definition: Specifies Digita firmware version

Data Type: UInt

Access: Read only

Associated File Tag:fwrv

Product/OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CCDD. Values are BCD, from 0 to 9, a to f. Leading zeros within group (AA, for example) are suppressed. For example, version 2.0.3.a would appear as 0x0200030a.	varies by product	ROM

hprs – Host PRAM Size

Definition: Number of EEPROM bytes available for use by Digita Desktop

Data Type: UInt

Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script		32	ROM

hwrv – Hardware Version

Definition: The version number for the hardware

Data Type: UInt

Access: Read only

Associated File Tag:hwrv

Product/OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CCDD. Values are BCD, from 0 to 9, a to f. Leading zeros within group (AA, for example) are suppressed. For example, version 2.0.3.a would appear as 0x0200030a.	varies by product	EEPROM

ipcn – IPM Copyright Notice

Definition: Copyright for the image processing chain.

Data Type: String

Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script		varies by product	ROM

iprn – IPM Subsystem Version

Definition: Version number for the image processing chain.

Data Type: UInt

Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CCDD. Values are BCD, from 0 to 9, a to f. Leading zeros within group (AA, for example) are suppressed. For example, version 2.0.3.a would appear as 0x0200030a.	varies by product	ROM

pcnt – Picture Count Taken

Definition: The number of pictures taken by the camera

Data Type: UInt

Access: Read only

Associated File Tag:pcnt

Product/OS	Details	Factory Default	Stored In
Digita Script	Shipped from the factory with a value of 0. Value is incremented every time Digita successfully takes a picture. Maximum value of 1048575.	0	EEPROM

ptid – Product Name

Definition: Name of the product

Data Type: String

Access: Read only

Associated File Tag:ptid

Product/OS	Definition	Factory Default	Stored In
Kodak DC 220	Product name	KODAK DIGITAL SCIENCE DC220	ROM
Kodak DC 260		KODAK DIGITAL SCIENCE DC260	ROM
Kodak DC 265		KODAK DC265 ZOOM DIGITAL CAMERA	ROM
Kodak DC 290		KODAK DC290 Zoom Digital Camera	ROM
Dimage EX 1500		Dimage EX	ROM

rams – Application RAM Size

Definition: Amount of RAM available for an alternate application

Data Type: UInt

Access: Read only

Product/OS	Details	Factory Default	Stored In
Kodak DC 220	Application RAM size in bytes (e.g., 4194304 is 4MB, and 6291456 is 6MB).	4194304	ROM
Kodak DC 260		6291456	ROM
Kodak DC 265		16777216	ROM
Kodak DC 290		12500000	ROM
Dimage EX 1500		4194304	ROM

scnt – Strobe Pictures Taken

Definition: Number of strobe images captured with this camera.

Data Type: UInt

Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	Shipped from the actory with a value of 0. Count is incremented every time Digita successfully takes a flash picture. Maximum value of 1048575.	0	EEPROM

sern – Camera Serial Number

Definition: A unique, factory set number for each device.

Data Type: UInt

Access: Read only

Asssociated File Tag:sern

Product/OS	Details	Factory Default	Stored In
Digita Script	Maximum length of 32 bits.		
Kodak DC 220/DC 260/DC 265	Maximum length of 24 bits.	XXXXXXXX	EEPROM
Dimage EX 1500	Maximum length of 24 bits.	XXXXXXXX	EEPROM

ucrv – Universal Command Set Version

Definition: The version number of the host command set.

Data Type: UInt - hex

Access: Read only

Product OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CCDD. Values are BCD, from 0 to 9, a to f. Leading zeros within group (AA, for example) are suppressed. For example, version 2.0.3.a would appear as 0x0200030a.	0x01000000	ROM

vdid – Vendor Name

Definition: The name of the company selling or making the product.

Data Type: String

Access: Read only

Associated File Tag:vdid

Product/OS	Details	Factory Default	Stored In
Kodak DC 220/ DC 260/DC 265/ DC 290	Vendor name.	Eastman Kodak Company	ROM
Dimage EX 1500	Vendor name.	Minolta Co., Ltd.	ROM

Appendix B Camera Capabilities Parameters

The following tables define the camera capability parameters (supplied in the PName parameter) for the Digita Operating System. The tables also detail the manufacturer defined operational differences for products currently supporting Digita scripts.

Note Be sure to verify the values in these tables, as different versions of camera software may have different defaults and ranges. For scripts that are generalized to run across different cameras (or different versions of the same camera), be sure to use the commands `GetCapabilityType`, `GetCapabilitiesRange`, `GetCapabilitiesCount`, `GetCapabilitiesListItem` and `GetCapabilitiesValue` to determine the parameter type and value.

Note Specific products are listed only when their details or defaults differ from the current Digita Script standard.

aagc – Analog Signal Gain Control

Definition: Sets the analog signal gain from the image sensor. Used in low light situations with the Programmed and Gain Priority exposure modes only (see xmod parameter).

Data Type: UInt

Access: Read/Write

Associated File Tag: aagc

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=100, Maximum=400, Units=ASA rating	100	EEPROM
DC 290	Minimum=100, Maximum=100	100	EEPROM
DC 220/DC 260/ DC 265	Tag not supported.		EEPROM
Dimage EX 1500	Minimum=600, Maximum=2400 (version 2.0 only)	800	EEPROM

acpd – AC Power Down Timeout

Definition: Sets the amount of time to sleep on AC power before powering down.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=5, Maximum=65535, Units=seconds	600	EEPROM
Dimage EX 1500		1800	EEPROM

acse – AC Sleep Timeout Enable

Definition: Indicates whether or not the AC sleep timeout is enabled.
 Data Type: Enum List
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Enable, 2=Disable, 3=Never Sleep	1	EEPROM
Kodak DC 290		2	EEPROM
Kodak DC 220/ DC 260/ DC 265	The value zero (0) indicates disabled. This parameter is always disabled for this camera.	0	EEPROM
Dimage EX 1500		0	EEPROM

actc – AC Sleep Timeout

Definition: Sets the number of seconds after no user activity and no image processing for the camera to go to sleep when using AC power. There is no sleep cycle if the acse parameter is set to off
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=5, Maximum =65535, Units=seconds	600	EEPROM

aelk – AE Lock Mode

Definition: AE Lock: Pressing the shutter release button partway down holds the exposure settings until the the image is captured or pressure is removed from the shutter release button.

Continuous: The exposure is constantly adjusted until the image is captured.

Data Type: Enum

Access: Read/Write

Associated File Tag: lkmd

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Continuous, 2=AE Lock	1	EEPROM
Kodak DC 220/ DC 265/DC 265/ DC 290		2	EEPROM
Dimage EX 1500		2	EEPROM

aflk – AF Lock Mode

Definition: AF Lock: Pressing the shutter release button partway down holds the focus settings until the the image is captured or pressure is removed from the shutter release button.

Continuous: Focus is constantly adjusted until the image is captured.

Data Type: Enum

Access: Read/Write

Associated File Tag: lkmd

Product	Details	Factory Default	Stored In
Digita Script	Continuous=1, AF Locked=F=2		EEPROM
Kodak DC 290	Continuous AF not supported.	2	EEPROM
Dimage EX 1500	Continuous AF not supported	2	EEPROM

aper – Aperture (F Number)

Definition: Aperture setting in F number in 0.01 step .
 Data Type: UInt
 Access: Read/Write
 Associated File Tag: aper (aper file tag not supported by the Kodak DC 220/260/265 and the Minolta Dimage EX 1500. Use the fnum file tag instead.)

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=200, Maximum=3200, Units= 0.01 (eg. F2.8 = 280).	300	EEPROM
Kodak DC 290	Minimum= 300, Maximum=1600,	560	EEPROM
Kodak DC 260/ DC 265	Minimum= 300, Maximum=1600	300	EEPROM
Kodak DC 220	Aperture is fixed at F4.0.	400	EEPROM
Dimage EX 1500	Aperture is fixed at F5.6.	560	EEPROM

bccl – Burst Capture Color Mode

Definition: Sets the color mode for burst capture
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: impn

Product/OS	Details	Factory Default	Stored In
Digita Script	1=24 bit color, 2= Gray scale, 3=Tone If tone is selected, icts parameter specifies the image tone.	1	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Gray scale (2) and Tone (3) not supported	1	EEPROM
Dimage EX 1500	Tone (3) not supported	1	EEPROM

bcpn – Burst Capture Compression

Definition: Sets the compression mode for burst capture images. Higher compression means lower image quality.

Data Type: Enum List

Access: Read/Write

Associated File Tag: cmpn

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Maximum, 2=High, 3=Normal, 4=Low, 5=Very Low, 6= Minimum, 7=Lossless	2	EEPROM
Kodak DC290	Settings 1, 5, and 6 not supported.	4	
Kodak DC 220/ DC 260	Settings 1, 5, 6 and 7 not supported.	4	EEPROM
Kodak DC 265	Settings 1, 6 and 7 not supported	4	EEPROM
Dimage EX 1500		4	EEPROM

bsiz – Burst Capture Size

Definition: Sets the size of the image for burst capture.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Full Size, 2=1/4 Size, 3=1/16 Size, 4=Video Size (640x480) and 5=LCD Size	2	EEPROM
Kodak 290	Setting 4 corresponds to Ultra (2,240x1,500). Setting 5 not supported	3	EEPROM
Kodak DC 260/ DC 265	Settings 4 and 5 not supported.	3	EEPROM
Kodak DC 220	Settings 2, 4 and 5 not supported.	3	EEPROM
Dimage EX 1500	Settings 3, 4 and 5 not supported.	1	EEPROM

btspd – Battery Power Down Timeout

Definition: Sets the number of seconds to sleep on battery power before powering down. (btse must be enabled)

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum: 5, Maximum: 65535, Units: second	480	EEPROM
Dimage EX 1500		360	EEPROM

btse – Battery Sleep Timeout Enable

Definition: Enables or disables the btsc parameter.

Data Type: Enum

Access: Read/Write

btse	Definition	Factory Default	Stored In
Digita Script	1=Enable, 2=Disable	1	EEPROM
Kodak DC 220/ DC260/DC 265/ DC 290	Disabled value not supported	1	EEPROM
Dimage EX 1500	Disabled value not supported	1	EEPROM

btsc – Battery Sleep Timeout

Definition: Sets the number of seconds of inactivity before the camera enters sleep, mode. This parameter is not applied to Capture mode when the LCD is off. (btse must be enabled)

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum: 5, Maximum: 65535, Units: seconds	120	EEPROM
Kodak DC 290		60	EEPROM
Dimage EX 1500		180	EEPROM

camf – Camera Folder Name

Definition: The name of the folder where images are stored.
 Data Type: DOS Filename
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Up to 8 characters can be entered. Only the first 6 characters are used. Numbers 01, 02,.....,99 are appended, as needed.	CAMERA	EEPROM
Kodak DC 290		DC290_	EEPROM
Kodak DC 265		DC265_	EEPROM
Kodak DC 260	The DOS name of the camera folder. This value can only be set to the factory default value.	DC260_	EEPROM
Kodak DC 220		DC220_	EEPROM
Dimage EX 1500		EX15_	EEPROM

cmne – User-defined Camera Name

Definition: The name used to reference the camera from Digita Desktop
 Data Type: String
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script		Digita Camera	EEPROM
Kodak DC 290	A maximum of 31 characters can be entered	DC290	EEPROM
Kodak DC 265		DC265	EEPROM
Kodak DC 260		DC260	EEPROM
Kodak DC 220		DC220	EEPROM

cppt – User Copyright Notice

Definition: User entered copywrite notice. Parameter is stored in every image file.
 Data Type: String
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Tag not supported. Replaced by ucpt.		EEPROM
Kodak DC 220/ DC 260/DC 265	A maximum of 31 characters can be entered	""	EEPROM
Dimage EX 1500		""	EEPROM

cpus – Processor Speed

Definition: Specifies the speed of the processor. Defined by hardware
 Data Type: UInt
 Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	Units: 0.01 MHz	4909	ROM
Kodak DC 290		6600	ROM
Kodak DC 220/ DC 260/DC 265	Tag not supported		
Dimage EX 1500	Tag not supported		

cpup– Processor Type

Definition: Specifies the speed of the processor. Defined by hardware
 Data Type: Enum
 Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	1=MPC823 (Motorola 823), 2=DCAM 101/103	1	ROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

crst– Image Counter Mode

Definition: Sets the counter to roll or odometer mode.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Roll Count, 2=Odometer Count	2	EEPROM
Kodak DC 290		1	EEPROM

deft– Short Date Format

Definition: Specifies the order of entry for dates in the Date Entry screens and other uses of the Short Date Format, including watermarks. Refer to “Date and Time Formats” starting on page 147.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD	1	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

dfmt – Date Display Format

Definition: Specifies the order of entry for dates in the Date Entry screens and other uses of the Short Date Format, including watermarks.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=short date, 2=abbreviated date, 3=long date	1	EEPROM

drdn – User RAMDisk Name

Definition: Defines the internal RAMDisk volume name, which will appear on the desktop if direct mounting is supported by Digita Desktop.

Data Type: DOS Filename

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Not currently supported by any Digita-enabled product	RAMDISK	

ebrs – Enable Built-in Scripts

Definition: Allows hidden scripts to be built into the system. Scripts remain hidden unless enabled via a script.

Data Type: BitFlags

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Not currently supported by any Digita-enabled product	0x0	

exsa – Expansion Slots Available

Definition: Specifies the number of expansion slots available.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script		1	ROM

fbti – Burst Time Interval

Definition: Specifies the time interval between captures for burst capture.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Miniimum=100, Maximum=1000	250	EEPROM
Kodak DC 290	Miniimum=500, Maximum=10000	500	EEPROM
Kodak DC 260/ DC 265	Minimum=-320 , Maximum=10000.	360	EEPROM
Kodak DC 220	Miniimum=320, Maximum=10000	320	EEPROM
Dimage EX 1500	minimum=30, Maximum=10000.	266	EEPROM

fdst – Focus Distance

Definition: Specifies the focus distance for Programmed focus mode. Set by the fmod parameter.
 Data Type: UInt
 Access: Read/Write
 Associated File Tag: fdst

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=30, Maximum (infinity) =-65535	1000	EEPROM
Kodak DC 260/ DC 265/DC 290	Minimum=50, Maximum (infinity) =-65535	300	EEPROM
Kodak DC 220	Fixed Focus Camera - Tag not supported.		EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

flty – File Type

Definition: Specifies the file type when capturing still images.
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: flty

Product/OS	Details	Factory Default	Stored In
Digita Script	1=JPEG, 2=EXIF, 3=FPIX, 4=TIFF	2	EEPROM
Kodak DC 290	2=EXIF, 4=TIFF	2	
Kodak DC 220/ DC 260/DC 265	FPIX=0x46505820, JPEG=0x4A504547. Associated file tag: imcn	0x4A504547	EEPROM
Dimage EX 1500	JPEG=0x4A504547. Associated file tag: imcn	0x4A504547	EEPROM

fmod – Focus Mode

Definition: Specifies the camera's focus mode.
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: fmod

Product	Definition	Factory Default	Stored In
Digita Script	1=Auto, 2=Program, 3=Manual	1	RAM
Kodak DC 260/ DC 265/DC 290	Setting 2 not supported.	1	RAM
Dimage EX 1500	Setting 2 not supported.	1	RAM
Kodak DC 220	Tag not supported		RAM

fmtd – Focus Method

Definition: Specifies the scene data to be used to compute the focus point.
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: fmtd

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Auto Matrix, 2=Center Spot, 3=Multi Spot, 4=Center Weighted, 5=Custom Matrix	1	RAM
Kodak DC 260/ DC 265/DC 290	Settings 3,4 and 5 are not supported.	1	RAM
Dimage EX 1500	Settings 1,3,4 and 5 not supported.	2	RAM
Kodak DC 220	Tag not supported.		RAM

fwrdd – Digita Release Date

Definition: Specifies the release date for the installed version of the Digita OE.
 Data Type: UInt
 Access: Read/ only

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum= 0, Maximum=0xFFFFFFFF, Units=seconds		ROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

hint – Hint Mode

Definition: Preferences the AE and AF systems for the type of subject being captured to improve the exposure and focus.

Data Type: Enum List

Access: Read/Write

Associated File Tag: hint

Product/OS	Definition	Factory Default	Stored In
Digita Script	1=Auto, 2=Portrait, 3=landscape, 4=Closeup, 5=Sports, 6=Evening , 7=Night and 8=sSow-Sync, 9=Panorama	1	RAM
Dimage EX 1500	Settings 4 and 5 are not supported. Setting 9 supported by the 3D1500 only.	1	RAM
Kodak DC 220/ DC 260/DC265	Tag not supported.		

icts – Image Capture Tone

Definition: Sets the tone (CrCb)value to be applied to the image.

Data Type: UInt

Access: Read/Write

Associated File Tag: icts

Product/OS	Details	Factory Default	Stored In
Digita Script	Not currently supported by any Digita-enabled product.	23609	

ihtc – Image Head Timeout

Definition: Specifies how much time camera subsystem continues AF/AE/AWB operation after S1 is released, when the LCD Finder is off, i.e., when optical viewfinder is being used.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum:0, Maximum=30, Units=seconds	8	EEPROM
Kodak DC 290	Minimum:0, Maximum=63	0	EEPROM
Kodak DC 220/ DC 260/DC 265	Minimum:0, Maximum=63	8	EEPROM
Dimage Ex 1500	Minimum:0, Maximum=63	8	EEPROM

iirv – Image Info Version

Definition: Specifies the version number used for image information file tags.

Data Type: UInt

Access: Read only

Associated File Tag: iirv

Product/OS	Details	Factory Default	Stored In
Digita Script	Format = 0xAABBCCDD for version AA.BB.CC.DD. Values are extended BCD, providing digits 0 to 9 and letters 'a' through 'f'. Leading zeros within a group are suppressed. Groups CC and DD will also be suppressed entirely if they are zero. For example, 0x0300020a would appear as version 3.0.2.a, 0x00090100 as version 0.9.1, and 0x02010000 would cause the version 2.1 to be displayed.	0x01000100	ROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage Ex 1500	Tag not supported.		

irev – Instant Review Duration

Definition: Sets the duration of Instant Review after capture.

Data Type: UInt

Access: Read /Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=3000, Units=0.01 seconds	300	EEPROM
Dimage Ex 1500	Minimum=0, Maximum=10000	1000	EEPROM

isam – Image Sharpness Amount

Definition: Controls sharpening performed when the image is captured.

Data Type: Signed Int

Access: Read /Write

associated File Tag: isam

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-25, Maximum=100 0 indicates no sharpening, negative numbers - if supported - are softening (unsharpen), and positive numbers indicate sharpening. Can be used in conjunction with other sharpen parameters for full unsharp mask functionality, or used alone for simple sharpen algorithms.	0	EEPROM
Kodak DC 290	Minimum=-25, Maximum=100	50	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage Ex 1500	Tag not supported.		

isra – Image Sharpness Radius

Definition: Sets sharpening radius for capture.
 Data Type: Fixed Point Range
 Access: Read /Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0.1, Maximum=10, Units=0.1 pixels	0.1	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage Ex 1500	Tag not supported,		

isth – Image Sharpness Threshold

Definition: Sets sharpening threshold for capture.
 Data Type: UInt
 Access: Read /Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=25, Units=pixels	4	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage Ex 1500	Tag not supported.		

mcap – Media Type Capture Mode

Definition: Sets the system to capture the specified media type.
 Data Type: Enum
 Access: Read /Write
 Associated File Tag: mcap

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Still, 2=Burst, 3=Timelapse	1	RAM
Kodak DC 220/ DC 260/DC 265	The media type capture mode, where 0=still, 1=burst and 2=timelapse.	0	RAM
Dimage EX 1500	The media type capture mode, where 0=still, 1=burst and 2=timelapse.	0	RAM

mcro – Macro Mode

Definition: Controls whether the optical system is in Macro mode.
 Data Type: Enum
 Access: Read /Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Normal, 2=Macro Focus distance should accurately reflect the macro distance (see the 'fdst' parameter).	1	EEPROM
Kodak DC 260/ DC 265/DC 290	Setting 2 not supported.	1	EEPROM

mhbs – Maximum Host Buffer Size

Definition: Specifies the largest buffer required for transferring images on the host size. Host can set this value.
 Data Type: UInt
 Access: Read /Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=10240, Maximum=32768, Units=Bytes	19456	EEPROM

mtdy – Self Timer Delay

Definition: Sets the time delay between the shutter button being fully depressed by the user and the picture being taken.
 Data Type: UInt
 Access: Read /Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1, Maximum=60, Units=seconds	10	EEPROM

oerv – Digita OE Version

Definition: Specifies version of the Digita OE currently running.
 Data Type: U hex
 Access: Read only

Product/OS	Details	Factory Default	Stored In
Digita Script	Format is 0xAABBCCDD for version AA.BB.CC.DD. Values are extended BCD, providing digits 0 to 9 and letters 'a' through 'f'. This parameter does not indicate the revision of any software specific to the hardware device the OE is running upon (see parameters 'carv', 'ccsv', 'fwrv' and 'iprn').	0x01000000	ROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage Ex 1500	Tag not supported.		

pasw – User Password

Definition: Used to lock out functions in scripts.
 Data Type: String
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Maximum length of 31 characters. Case sensitive.	""	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage Ex 1500	Tag not supported.		

pbbr – Burst Playback Interval

Definition: Sets the playback rate for burst captured data sets.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1000, Maximum=30000, Units= 0.01 seconds	5000	EEPROM
Kodak DC 290	Minimum=1000, Maximum=30000	3000	EEPROM
Kodak DC 220/ DC 260/DC 265	Minimum=1 Maximum=10	3	EEPROM
Dimage EX 1500	Units=seconds	3	EEPROM

pbgr – Group Playback Interval

Definition: Sets the playback rate for user grouped data sets.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1000, Maximum=30000, Units= 0.01 seconds	5000	EEPROM
Kodak DC 290	Minimum=1000, Maximum=30000	3000	EEPROM
Kodak DC 220/ DC 260/DC 265	Minimum=1 Maximum=10	3	EEPROM
Dimage EX 1500	Units=seconds	3	EEPROM

pbrr – Print Brightness

Definition: Sets the relative brightness of the image.

Data Type: Int

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-25, Maximum=25, Units= percent Not currently supported by any Digita enabled device.	0	

pbtr – Timelapse Playback Interval

Definition: Sets the playback rate for Timelapse captured data sets.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1000, Maximum=30000, Units= 0.01 seconds	5000	EEPROM
Kodak DC 290	Minimum=1000, Maximum=30000	3000	EEPROM
Kodak DC 220/ DC 260/DC 265	Minimum=1 Maximum=10	3	EEPROM
Dimage EX 1500	Units=seconds	3	EEPROM

pchc – Patch Count

Definition: Specifies how many patches were applied during boot-up.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script		0	ROM
Kodak DC 220/ DC 260/DC265	Tag not supported.	0	

pcon – Print Contrast

Definition: Specifies relative adjustment of contrast.
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Not currently supported by any Digita enabled device.	0	

pcpy – Print Copy Count

Definition: Specifies the number of copies to print.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1, Maximum=50 Units=Copies Not currently supported by any Digita-enabled device.	1	

pgms – Program Shift

Definition: Used to 'bias' the exposure toward larger or smaller apertures.
 Data Type: Int
 Access: Read/Write
 Associated File Tag: pgms

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-300, Maximum=300, Units=0.01EV Not currently supported by any Digita-enabled device.	0	

play – Print Layout

Definition: Sets the default page layout. This can be overridden by a template.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=1 up, 2=2 up, 3=3up, 4=4 up,5=6up, 6=20 up Not currently supported by any Digita-enabled device.	1	

port – Print Paper Orientation

Definition: Sets the orientation of the paper in the printer.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Portrait, 2=Landscape Not currently supported by any Digita-enabled device.	1	

posh – Print Horizontal Shift

Definition: Adjusts the horizontal offset for printing.
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-10, Maximum=10, Units=pixels Not currently supported by any Digita-enabled device.	0	

posv – Print Vertical Shift

Definition: Adjusts the vertical offset for printing.
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-10, Maximum=10, Units=pixels Not currently supported by any Digita-enabled device.	0	

pqly – Print Quality

Definition: Specifies the quality of print vs. speed of print..
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Best, 2=Better, 3=Good Not currently supported by any Digita-enabled device.	1	

pres – Print Resize

Definition: Sets the method used for filling in the print rectangles specified by the “play” parameter.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Fit wiithin, 2=Fill and clip, 3=Stretch and fill Not currently supported by any Digita-enabled device.	1	

psam – Print Sharpness

Definition: Sets sharpening for print.
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum= -25, Maximum= 100 0 is no sharpening, negative numbers - if supported - are softening (unsharpen), and positive numbers increase sharpening. Not currently supported by any Digita-enabled device.	0	

psat – Print Saturation

Definition: Sets the color saturation for the print.
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum= -25, Maximum= 25 Units=percent 0 is no change, negative numbers reduce saturation, and positive numbers increase saturation. Not currently supported by any Digita-enabled device.	0	

psel – Printer Selection

Definition: Sets the destination printer.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Enumeration list contents set by available drivers. Not currently supported by any Digita-enabled device.	1	

psiz – Print Paper Size

Definition: Specifies the paper size for the printer.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=US Letter, 2=US Legal, 3=A2, 4=A3, 5=A4, 6=4X6, 7=Hagaki Not currently supported by any Digita-enabled device.	1	

psra – Print Sharpness Radius

Definition: Sets sharpening radius for print. Standard “unsharp mask” functionality.
 Data Type: Fixed Point Range
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0.1, Maximum=10, Units=pixels Not currently supported by any Digita-enabled device.	0.1	

psth – Print Sharpness Threshold

Definition: Sets sharpening threshold for print. Standard “unsharp mask” functionality.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=25 Not currently supported by any Digita-enabled device.	4	

ptyp – Print Paper Type

Definition: Specifies what kind of paper is being used in the printer.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Plain Paper, 2=Coated Paper, 3=Glossy Paper, 4=Transparency Not currently supported by any Digita-enabled device.	1	

qbti – Quarter Burst Time Interval

Definition: Specifies the time interval between captures for 1/4 size burst capture.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=100, Maximum=1000, Units=0.01 seconds	250	EEPROM
Kodak DC 260/ DC 265/DC 290	Minimum=360, Maximum=10000.	360	EEPROM
Kodak DC 220	Minimum=80, Maximum=10000	80	EEPROM
Dimage EX 1500	Minimum=30, Maximum=10000	266	EEPROM

rdsz – RAM Disk Size

Definition: Sets the size of the internal RAM disk (in sectors).
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=8192 Not currently supported by any Digita-enabled device.	0	

rgnc – Region Code

Definition: Specifies the current language selection used in the product.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=US English, 2=UK English, 3=French, 4=Italian, 5=Spanish, 6=German, 7=Swedish, 8=Japanese, 9=Dutch	1	EEPROM
Kodak 290	Settings 1, 3, 6, and 8 supported	1 or 8	
Kodak DC 220/ DC260/DC 265/	Settings 1 and 8 supported.	1 or 8	EEPROM
Dimage EX 1500	Settings 1, 3, 6, and 8 supported	depends on region	EEPROM

rmod – Auto Rotate Image

Definition: Enables or disables automatic image rotation (if available).

Data Type: Enum

Access: Read/Write

Associated File Tag: rmod

Product/OS	Details	Factory Default	Stored In
Digita Script	0=Off, 1=On	1	EEPROM
Kodak DC 220/ DC 260/DC 265	1=Auto and 2=Landscape.	1	EEPROM
Dimage EX 1500	Always ON, Off not supported.	1	EEPROM

saim – Script Activity Mask

Definition: Used by scripts to keep track of parameter changes which could affect product operation.

Data Type: BitFlags

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=0xFFFFFFFF Not currently supported by any Digita-enabled device.	0	

sccl – Still Capture Color Mode

Definition: controls the color mode used for still capture.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=24 bit color, 2=Gray scale, 3=Tone If 3 (Tone) is selected, the icts parameter specifies the image tone.	1	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Settings 2 and 3 not supported.	1	EEPROM
Dimage EX 1500	Setting 3 not supported.	1	EEPROM

scmp – Strobe Compensation

Definition: Biases the amount of additional light contributed to the exposure by the the automatic strobe

Data Type: Int

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-200, Maximum=200, Units=0.01EV.	0	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		

scpn – Still Capture Compression

Definition: Sets the compression level for still capture images.

Data Type: Enum

Access: Read/Write

Associated File Tag: cmpn

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Maximum, 2=High, 3=Normal, 4=Low, 5=Very Low, 6=Minimum, 7=Lossless	3	EEPROM
Kodak DC 290	Settings 1, 5 and 6 not supported	4	EEPROM
Kodak DC 220/ DC 260DC 265	Settings 1,5, 6 and 7 not supported	4	EEPROM
Dimage EX 1500	Settings 1, 6 and 7 not supported	3	EEPROM

shut – Shutter Speed

Definition: Controls the shutter speed in the Shutter Priority and Programmed exposure modes (see the xmod parameter).

Data Type: UInt

Access: Read/Write

Associated File Tag: shut

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1000, Maximum=1000000, Units=0.000001 seconds (500000=0.5 seconds)	10000	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Minimum=500000, Maximum=16000000, Units=0.000001 seconds (500000=0.5 seconds)	500000	EEPROM
Dimage EX 1500	Minimum=250, Maximum=2000000, Units=0.000001 seconds (500000=0.5 seconds)	62500	EEPROM

smod – Strobe Mode

Definition: Sets the operating mode for the strobe.

Data Type: Enum

Access: Read/Write

Associated File Tag: smod

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Auto -Built-in flash fires when necessary. 2=Fill -Built-in flash fires in all situations. 3=Off -Built-in flash wil not fire. 4=Slave -Built-in flash is used to trigger flash units not attached to the camera. 5=Sync -triggers a PC connected strobe (built-in flash will not fire).	1	EEPROM
Kodak DC 260/ DC 265/DC 290	Settings 1, 2, 3 and 5 supported.	2	EEPROM
Kodak DC 220	Settings 1, 2 and 3 supported.	2	EEPROM
Dimage EX 1500	Settings 1, 2, and 3 supported.	2	EEPROM

spre – Strobe Precount

Definition: Specifies the number of preflashes fired prior to the “main” exposure flash. Used to reduce the red-eye effect.

Data Type: UInt

Access: Read/Write

Associated File Tag: spre

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum =0, Maximum=3	0	EEPROM
Kodak DC 220/ DC 260/DC 265	0=Off, 1=On Strobe preflashes according to manufacturer specifications.	0	EEPROM
Dimage EX 1500		0	EEPROM

sscs – Slideshow Category String

Definition: Specifies the category used to filter displayed images for the ssct parameter.

Data Type: String

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	ssct parameter must be set to category	first category	EEPROM

ssct – Slideshow Content

Definition: Selects whether all images will be shown during the slide show, or if they will be filtered by category.

Data Type: Enum

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Category, 2=All Category set in sscs parameter	2	EEPROM

ssdr – Slide Show Frame Interval

Definition: Sets the image display time for a slide shows without a timetrack.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1, Maximum=99, Units=seconds.	5	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290		3	EEPROM
Dimage EX 1500	Minimum=1, Maximum=10	3	EEPROM

ssis – Slide Show Image Sound

Definition: Turns image sound annotations on and off.

Data Type: Enum

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	0=OFF, 1=ON	1	EEPROM
Dimage EX 1500	Always off. On not supported	0	EEPROM

ssiz – Still Capture Size

Definition: Specifies the image size for still capture.
 Data Type: Enum List
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Full Size, 2=1/4 Size, 3=1/16 Size, 4=Video Size (640x480) and 5=LCD Size	1	EEPROM
Kodak DC 290	Setting 4 corresponds to Ultra (2,240x1,500). Setting 5 not supported	1	EEPROM
Kodak DC 260/ DC 265	Settings 4 and 5 not supported	1	EEPROM
Kodak DC 220	Settings 2, 4 and 5 not supported	1	EEPROM
Dimage EX 1500	Settings 3, 4 and 5 not supported	1	EEPROM

sslp – Slide Show Loop Indicator

Definition: Specifies ON or Off for automatic looping.
 Data Type: Enum List
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	0=Off, 1=On	0	EEPROM

ssnd – System Sound Selector

Definition: Selects from a set of system sounds, which can be loaded via disk or installed during build. (Older tag definition-Shutter Sound Spec- turned shutter sound On and Off.)

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1, Maximum=10 Any loaded sound can be accessed for any sound function, i.e., it is table driven. A given event generates a request for a sound, and the table points to correct resource, or is null. Mechanism allows user to create sets of sounds, including use of existing sounds, and download to camera. Parameter selects which set to use.	1	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Tag not supported.		
Dimage EX 1500	0=ON	0	EEPROM

ssvl – System Sound Volume

Definition: Specifies the volume of the system sounds. (Older spec. specified the volume of the shutter sound only.)

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0 (OFF), Maximum=7	7	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Minimum=0, Maximum=1	1	EEPROM
Dimage EX 1500	Minimum=0, Maximum=1	1	EEPROM

stim – Self Timer Enable

Definition: Enables the self timer mode.
 Data Type: Enum List
 Access: Read/Write

Product/OS	Definition	Factory Default	Stored In
Digita Script	0=OFF, 1=ON The amount of delay is set by the mtdy parameter	0	RAM

tccl – Timelapse Capture Color Mode

Definition: Controls the color mode used for timelapse capture.
 Data Type: Enum List
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=24 bit color, 2=Gray scale, 3=Tone If 3 (Tone) is selected, the icts parameter specifies the image tone.	1	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Settings 2 and 3 not supported.	1	EEPROM
Dimage EX 1500	Setting 3 not supported.	1	EEPROM

tcnt – Timelapse Capture Count

Definition: Sets the maximum number of images to capture for a timelapse.
 Data Type: Int
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=-2, Maximum=1000, Units=Images	10	EEPROM
Dimage EZ1500	Minimum=-2, Maximum=200.	10	EEPROM

tcpn – Timelapse Capture Compression

Definition: Sets the compression level for timelapse capture images.

Data Type: Enum List

Access: Read/Write

Associated File Tag: cmpn

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Maximum, 2=High, 3=Normal, 4=Low, 5=Very Low, 6=Minimum, 7=Lossless	3	EEPROM
Kodak DC 290	Settings 1, 5 and 6 not supported	4	EEPROM
Kodak DC 220/ DC 260DC 265	Settings 1,5, 6 and 7 not supported	4	EEPROM
Dimage EX 1500	Settings 1, 6 and 7 not supported	3	EEPROM

tfmt – Time Format

Definition: Time format used for entering and editing time values. Used for watermarks and some display screens.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=hh:mm am/pm, 2=hh:mm:ss am/pm, 3=hh:mm, 4=hh:mm:ss	1	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	The time format, where 1=hour/minute and 2=hour/minute/second.	1	EEPROM
Dimage EX 1500	The time format, where 1=seconds and 2=minutes.	1	EEPROM

tldy – Timelapse Capture Interval

Definition: Specifies the time interval between captures.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1500, Maximum=8640000, Units=0.01seconds.	6000	EEPROM
Kodak DC 290	Minimum=6000, Maximum=8640000, Units=0.01seconds.	6000	EEPROM
Kodak DC 220/ DC 260/DC 265	Minimum=60, Maximum=86400, Units=seconds.	60	EEPROM
Dimage EX 1500	Minimum=60, Maximum=86400, Units=seconds.	60	EEPROM

tsiz – Timelapse Capture Size

Definition: Sets the image size for timelapse Captures.
 Data Type: Enum List
 Access: Read/Write

Product	Details	Factory Default	Stored In
Digita Script	1=Full Size, 2=1/4 Size, 3=1/16 Size, 4=Video Size (640x480) and 5=LCD Size	1	EEPROM
Kodak DC 290	Setting 4 corresponds to Ultra (2,240x1,500). Setting 5 not supported	1	EEPROM
Kodak DC 260/ DC 265	Settings 4 and 5 not supported	1	EEPROM
Kodak DC 220	Settings 2, 4 and 5 not supported	1	EEPROM
Dimage EX 1500	Settings 3, 4 and 5 not supported	1	EEPROM

ucpt – User Copyright Notice

Definition: Stored in every image file. It allows the user to tag his/her images with a personal copyright notice.

Data Type: String

Access: Read/Write

Associated File Tag: ucpt

Product/OS	Details	Factory Default	Stored In
Digita Script		""	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported. Use cprt instead.		
Dimage EX 1500	Tag not supported. Use cprt instead.		

vmod – Video Output Mode

Definition: Sets the video output mode.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=NTSC, 2=PAL, 3=SECAM	1	EEPROM
Kodak DC 290	Setting 3 not supported	1	EEPROM
Kodak DC 220/ DC 260/DC 265	0=NTSC, 1=PAL	0	EEPROM
Dimage EX 1500	0=NTSC, 1=PAL	0	EEPROM

wbcb – White Balance Blue

Definition: Controls the value for the blue component when the wmod parameter is set to Custom.

Data Type: Fixed Point Range

Access: Read/Write

Associated File Tags: wbcb

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1.0, Maximum=3.99	1.0	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Minimum =0.25 (expressed as 0x00004000) Maximum=4.00 (expressed as 0x00040000)	1.0 (0x00010000)	EEPROM
Dimage EX 1500	Minimum=-0x4000, Maximum=0x40000	0x10000	EEPROM

wbcg – White Balance Green

Definition: Controls the value for the green component when the wmod parameter is set to Custom.

Data Type: Fixed Point Range

Access: Read/Write

Associated File Tags: wbcg

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1.0, Maximum=3.99	1.0	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Minimum =0.25 (expressed as 0x00004000) Maximum=4.00 (expressed as 0x00040000)	1.0 (0x00010000)	EEPROM
Dimage EX 1500	Minjimium=-0x4000, Maximum=0x40000	0x10000	EEPROM

wbcr – White Balance Red

Definition: Controls the value for the red component when the wmod parameter is set to Custom.

Data Type: Fixed Point Range

Access: Read/Write

Associated File Tags: wbcr

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=1.0, Maximum=3.99	1.0	EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Minimum =0.25 (expressed as 0x00004000) Maximum=4.00 (expressed as 0x00040000)	1.0 (0x00010000)	EEPROM
Dimage EX 1500	Minjimum=-0x4000, Maximum=0x40000	0x10000	EEPROM

wena – Watermark Enable

Definition: Sets which kind of watermark will be applied to an image during image capture processing.

Data Type: BitFlags

Access: Read/Write

Product OS	Details	Factory Default	Stored In
Digita Script	Minimum=0b0000, Maximum=0b1111 Bit functions are: Logo enable (0b1000), Text enable (0b0100), Time enable (0b00!0), Date enable (0b0001).	0b0000	EEPROM
Dimage EX 1500	Tag not supported.		

wmod – White Balance Mode

Definition: Sets the mode for white balance during image capture.

Data Type: Enum List

Access: Read/Write

Associated File Tag: wmod

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Auto, 2=Bright Sun, 3=Daylight, 4=Dim Sun, 5=Cloudy, 6=US Fluorescent, 7=Japan Fluorescent, 8=UK Fluorescent, 9=Tungsten, 10=Strobe, 11=Off, 12=Custom	1	EEPROM
Kodak DC 220/ DC 260/DC265/ DC 290	Only supports settings 1,3,6, 9 and 11 Set to Off, 11, to set the white balance manually.	1	EEPROM
Dimage EX 1500	Only supports settings 1,3,6, 9 and 11.	1	EEPROM

wpic – Watermark Logo File Name

Definition: The DOS filename of the watermark logo.

Data Type: DosName

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Requires wena parameter to be set to enable the logo.	""	EEPROM
Kodak DC 220	Tag not supported.		EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wpop – Watermark Logo Operation Mode

Definition: Sets the graphics mode for merging of the logo with the image.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	0=SourceCopy (Opaque Logo), 1=SourceOR (Translucent Logo)	0	EEPROM
Kodak DC 260, DC265, DC290	Setting 1, SourceOR, sets the logo's background color (alpha channel) to transparent.		
Kodak DC 220	Tag not supported.		EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wpxp – Watermark Logo X Position

Definition: Sets the horizontal position and justification for the logo.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=100, Units=percent (0 being left edge, 100 being right edge). < 50: Image is left justified at the x-coord equal to (image width)(value/100) = 50: Image is centered horizontally > 50: Image is right justified at the x-coord equal to (image width)(value/100)	4	EEPROM
Kodak DC 220	Tag not supported.		EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wpyp – Watermark Logo Y Position

Definition: Sets the vertical position and justification for the logo.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=100, Units=percent (0 being top edge, 100 being bottom edge). < 50: Image is top justified at the y-coord equal to (image width)(value/100) = 50: Image is centered vertically > 50: Image is bottom justified at the y-coord equal to (image width)(value/100)	96	EEPROM
Kodak DC 220	Tag not supported.		EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wsrc – Watermark Text Background Color

Definition: Sets the background color for watermark text.
 Data Type: Uhex
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0x0, Maximum=0xFFFFFFFF Color stored as YCrYCb where both Y's are the same. Default: Y=0 Cr=Cb=128(Black) Example hex color values: white=0xff80ff80, red=0x41d44164, green=0x703a7048, blue=0x237223d4, cyan=0x832c839c, magenta=0x54c654b8, yellow=0xa28ea22c, black=0x00800080	0x00800080	EEPROM
Kodak DC 220/ DC 260/DC 265	Data Type: UInt 0=white, 1=red, 2=green, 3=blue, 4=cyan, 5=magenta, 6=yellow and 7=black.	7	EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wsfc – Watermark Text Foreground Color

Definition: Sets the text color for watermark text.

Data Type: Uhex

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0x0, Maximum=0xFFFFFFFF Color stored as YCrYCb where both Y's are the same. Default:Y=255 Cr=Cb=128(White) Example hex color values: white=0xff80ff80, red=0x41d44164, green=0x703a7048, blue=0x237223d4, cyan=0x832c839c, magenta=0x54c654b8, yellow=0xa28ea22c, black=0x00800080	0xFF80FF80	EEPROM
Kodak DC 220/ DC 260/DC 265	Data Type: UInt 0=white, 1=red, 2=green, 3=blue, 4=cyan, 5=magenta, 6=yellow and 7=black.	0	EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wsop – Watermark Text Operation Mode

Definition: Sets the graphics mode (opaque, transparent or translucent) for merging the text with the image.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Opaque, 2=Translucent, 3=Transparent	1	EEPROM
Kodak DC 220/ DC 260/ DC 265	Setting 3 not supported Setting 2 makes the background color transparent.	0	EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wstr– Watermark Text String

Definition: Sets the user watermark text.
 Data Type: String
 Access: Read/Write
 Associated File Tag: wstr

Product/OS	Details	Factory Default	Stored In
Digita Script		""	EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wsts– Watermark Text Font Size

Definition: Sets the 'effective' font size for watermark text. The 'effective' size assumes the image will be scaled to a 4 x 6 inch print size.
 Data Type: Enum
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	9=9 point, 10 = 10 point, 12=12 point, 14=14 point, 16=16 point, 18=18 point, 24=24 point, 36=36 point, 48=48 point	12	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

wsxp – Watermark Text X Position

Definition: Sets the horizontal position and justification for the text.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=100, Units=percent (0 being left edge, 100 being right edge). < 50: Image is left justified at the y-coord equal to (image width)(value/100) = 50: Image is centered horizontally > 50: Image is right justified at the y-coord equal to (image width)(value/100)	96	EEPROM
Dimage EX 1500	Tag not supported.		

wsyp – Watermark Text Y Position

Definition: Sets the horizontal position and justification for the text.

Data Type: UInt

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=100, Units=percent (0 being top edge, 100 being bottom edge). < 50: Image is top justified at the y-coord equal to (image width)(value/100) = 50: Image is centered vertically > 50: Image is bottom justified at the y-coord equal to (image width)(value/100)	96	EEPROM
Dimage EX 1500	Tag not supported.		

wtbc – Watermark Time Background Color

Definition: Sets the background color for watermark date/time.
 Data Type: Uhex
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0x0, Maximum=0xFFFFFFFF wtop must be set to 1 (Opaque color) Color stored as YCrYCb where both Y's are the same. Default:Y=0 Cr=Cb=128(black) Example hex color values: white=0xff80ff80, red=0x41d44164, green=0x703a7048, blue=0x237223d4, cyan=0x832c839c, magenta=0x54c654b8, yellow=0xa28ea22c, black=0x00800080	0x00800080	EEPROM
Kodak DC 220/ DC 260/DC 265	Data Type: UInt 0=white, 1=red, 2=green, 3=blue, 4=cyan, 5=magenta, 6=yellow and 7=black.	0	EEPROM
Dimage EX 1500	Tag not supported.		

wtfc – Watermark Time Foreground Color

Definition: Sets the foreground color for watermark date/time.
 Data Type: Uhex
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0x0, Maximum=0xFFFFFFFF wtop must be set to 1 (Opaque color) Color stored as YCrYCb where both Y's are the same. Default:Y=255 Cr=Cb=128(white) Example hex color values: white=0xff80ff80, red=0x41d44164, green=0x703a7048, blue=0x237223d4, cyan=0x832c839c, magenta=0x54c654b8, yellow=0xa28ea22c, black=0x00800080	0xFF80FF80	EEPROM
Kodak DC 220/ DC 260/DC 265	Data Type: UInt 0=white, 1=red, 2=green, 3=blue, 4=cyan, 5=magenta, 6=yellow and 7=black.	0	EEPROM
Dimage EX 1500	Tag not supported.		

wtop – Watermark Date Background Mode

Definition: Sets the graphics mode (opaque, transparent or translucent) for merging the date/time with the image.

Data Type: Enum List

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Opaque, 2=Translucent, 3=Transparent	1	EEPROM
Kodak DC 220/ DC 260/ DC 265	Setting 3 not supported	0	EEPROM
Dimage EX 1500	Tag not supported.		EEPROM

wttts – Watermark Date Font Size

Definition: Sets the 'effective' font size for watermark date/time. The 'effective' size assumes the image will be scaled to a 4 x 6 inch print size.

Data Type: Enum

Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	9=9 point, 10 = 10 point, 12=12 point, 14=14 point, 16=16 point, 18=18 point, 24=24 point, 36=36 point, 48=48 point	12	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

wtxp – Watermark Time X Position

Definition: Sets the horizontal position and justification for the watermark date/time.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=100, Units=percent (0 being left edge, 100 being right edge). < 50: Image is left justified at the y-coord equal to (image width)(value/100) = 50: Image is centered horizontally > 50: Image is right justified at the y-coord equal to (image width)(value/100)	96	EEPROM
Dimage EX 1500	Tag not supported.		

wtyp – Watermark Time Y Position

Definition: Sets the vertical position and justification for the watermark date/time.
 Data Type: UInt
 Access: Read/Write

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=0, Maximum=100, Units=percent (0 being left edge, 100 being right edge). < 50: Image is top justified at the y-coord equal to (image width)(value/100) = 50: Image is centered vertically > 50: Image is bottom justified at the y-coord equal to (image width)(value/100)	4	EEPROM
Dimage EX 1500	Tag not supported.		

xcmp – Exposure Compensation

Definition: Sets the amount of forced change to the automatic exposure (AE).

Data Type: SInt

Access: Read/Write

Associated File Tag: xcmp

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum= -300, Maximum= 300. Units= 0.01 EV Not applicable for Manual or Programmed exposure mode set by "xmod" parameter.	0	RAM
Kodak DC 220/ DC 260/DC 265/ DC 290	Minimum= -200, Maximum= 200	0	RAM
Dimage EX 1500	Minimum= -300, Maximum= 300	100	RAM

xmod – Exposure Mode

Definition: Sets the exposure mode for the camera.

Data Type: Enum List

Access: Read/Write

Associated File Tag: xmod

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Auto, 2=Shutter Priority, 3=Aperture Priority, 4=Gain Priority, 5=Programmed, 6=Manual		EEPROM
Kodak DC 260/ DC 265DC 290	2 =long exposure mode, 3=external flash sync mode, Settings 4, 5 and 6 not supported.	1	EEPROM
Kodak DC 220	2 =long exposure mode, Settings 3, 4, 5 and 6 not supported.	1	EEPROM
Dimage EX 1500	Settings 2, 5 and 6 not supported.	1	EEPROM

xmtd– Exposure Method

Definition: Specifies the metering pattern used to determine the exposure.
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: xmtd

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Auto Matrix, 2=Multi Spot, 3=Center Weighted, 4=Center Spot, 5=Custom Matrix		EEPROM
Kodak DC 220/ DC 260/DC 265/ DC 290	Supports setting 3 only	3	EEPROM
Dimage EX 1500	Settings 4 and 5 not supported.	1	EEPROM

zmod– Zoom Mode

Definition: Specifies the operational mode of the zoom function.
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: zmod

Product/OS	Details	Factory Default	Stored In
Digita Script	1=Motorized, 2=Program, 3=Manual	1	EEPROM
Kodak DC 290	Supports setting 2only	2	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

zpos– Zoom Position

Definition: Controls the position of the zoom lens when the 'zmod' parameter is set to Programmed (2).
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: zpos

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=100, Maximum=300, Units=percent.	100	RAM
Kodak DC 260/ DC 265/DC 290	Minimum=100, Maximum=300	130	RAM
Kodak DC 220	Minimum=100, Maximum=200	100	RAM

zspd– Zoom Speed

Definition: Controls the position of the zoom lens when the 'zmod' parameter is set to Programmed (2).
 Data Type: Enum List
 Access: Read/Write
 Associated File Tag: zpos

Product/OS	Details	Factory Default	Stored In
Digita Script	Minimum=50, Maximum=200, Units=percent. 100 = normal zoom speed, 50 = half normal zoom speed, 200 = twice normal zoom speed.	100	EEPROM
Kodak DC 290	Minimum=100, Maximum=100	100	EEPROM
Kodak DC 220/ DC 260/DC 265	Tag not supported.		
Dimage EX 1500	Tag not supported.		

Appendix C Image File Tags

Image file tags are saved with each image file. Parameters that start with the lower case 'u' can be modified by the user. These tags need to be mapped to the tags that are either required or optional for different file formats. The image file tags vary by camera.

aagc – Analog Gain Value

Data Type: UInt

Product	Definition
Kodak DC 220	This field is variable and contains a value returned from the CCS.
Kodak DC 260/DC 265	This field is variable and contains a value returned from the CCS.
Kodak DC 290	This field is variable and contains a value returned from the CCS.
Dimage EX 1500	This field is variable and contains a value returned from the CCS.

aper – Aperture (F Number)

Data Type: UInt

Product	Definition
Kodak DC 220	Tag not supported
Kodak DC 260/DC 265	Tag not supported
Kodak DC 290	This field is variable in 0.01 unit, F4.0 = 400
Dimage EX 1500	Tag not supported

bsfd – Image Group Folder Name

Data Type: DosName

Product	Definition
Kodak DC 220	This field is variable and contains a DOS name with no DOS extension.
Kodak DC 260/DC 265	This field is variable and contains a DOS name with no DOS extension.
Kodak DC 290	Tag not supported. Replaced by cngn
Dimage EX 1500	This field is variable and contains a DOS name with no DOS extension.

ccsv – Camera Firmware Version

Data Type: UInt

Product	Definition
Kodak DC 220	The firmware version for this CCS, e.g., Version 1.0 is 0x01000000.
Kodak DC 260/DC 265	The firmware version for this CCS, e.g., Version 1.0 is 0x01000000.
Kodak DC 290	The firmware version for this CCS, e.g., Version 1.0.2 is 0x01000200.
Dimage EX 1500	The firmware version for this CCS, e.g., Version 1.0 is 0x01000000.

clos – Closest Subject Distance

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported — always set to zero (0).
Kodak DC 260/DC 265	Not supported — always set to zero (0).
Kodak DC 290	Not supported — always set to zero (0).
Dimage EX 1500	Not supported — always set to zero (0).

cmpn – Image Compression Level

Data Type: UInt

Product	Definition
Kodak DC 220	The degree of image compression, where 2=high (good), 3=normal (better) and 4=low (best).
Kodak DC 260	The degree of image compression, where 2=high (good), 3=normal (better) and 4=low (best).
Kodak DC 265	The degree of image compression, where 2=high (good), 3=normal (better), 4=low (best) and 5=least (super).
Kodak DC 290	The degree of image compression, where 2=high (good), 3=normal (better), 4=low (best) and 5=least, 7=Lossless.
Dimage EX 1500	The degree of image compression, where 2=economy, 3=standard, 4=fine and 5=super fine.

cmrc – Camera Region Code

Data Type: UInt

Product	Definition
Kodak DC 220	The code for the camera region, where 1=U.S. English and 8=Japanese.
Kodak DC 260/DC 265	The code for the camera region, where 1=U.S. English and 8=Japanese.
Kodak DC 290	Tag not supported. Replaced by rgnc .
Dimage EX 1500	The code for the camera region, where 1=U.S. English, 3=French, 6=German and 8=Japanese.

cmtyp – Image Compression Type

Data Type: Enum

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	1="None", 2="JPEG"
Dimage EX 1500	Tag not supported.

cngn – Image Natural Group Name

Data Type: DOS filename

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	This field is variable and contains a DOS file name without a DOS extension.
Dimage EX 1500	Tag not supported.

cpgn – Image Group Folder Name

Data Type: DOS filename

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	This field is variable and contains a DOS file name.
Dimage EX 1500	Tag not supported.

date – Image Capture Date

Data Type: UInt

Product	Definition
Kodak DC 220	The image capture date. For example, 0x00031595 is March 15, 1995.
Kodak DC 260/DC 265	The image capture date. For example, 0x00031595 is March 15, 1995.
Kodak DC 290	The image capture date. For example, 0x00031595 is March 15, 1995.
Dimage EX 1500	The image capture date. For example, 0x00031595 is March 15, 1995.

eval – Exposure Value

Data Type: UInt

eval	Definition
Kodak DC 220	The exposure value, in 0.01 EV units.
Kodak DC 260/DC 265	The exposure value, in 0.01 EV units.
Kodak DC 290	The exposure value, in 0.01 EV units.
Dimage EX 1500	The exposure value, in 0.01 EV units.

exht – Extracted Image Height

Data Type: UInt

Product	Definition
Kodak DC 220	The extracted image height.
Kodak DC 260/DC 265	The extracted image height.
Kodak DC 290	The extracted image height.
Dimage EX 1500	The extracted image height.

exwd – Extracted Image Width

Data Type: UInt

exwd	Definition
Kodak DC 220	The extracted image width.
Kodak DC 260/DC 265	The extracted image width.
Kodak DC 290	The extracted image width.
Dimage EX 1500	The extracted image width.

fars – Farthest Subject Distance

Data Type: UInt

Product	Definition
Kodak DC 220	not supported
Kodak DC 260/DC 265	not supported
Kodak DC 290	not supported
Dimage EX 1500	not supported

fdst – Focus Distance

Data Type: UInt

Product	Definition
Kodak DC 220	not supported
Kodak DC 260/DC 265	Focus distance — maximum value is 65535, which equals infinity.
Kodak DC 290	Focus distance — maximum value is 65535, which equals infinity.
Dimage EX 1500	not supported

flty – Image File Type

Data Type: Enum

Product	Definition
Kodak DC 220	Tag not supported
Kodak DC 260/DC 265	Tag not supported
Kodak DC 290	2=EXIF, 4=TIFF
Dimage EX 1500	Tag not supported

fmod – Focus Mode

Data Type: UInt

Product	Definition
Kodak DC 220	not supported
Kodak DC 260/DC 265	The focus mode, where 1=Auto- Fcus and 3=Manual Focus.
Kodak DC 290	The focus mode, where 1: Auto Focus, 2=Programmed, 3= Manual Focus.
Dimage EX 1500	The focus mode, where 1=Auto Focus. 3= Manual Focus.

fmys – Focus Method

Data Type: Enum

Product	Definition
Kodak DC 220	not supported
Kodak DC 260/DC 265	The focus method, where 1=multi-spot and 2=single-spot.
Kodak DC 290	The focus method, where 1=multi-spot Auto and 2=single-spot.
Dimage EX 1500	The focus method, where 2=center-spot.

fnum – F Number

Data Type: UInt

Product	Definition
Kodak DC 220	The F number, in 0.01 units. For example, F4.0 is 400.
Kodak DC 260/DC 265	The F number, in 0.01 units. For example, F4.0 is 400.
Kodak DC 290	Tag not supported. Replaced by aper .
Dimage EX 1500	The F number, in 0.01 units. For example, F4.0 is 400.

fwrv – Digita Firmware Version

Data Type: UInt

Product	Definition
Kodak DC 220	The Digita firmware version number. Version 1.0 is 0x01000000.
Kodak DC 260/DC 265	The Digita firmware version number. Version 1.0 is 0x01000000.
Kodak DC 290	The Digita firmware version number. Version 1.0.2 is 0x01000200.
Dimage EX 1500	The Digita firmware version number, e.g., 168886272.

grct – Image Group Sequence Number

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Not supported. Returns a value of 1..
Dimage EX 1500	Not supported.

grd1-4 – Image Group Data 1-4

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Not supported. Returns a value of 0.
Dimage EX 1500	Not supported.

grfd – Capture Group Folder Name

Data Type: DosName

Product	Definition
Kodak DC 220	Capture group folder DOS name, with no extension.
Kodak DC 260/DC 265	Capture group folder DOS name, with no extension.
Kodak DC 290	Tag not supported. Replaced by cpgn .
Dimage EX 1500	Capture group folder DOS name, with no extension.

grtg – Image Group Tag

Data Type: PName

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Not supported.
Dimage EX 1500	Not supported.

hint – Image Hint Mode

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Hint mode, where 1=Auto, 2=Portrait, 3=Landscape, 4=Closeup, 5=Sport, 6=Evening, 7=Night, 8=Slow Sync, 9=Panorama
Dimage EX 1500	Hint mode, where 1=Auto, 2=Portrait, 3=Landscape, 6=Evening, 7=Night and 8=Slow-Sync.

hwrv – Hardware Version

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Hardware version number 0x01000000 = version 1.0
Dimage EX 1500	Hardware version number 0x01000000 = version 1.0

icts – Image Capture Tone

Data Type: UInt

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	Tag not supported. Returns a value of 0.
Dimage EX 1500	Tag not supported.

iirv – Image Info Version

Data Type: UInt

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	version number 0x01000100 = version 1.0.1
Dimage EX 1500	Tag not supported.

imcn – Image File Type

Data Type: PName

Product	Definition
Kodak DC 220	The image file type: JPEG (0x4A504547) or FPX (0x46505820).
Kodak DC 260/DC 265	The image file type: JPEG (0x4A504547) or FPX (0x46505820).
Kodak DC 290	Tag not supported. Replaced by flty.
Dimage EX 1500	The image file type: EXIF (1246774599).

imfg – Image Flag

Data Type: UInt

Product	Definition
Kodak DC 220	Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported, not supported).
Kodak DC 260/DC 265	Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported, not supported).
Kodak DC 290	Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported, not supported).
Dimage EX 1500	Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported, not supported).

imht – Image Height

Data Type: UInt

Product	Definition
Kodak DC 220	Image height.
Kodak DC 260/DC 265	Image height.
Kodak DC 290	Image height.
Dimage EX 1500	Image height.

imis – Image Info Size

Data Type: UInt

Product	Definition
Kodak DC 220	1024 bytes.
Kodak DC 260/DC 265	1024 bytes.
Kodak DC 290	1024 bytes.
Dimage EX 1500	1024 bytes.

imiv – Image Info Version

Data Type: UInt

Product	Definition
Kodak DC 220	Version 1.0 (0x01000000).
Kodak DC 260/DC 265	Version 1.0 (0x01000000).
Kodak DC 290	Tag not supported. Replaced by iirv .
Dimage EX 1500	Image information version number.

impn – Image Color Depth

Data Type: Enum

Product	Definition
Kodak DC 220	24-bit full color (1).
Kodak DC 260/DC 265	24-bit full color (1).
Kodak DC 290	24-bit full color (1).
Dimage EX 1500	24-bit full color (1) or black/white (2).

imwd – Image Width

Data Type: UInt

Product	Definition
Kodak DC 220	Image width.
Kodak DC 260/DC 265	Image width.
Kodak DC 290	Image width.
Dimage EX 1500	Image width.

isam – Image Sharpness Amount

Data Type: UInt

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	Tag not supported. Returns a value of 0.
Dimage EX 1500	Tag not supported.

isra – Image Sharpness Radius

Data Type: Fixed

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	Field is variable. Minimum=0.1. Maximum=10.0
Dimage EX 1500	Tag not supported.

isth – Image Sharpness Threshold

Data Type: UInt

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	Field is variable. Minimum=0, Maximum=25
Dimage EX 1500	Tag not supported.

Ikmd – AE/AF Lock Mode

Data Type: UInt

Product	Definition
Kodak DC 220	Auto-exposure/auto-focus mode. Set to 1 (AE locked).
Kodak DC 260/DC 265	Auto-exposure/auto-focus mode. Set to 3 (AE and AF locked).
Kodak DC 290	Auto-exposure/auto-focus mode. Set to 3 (AE and AF locked).
Dimage EX 1500	Auto-exposure/auto-focus mode. Set to 3 (AE and AF locked).

mcap – Media Type Capture Mode

Data Type: Enum

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	1=Still;, 2=Burs, 3=TimeLapse
Dimage EX 1500	Tag not supported.

orn – Capture Orientation

Data Type: UInt

Product	Definition
Kodak DC 220	Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus.
Kodak DC 260/DC 265	Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus.
Kodak DC 290	Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus.
Dimage EX 1500	Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus.

pcnt – Picture Taken Count

Data Type: UInt

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	Variable
Dimage EX 1500	Tag not supported.

pgms – Program Shift

Data Type: SInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Not supported. Returns a value of 0.
Dimage EX 1500	Not supported.

ptid – Product Name

Data Type: String

Product	Definition
Kodak DC 220	KODAK DIGITAL SCIENCE DC220.
Kodak DC 260	KODAK DIGITAL SCIENCE DC260.
Kodak DC 265	KODAK DC265 ZOOM DIGITAL CAMERA
Kodak DC 290	KODAK DC290 Zoom Digital Camera
Dimage EX 1500	Dimage EX

rgnc – Region Code

Data Type: SInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	1=English, 3=French, 6=German, 8=Japanese
Dimage EX 1500	Not supported.

rmod – Auto Image Rotation Mode

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	0=OFF, 1=ON
Dimage EX 1500	Not supported.

scmp – Strobe Compensation

Data Type: SInt

scmp	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Not supported. Returns a value of 0.
Dimage EX 1500	Values in increments of 0.01EV..

scom – Sound Compression Type

Data Type: UInt

Product	Definition
Kodak DC 220	Set to 2 — ADPCM if sound exists, otherwise 0.
Kodak DC 260/DC 265	Set to 2 — ADPCM if sound exists, otherwise 0.
Kodak DC 290	Set to 2 — ADPCM if sound exists, otherwise 0.
Dimage EX 1500	not supported

sdsr – Sound Sampling Rate

Data Type: UInt

Product	Definition
Kodak DC 220	Set to 11025 — 11.025 KHz if sound exists, otherwise 0.
Kodak DC 260/DC 265	Set to 11025 — 11.025 KHz if sound exists, otherwise 0.
Kodak DC 290	Set to 11025 — 11.025 KHz if sound exists, otherwise 0.
Dimage EX 1500	not supported

sdss – Sound Sampling Size

Data Type: UInt

Product	Definition
Kodak DC 220	Set to 16 — mono 16 if sound exists, otherwise 0.
Kodak DC 260/DC 265	Set to 16 — mono 16 if sound exists, otherwise 0.
Kodak DC 290	Set to 16 — mono 16 if sound exists, otherwise 0.
Dimage EX 1500	not supported

sern – Serial Number

Data Type: UInt

Product	Definition
Kodak DC 220	Serial number.
Kodak DC 260/DC 265	Serial number.
Kodak DC 290	Serial number.
Dimage EX 1500	Serial number.

shpn – Sharpening Level

Data Type: UInt

Product	Definition
Kodak DC 220	Not supported.
Kodak DC 260/DC 265	Not supported.
Kodak DC 290	Not supported. Replaced by isam .
Dimage EX 1500	Not supported.

shut – Shutter Speed

Data Type: UInt

Product	Definition
Kodak DC 220	Shutter speed.
Kodak DC 260/DC 265	Shutter speed.
Kodak DC 290	Shutter speed.
Dimage EX 1500	Shutter speed.

smod – Strobe Mode

Data Type: Enum

Product	Definition
Kodak DC 220	Strobe mode, where 1=off, 2=auto, 3=fill.
Kodak DC 260/DC 265	Strobe mode, where 1=off, 2=auto, 3=fill, 5=external sync.
Kodak DC 290	Strobe mode, where 1=off, 2=auto, 3=fill, 5=external sync.
Dimage EX 1500	Strobe mode, where 1=off, 2=auto, 3=fill.

spre – Strobe Precount

Data Type: UInt

Product	Definition
Kodak DC 220	Strobe precount, where 1=red-eye reduction.
Kodak DC 260/DC 265	Strobe precount, where 1=red-eye reduction.
Kodak DC 290	Strobe precount, where 1=red-eye reduction.
Dimage EX 1500	Strobe precount, where 1=red-eye reduction.

time – Image Capture Time

Data Type: UInt

Product	Definition
Kodak DC 220	Image capture time, as for example 0x00211530 for 9:15:30 pm.
Kodak DC 260/DC 265	Image capture time, as for example 0x00211530 for 9:15:30 pm.
Kodak DC 290	Image capture time, as for example 0x00211530 for 9:15:30 pm.
Dimage EX 1500	Image capture time, as for example 0x00211530 for 9:15:30 pm.

ucnr – User Caption Region Code

Data Type: UInt

Product	Definition
Kodak DC 220	The default is 0 for no null string. This field can be set to any value by the user.
Kodak DC 260/DC 265	The default is 0 for no null string. This field can be set to any value by the user.
Kodak DC 290	The default is 0 for no null string. This field can be set to any value by the user.
Dimage EX 1500	The default is 0 for no null string. This field can be set to any value by the user.

ucnt – User Caption Text

Data Type: String

Product	Definition
Kodak DC 220	User caption text. The default value is the null string, ""
Kodak DC 260/DC 265	User caption text. The default value is the null string, ""
Kodak DC 290	User caption text. The default value is the null string, ""
Dimage EX 1500	User caption text. The default value is the null string, ""

ucpt – User Copyright Notice

Data Type: String

Product	Definition
Kodak DC 220	User copyright notice.
Kodak DC 260/DC 265	User copyright notice.
Kodak DC 290	User copyright notice.
Dimage EX 1500	User copyright notice.

ucrc – User Copyright Region Code

Data Type: UInt

Product	Definition
Kodak DC 220	User copyright region code. Defaults to 0 for no null string. This field can be set to any value by the user.
Kodak DC 260/DC 265	User copyright region code. Defaults to 0 for no null string. This field can be set to any value by the user.
Kodak DC 290	User copyright region code. Defaults to 0 for no null string. This field can be set to any value by the user.
Dimage EX 1500	User copyright region code. Defaults to 0 for no null string. This field can be set to any value by the user.

uc01-uc16 – User Categories 01-16

Data Type: DosName

Product	Definition
Kodak DC 220	User category. The default value is the null string, "".
Kodak DC 260/DC 265	User category. The default value is the null string, "".
Kodak DC 290	User category. The default value is the null string, "".
Dimage EX 1500	User category. The default value is the null string, "".

udt1-udt8 – User Data 1-8

Data Type: UInt

Product	Definition
Kodak DC 220	User data.
Kodak DC 260/DC 265	User data.
Kodak DC 290	User data.
Dimage EX 1500	User data.

uptd – User Protected

Data Type: UInt

Product	Definition
Kodak DC 220	User protected or non-protected.
Kodak DC 260/DC 265	User protected or non-protected.
Kodak DC 290	User protected or non-protected.
Dimage EX 1500	User protected or non-protected.

ur01-ur16 – User Categories 01-16 Region Code

Data Type: UInt

Product	Definition
Kodak DC 220	User category region code; defaults to 0 for no null string. This field can be set to any value by the user.
Kodak DC 260/DC 265	User category region code; defaults to 0 for no null string. This field can be set to any value by the user.
Kodak DC 290	User category region code; defaults to 0 for no null string. This field can be set to any value by the user.
Dimage EX 1500	User category region code; defaults to 0 for no null string. This field can be set to any value by the user.

usr1-usr4 – User Strings 1-4 Region Code

Data Type: UInt

Product	Definition
Kodak DC 220	User string region code; defaults to 0 for no null string. This field can be set to any value by the user.
Kodak DC 260/DC 265	User string region code; defaults to 0 for no null string. This field can be set to any value by the user.
Kodak DC 290	User string region code; defaults to 0 for no null string. This field can be set to any value by the user.
Dimage EX 1500	User string region code; defaults to 0 for no null string. This field can be set to any value by the user.

ust1-ust4 – User Strings 1-4

Data Type: String

Product	Definition
Kodak DC 220	User string. The default value is the null string, "".
Kodak DC 260/DC 265	User string. The default value is the null string, "".
Kodak DC 290	User string. The default value is the null string, "".
Dimage EX 1500	User string. The default value is the null string, "".

utag – User Tag

Data Type: PName

Product	Definition
Kodak DC 220	The user tag identifier.
Kodak DC 260/DC 265	The user tag identifier.
Kodak DC 290	The user tag identifier.
Dimage EX 1500	The user tag identifier.

vdid – Vendor ID

Data Type: String

Product	Definition
Kodak DC 220	Eastman Kodak Company.
Kodak DC 260/DC 265	Eastman Kodak Company.
Kodak DC 290	Eastman Kodak Company.
Dimage EX 1500	Minolta Co., Ltd.

vd01–vd04 – Vendor Data 1-4

Data Type: UInt

Product	Definition
Kodak DC 220	Vendor data.
Kodak DC 260/DC 265	Vendor data.
Kodak DC 290	Vendor data.
Dimage EX 1500	Vendor data.

wbcb – White Balance Blue

Data Type: Fixed

Product	Definition
Kodak DC 220	White balance blue.
Kodak DC 260/DC 265	White balance blue.
Kodak DC 290	White balance blue.
Dimage EX 1500	White balance blue.

wbcg – White Balance Green

Data Type: Fixed

Product	Definition
Kodak DC 220	White balance green.
Kodak DC 260/DC 265	White balance green.
Kodak DC 290	White balance green.
Dimage EX 1500	White balance green.

wbcr – White Balance Red

Data Type: Fixed

Product	Definition
Kodak DC 220	White balance red.
Kodak DC 260/DC 265	White balance red.
Kodak DC 2690	White balance red.
Dimage EX 1500	White balance red.

wmod – White Balance Mode

Data Type: UInt

Product	Definition
Kodak DC 220	White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten, 11=off (set manually).
Kodak DC 260/DC 265	White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten, 11=off (set manually).
Kodak DC 290	White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten, 11=off (set manually).
Dimage EX 1500	Indicates the auto-white balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten, 11=off (set manually).

wmrc – Watermark Region Code

Data Type: UInt

Product	Definition
Kodak DC 220	Watermark region code, where 1=U.S. English and 8=Japanese.
Kodak DC 260/DC 265	Watermark region code, where 1=U.S. English and 8=Japanese.
Kodak DC 2690	Watermark region code, where 1=U.S. English and 8=Japanese.
Dimage EX 1500	The region code, where 1=U.S. English, 3=French, 6=German and 8=Japanese.

wstr – Watermark Text String

Data Type: String

Product	Definition
Kodak DC 220	Watermark text string.
Kodak DC 260/DC 265	Watermark text string.
Kodak DC 290	Watermark text string.
Dimage EX 1500	Watermark text string.

xcmp – Exposure Compensation

Data Type: Int

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	Minimum=-200, Maximum=200, Default=0
Dimage EX 1500	Minimum=-300, Maximum=300, Default=100.

xmod – Exposure Mode

Data Type: Enum

Product	Definition
Kodak DC 220	Exposure mode, where 1=auto-exposure (program AE) and 2=shutter priority (long time exposure).
Kodak DC 260/DC 265	Exposure mode, where 1=auto-exposure (program AE), 2=shutter priority (long time exposure) and 3=aperture priority (external flash sync).
Kodak DC 290	Exposure mode, where 1=auto-exposure (program AE), 2=shutter priority (long time exposure) and 3=aperture priority (external flash sync).
Dimage EX 1500	Exposure mode, where 1 is program auto-exposure.

xmtd – Exposure Method

Data Type: Enum

Product	Definition
Kodak DC 220	Exposure method, where 3=centerweighted matrix.
Kodak DC 260/DC 265	Exposure method, where 3=centerweighted matrix.
Kodak DC 290	Exposure method, where 3=centerweighted matrix.
Dimage EX 1500	Exposure method, where 1=auto matrix.

zmod– Zoom Mode

Data Type: Enum

Product	Definition
Kodak DC 220	Tag not supported.
Kodak DC 260/DC 265	Tag not supported.
Kodak DC 290	2=Programmed mode The 'zpos' parameter controls the zoom position.
Dimage EX 1500	Tag not supported.

zpos – Zoom Position

Data Type: UInt

Product	Definition
Kodak DC 220	Zoom position.
Kodak DC 260/DC 265	Zoom position.
Kodak DC 290	Zoom position.
Dimage EX 1500	Zoom position.

Appendix D Date and Time Formats

Table 6 gives examples of U.S. English time formats.

Table 6. U.S. English Time Formats

Time Format	Example
1	1:25 PM
2	1:25:00 PM

Table 7 gives examples of U.S. English date formats.

Table 7. U.S. English Date Formats

Date Format	Example
1	3/23/95
2	Mar 23, 1995
3	Thu, Mar 23, 1995

Table 8 shows time-hour-minute formats for different regions.

Table 8. Sample Time-Hour-Minute Formats by Region

Region	Example
US English = 1	1:25 PM
UK English = 2	1:25 pm
French = 3	13:25
Italian = 4	13:25
Spanish = 5	13:25
German = 6	13:25 Uhr
Swedish = 7	13:25
Japanese = 8	13:25

Table 9 shows short-date formats for different regions

Table 9. Sample Short-Date Formats by Region

Region	Example
USEnglish = 1	3/23/95
UKEnglish = 2	23/3/95
French = 3	23/3/95
Italian = 4	23-03-1995
Spanish = 5	23/3/95
German = 6	23.03.1995
Swedish = 7	95-03-23
Japanese = 8	3/23/95

Appendix E Bitmask Table

The following table contains bitmasks, which when ANDed with a 32-bit bitfield will return the status of the specified individual bit. This assumes that bit 1 is the high (most significant) bit.

Bit	Bitmask
1	0x80000000
2	0x40000000
3	0x20000000
4	0x10000000
5	0x08000000
6	0x04000000
7	0x02000000
8	0x01000000
9	0x00800000
10	0x00400000
11	0x00200000
12	0x00100000
13	0x00080000
14	0x00040000
15	0x00020000
16	0x00010000
17	0x00008000
18	0x00004000
19	0x00002000
20	0x00001000
21	0x00000800
22	0x00000400
23	0x00000200
24	0x00000100
25	0x00000080
26	0x00000040
27	0x00000020
28	0x00000010
29	0x00000008
30	0x00000004
31	0x00000002
32	0x00000001

Appendix F **Japanese Characters**

The tables on the following pages contain the hexadecimal codes for Japanese characters.

FP CODE	Character	FP CODE	Character
01	♪	29)
02	♠	2A	*
03	♣	2B	+
04	♥	2C	,
05	♦	2D	-
06	を	2E	.
07	あ	2F	/
08	い	30	0
09	う	31	1
0A	え	32	2
0B	お	33	3
0C	や	34	4
0D	ゆ	35	5
0E	よ	36	6
0F	っ	37	7
10	二	38	8
11	あ	39	9
12	い	3A	:
13	う	3B	;
14	え	3C	<
15	お	3D	=
16	か	3E	>
17	き	3F	?
18	く	40	@
19	け	41	A
1A	こ	42	B
1B	さ	43	C
1C	し	44	D
1D	す	45	E
1E	せ	46	F
1F	そ	47	G
20		48	H
21	!	49	I
22	"	4A	J
23	#	4B	K
24	\$	4C	L
25	%	4D	M
26	&	4E	N
27	'	4F	O
28	(50	P

FP CODE	Character	FP CODE	Character
51	Q	79	y
52	R	7A	z
53	S	7B	{
54	T	7C	
55	U	7D	}
56	V	7E	~
57	W	7F	押
58	X	80	た
59	Y	81	ち
5A	Z	82	つ
5B	[83	て
5C	≠	84	と
5D]	85	な
5E	^	86	に
5F	_	87	ぬ
60	`	88	ね
61	a	89	の
62	b	8A	は
63	c	8B	ひ
64	d	8C	ふ
65	e	8D	へ
66	f	8E	ほ
67	g	8F	ま
68	h	90	み
69	i	91	む
6A	j	92	め
6B	k	93	も
6C	l	94	や
6D	m	95	ゆ
6E	n	96	よ
6F	o	97	ら
70	p	98	り
71	q	99	る
72	r	9A	れ
73	s	9B	ろ
74	t	9C	わ
75	u	9D	ん
76	v	9E	入
77	w	9F	切
78	x	A0	※

FP CODE	Character	FP CODE	Character
A1	。	C9	ノ
A2	「	CA	ハ
A3	」	CB	ヒ
A4	、	CC	フ
A5	・	CD	ヘ
A6	ヲ	CE	ホ
A7	ア	CF	マ
A8	イ	D0	ミ
A9	ウ	D1	ム
AA	エ	D2	メ
AB	オ	D3	モ
AC	ヤ	D4	ヤ
AD	ユ	D5	ユ
AE	ヨ	D6	ヨ
AF	ッ	D7	ラ
B0	ー	D8	リ
B1	ア	D9	ル
B2	イ	DA	レ
B3	ウ	DB	ロ
B4	エ	DC	ワ
B5	オ	DD	ン
B6	カ	DE	ゝ
B7	キ	DF	。
B8	ク	E0	☆
B9	ケ	E1	★
BA	コ	E2	○
BB	サ	E3	●
BC	シ	E4	◎
BD	ス	E5	◇
BE	セ	E6	◆
BF	ソ	E7	□
C0	タ	E8	■
C1	チ	E9	△
C2	ツ	EA	▲
C3	テ	EB	▽
C4	ト	EC	▼
C5	ナ	ED	☞
C6	ニ	EE	☜
C7	ヌ	EF	☝
C8	ネ	FO	☞

FP CODE	Character
F1	年
F2	月
F3	日
F4	時
F5	分
F6	秒
F7	午
F8	前
F9	後
FA	上
FB	下
FC	左
FD	右
FE	撮
FF	影

Index

A

AF system 24
 Alert 16
 arithmetic assignment 5
 arithmetic operators 3
 assignment
 arithmetic 5
 bitwise 5
 operators 3
 statement syntax 5
 string 5
 automatic
 counting 53
 exposure 24
 focus 24

B

b data type 2
 backslash character 2
 backward reference 9
 batteries, low 23
 baud rate 46
 BitFlag tables 4
 bitmasks 149
 bitwise assignment 5
 bitwise operators 3

C

camera
 capabilities commands
 13
 error 23
 parameter name-value
 pair 26
 parameter, current value
 21
 parameters, restore to
 defaults 44
 power level 37
 powering off 51
 product information 38
 ready flag 24
 restarting with app 45

 script execution
 commands 14
 state, setting 49
 storage status 39
 Camera Control Manager 32
 camr 24
 capability
 list items, number of 26
 type 28
 value 27
 capture
 commands 14
 minimal 37
 mode 10
 process, ending 18
 reduced 37
 sequence 49
 sequence, starting 52
 captured
 files, list of 31
 files, number of 39
 images, number of 30
 CaptureStatus 22, 24
 carriage return 1, 58
 CCD specifications 33
 cerr 23, 29
 clearing error flag 29
 clearing the LCD 17
 clock
 commands 14
 current state of 28
 setting 50
 close serial port 45
 closing a file 19
 command statement 6
 commands
 conventions 1
 error codes 11
 format 4
 comment statement 7
 comments 1
 compressed files 39
 conditional statements 8
 continuation statements 1
 conventions for commands 1
 counting, automatic 53
 creating a file 19

current date 28, 50
 current time 28, 50
 custom script function 25

D

data
 alignment 4
 data type
 value assumptions 3
 date
 formats, by region 148
 formatted 29
 get current 28
 setting 50
 declarative statement 9
 declare statement 9
 default
 param value, factory 44
 param value, user 44
 user values, setting 48
 values, retrieving 20
 DEFAULT.CAM 45
 delay timer 23
 delete image files 18
 devices, external 46
 dfmt 29
 disk, number of captured
 images 30
 disk, RAM 25
 Display 17
 display an alert 16
 display commands 15
 display feedback text 16
 DisplayClear 17
 DisplayLine 17
 DOS
 filenames, script files 2
 name, image files 31

E

ed1a 23
 ed2a 23
 edch 23
 EndCapture 18

EraseFile 18
 error
 camera 23
 codes 11
 description 29
 flag, clearing 29
 script 25
 exit statement 9
 exitscript 9
 expansion
 disk 1 23
 disk 1, file info 31, 36
 disk 1, vendor status 25
 disk 2 23
 disk 2, file info 31, 36
 disk 2, vendor status 25
 disk changed 23
 expl 24
 exposure
 locked 24
 zone specification 34
 external devices 46
 external power status 23
 extp 23

F

f data type 2
 factory defaults 20, 44
 feedback text 16, 17
 file
 commands 15
 FileClose 19
 FileOpen 19
 files
 captured, list of 31
 captured, number of 39
 commands 14
 compressed 39
 creating 19
 list status change 23
 name, system 41
 names, image files 31
 naming conventions,
 script files 2
 new, information on 36
 number of new 35
 system, number of 40
 tag 32
 unprocessed 39
 writing to 57
 flsc 23, 31, 36
 focl 24

focus
 locked 24
 zone specifications 34
 format of commands 4
 formatted date 29
 forward reference 9
 free storage 39
 full operation 37
 full size picture 31, 36

G

GetCameraDefault 20
 GetCameraState 21
 GetCameraStatus 22
 GetCapabilitiesCount 26
 GetCapabilitiesListItem 26
 GetCapabilitiesRange 27
 GetCapabilitiesValue 27
 GetCapabilityType 28
 GetClock 28
 GetDateString 29
 GetError 29
 GetFileCount 30
 GetFileInfo 31
 GetFileTag 32
 GetImageSpecifications 32
 GetMarkedImage 34
 GetMarkedImageCount 35
 GetNewFileCount 35
 GetNewFileInfo 36
 GetOption 37
 GetPowerMode 37
 GetProductInfo 38
 GetScriptName 38
 GetStorageStatus 39
 GetString 40
 GetSystemFileCount 40
 GetSystemFileName 41
 GetTimeString 41
 goto statements 9
 group
 capture 18
 capture sequence 49
 name 53
 sequence number 53
 guidelines for script
 statements 1

H

hed1 25
 hed2 25

horizontal tabs 1
 horizontal zones 34
 hrmd 25

I

i data type 2
 identifiers 1
 identifiers, length of 1
 image
 captured, number of 30
 current number of
 marked 35
 file name format 31
 file processing completed
 flag 23
 file retrieval 34
 information commands
 13
 information, hardware-
 related 32
 marked 34
 number of new 35
 number, in file name 31
 partial 31, 36
 processing chain 23
 processing complete 31,
 36, 52
 processing in progress
 23, 52
 input text string 40
 internal storage 31, 36, 39
 internal storage, deleting files
 18
 ipcm 31, 36
 ipip 23, 52

J

jump to (goto) 9

L

label marker definition 9
 label statement 10
 LCD, clearing 17
 left orientation sensor 25
 left portrait position 25
 linefeed 1
 long operations 16
 lors 25
 lower case letters 1
 low-power state 23

M

macro
 button 23
 position 23
 MakeFolder 42
 marked image commands 15
 marked images 34
 mcro 23
 memf 23
 memory full flag 23
 memory, working 31, 36
 menu label 10
 menu statement 10
 minimal capture 37
 min-max range 27
 mode statement 10

N

n data type 2
 name of script 38
 name statement 11
 name-value pair 26
 naming conventions 2
 new files 36
 NULL strings 3
 number of captured images 30

O

oexp 24
 open serial channel 46
 opening a file 19
 operators 3
 arithmetic 3
 assignment 3
 bitwise 3
 relational 3
 option
 list 37
 list commands 13
 list, displaying to user 51
 selecting 37
 orientation
 sensor 25
 overexposure 24

P

parameter
 capability type 28

 capability value 27
 name-value pair 26
 settings, updating 49
 value, restore defaults 44
 partial image 31, 36
 ped1 31, 36
 ped2 31, 36
 piav 31, 36
 picture type 31, 36
 piwm 31, 36
 play mode 10
 portrait
 position, left 25
 position, right 25
 power
 alert 23
 commands 14
 level, camera 37
 low 23
 powering off the camera 51
 prmd 31, 36
 product
 commands 13
 information 38
 programmed sequence,
 terminating 18
 ptyp 31, 36
 pwra 23

Q

quotes 2

R

RAM
 disk 23, 25
 ramda 23
 range, min-max 27
 raw image data 39
 read
 commands 15
 receive cycle 46
 receive from serial port 46
 reduced capture 37
 region code 29
 relational operators 3, 8
 removable media 39
 removable media, deleting
 files 18
 RestoreCameraDefault 42,
 43, 44
 retrieving image files 34

review mode 10
 rgnc 29, 41
 right orientation sensor 25
 right portrait position 25
 rors 25
 rsvd 23, 24, 25
 RunApp 45
 RunScript 45

S

s data type 2
 sbs1 23
 sbs2 23
 scrennail specifications 34
 script
 command summary 13
 commands, error codes
 11
 custom function 25
 defining menu for 10
 error 25
 file name conventions 2
 file, running 45
 language operators 3
 language statements 4
 menu label for 10
 name, retrieving 38
 statement guidelines 1
 wait for shutter button 57
 waiting during 56
 select option 37
 self test
 complete 23
 in progress 23
 sendlist 6
 sensor
 left orientation 25
 right orientation 25
 upside down 25
 separators 1
 serial channel, opening 46
 serial port
 closing 45
 opening 46
 receiving data 46
 sending data 46
 serial processing commands
 14
 SerialClose 45
 SerialOpen 46
 SerialReceive 46
 SerialSend 47

- SerialSendReceive 48
- serr 25
- set clock value 50
- SetCameraDefault 48
- SetCameraState 49
- SetCaptureMode 49
- SetClock 50
- SetOption 50, 51
- SetPowerMode 51
- setting user tag 52
- SetUserFileTag 52
- shake warning 24
- shkw 24
- shutter
 - button 57
 - button status 23
 - wait for 57
- srun 25
- StartCapture 53
 - when not accepted 23
- state commands 13
- status commands 13
- status display 25
- stcl 24
- stdc 25
- stfa 24
- still
 - capture sequence 49
- storage status 39
- storage, internal 31, 36, 39
- strc 24
- string
 - commands 15
- string assignment 5
- string commands 15
- strobe
 - charging flag 24
 - ready flag 24
 - subject too close 24
 - subject too far away 24
- strr 24
- stsc 23
- stst 23
- subject
 - too close 24
 - too far away 24
- supported statements 4
- system file name 41
- system files, number of 40
- SYSTEM folder 40, 41
- SystemStatus 22, 23

T

- tabs, horizontal 1
- terminate a programmed sequence 18
- text, input from user 40
- tfmt 41
- thumbnail 31, 36
 - available 25
 - specifications 33
- time
 - burst timer flag 23
 - format, by region 147
 - get current 28
 - get formatted 41
 - setting 50
- time lapse
 - capture mode, side-effects 54
 - capture sequence 49
 - timer flag 23
- tnav 25
- trun 23
- truncation 3

U

- u data type 2
- udos 25
- uexp 24
- underexposure 24
- uninitialized variables 3
- unprocessed files 39
- upper case letters 1
- upside down orientation
- sensor 25
- user
 - default values 48
 - defaults 20, 44
 - input text 40
 - tag, setting 52

V

- variables, uninitialized 3
- VendorStatus 22, 25
- vertical zones 34

W

- Wait 56
- wait commands 15
- WaitForShutter 57

- working memory 31, 36
- Write 57
- write
 - commands 15
- WriteLine 58

Z

- zone
 - horizontal 34
 - organization 32
 - specification, exposure 34
 - specification, focus 34
 - vertical 34
- zoom
 - button, status of 23
- zoom (system status bit) 23