



Scripting Manual

for DigitaScript 1.5

Part Number 6-2120-0001-01

Document Version 1.5.2



© 2000 FlashPoint Technology, Inc. All Rights Reserved.

Digita, DigitaScript, 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.

Contents

About This Book xi

| | |
|--|--------------|
| Chapter 1 Script Grammar | 1 |
| Naming and Installing Scripts | 1 |
| DigitaScript Command Format | 2 |
| DigitaScript Conventions | 2 |
| Statements | 2 |
| Script Language File Names | 10 |
| Naming Conventions for Variables | 11 |
| Operators | 13 |
| Data Type Assumptions | 13 |
| Error Codes | 13 |
| Chapter 2 Script Command Reference | 15 |
| Script Commands | 15 |
| Script Commands by Function | 15 |
| Chapter 3 Scripting Techniques | 57 |
| Error Checking | 57 |
| Capturing Images | 58 |
| Startup Scripts | 59 |
| Extracting Bits From a Bitfield | 60 |
| Watermark Placement | 61 |
| Japanese Text Usage In Scripts | 62 |
| Appendix A Product Information Parameters | A-65 |
| Appendix B Camera Capabilities Parameters | B-79 |
| Standard Parameters | B-80 |
| Custom Parameters | B-161 |
| Appendix C Image File Tags | C-165 |
| Appendix D Date and Time Formats | D-203 |
| Appendix E Bitmask Table | E-205 |
| Appendix F Japanese Characters | F-207 |
| Index | I-213 |

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 and Websites

The *DigitaScript 101 Guide* provides an introduction to the DigitaScript language. For a listing of downloadable scripts for each Digita enabled camera, visit our website at:

http://www.digitaphoto.com/script_central/.

Text Conventions

code

sample code

italics

references to other manuals

DigitaScript is a series of commands, expressed in plain ASCII text, which can be interpreted by system software in a Digita-equipped camera. The scripts are independent of the actual Digita camera. As scripts are loaded, the camera menus display the available scripts. Users select scripts from the menu options through the camera control buttons. The script language provides commands to request additional user input during script execution.

The script interpreter processes the script line by line. The script language includes conditional branching (“if” constructs) as well as unconditional goto statements. In addition to these basics, the language provides specialized commands to control the camera, retrieve information from the camera, and access the image data on disk. Also, the serial port can be used to control remote devices.

The DigitaScript language supports marker and goto control statements, keywords, common data types (signed and unsigned integer, bitwise Boolean, fixed point, and strings) numeric and logical operators, declaration statements, and variables and constants for all data types.

Table 1 displays the DigitaScript version associated with each of the Digita enabled cameras.

Table 1. DigitaScript Versions

| Manufacturer | Camera Name | Script Version |
|-----------------|---------------------------|--------------------------|
| Kodak | DC 220 Zoom | DigitaScript Version 1.0 |
| | DC 260 Zoom | DigitaScript Version 1.0 |
| | DC 265 Zoom | DigitaScript Version 1.1 |
| | DC 290 Zoom | DigitaScript Version 1.5 |
| Minolta | Dimage Ex Zoom 1500 | DigitaScript Version 1.1 |
| | Dimage Ex Wide 1500 | |
| | Dimage Ex Zoom 1500 Ver 2 | DigitaScript Version 1.1 |
| | Dimage Ex Wide 1500 Ver 2 | |
| Hewlett Packard | PhotoSmart C500 | DigitaScript Version 1.5 |
| | PhotoSmart 618 | DigitaScript Version 1.5 |
| | PhotoSmart 912 | DigitaScript Version 1.5 |
| Pentax | EI-200 | DigitaScript Version 1.5 |
| | EI-2000 | DigitaScript Version 1.5 |

Naming and Installing Scripts

Scripts are plain text files created on Macintosh[®], Windows[®], or Unix[®] systems. The names of the files follow DOS[®] naming conventions; that is, they are of the form XXXXXXXX.CSM. The first eight characters can be any combination of letters, numbers, and underscore characters (_). The last four characters must be .CSM in order for Digita to recognize the file as a script. Scripts are placed in the SYSTEM directory on the camera, usually on a CompactFlash card. Each file name must be unique.

To install a script, simply place it in the SYSTEM directory of the camera. To do so, copy the file to the SYSTEM directory from a computer or via a connection utility. Please see the user manual for the application or the computer operating system for more information on copying files. The Digita operating environment recognizes that a new script has been installed when the camera is restarted or when the CompactFlash card is removed and reinserted.

To actually use a script, enter menu mode on the camera. Then, select the script from whichever menu it appears under. Press the softkey labeled “Start” (Minolta users press the “Edit” softkey) to run the script.

DigitaScript Command Format

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

CommandName (input-parameters, output-parameters)

- Input parameters are underlined.
- 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.

Syntax Notes:

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

DigitaScript Conventions

This section describes the conventions for the following elements DigitaScript command set:

- Statements
 - Required Statements
 - Additional Statements
- Comments
- Script Language File Names
- Variable Types
- Operators
- Data Type Assumptions
- Error Codes

Statements

Statements in the DigitaScript 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 statements are always on one line, and continuation statements are not supported.
- Elements in a statement are separated with blanks or horizontal tabs.

Scripts have several statements which are required. A script begins with a name statement, followed by mode, menu, and label statements. After the label statement comes the body of the script. The last line of the script is an exitscript statement.

Required Statements

Name Statement

The name statement is a long name for the script. The name is a string of up to 31 characters. It must be enclosed in double quotes (""). This name is used primarily from within other scripts via the GetScriptName() command. It does not appear in menus.

The format of the name statement is:

```
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.

Mode Statement

The mode statement is used to tell the camera in which mode to display the script in a menu. The name of the script will be displayed on its designated menu only when the Digita enabled device is in this mode. The mode is a single number which specifies the mode. Currently, the supported modes are 0 (capture), 1 (review), and 2 (play). Note that the review and play modes have been combined on the Kodak DC220, DC260, DC265, and DC290, and are the same for scripting purposes.

The format of the mode statement is:

```
mode n
n          numerical value (0, 1, 2) 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
```

This sets the script to be displayed in capture mode.

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.

Menu Statement

The menu statement tells the camera under which menu to display the script label. The menu name is a string of up to 31 characters enclosed in double quotes (""). (The string is truncated to fit the display screen; this varies among cameras.) If the menu does not exist in the specified mode, it is created. If the menu does exist, the camera places the script under that menu after the last item in the menu.

The format of the menu statement 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.

Label Statement

The label statement tells the camera what name to display for the script under the specified menu. The label consists of up to 31 characters enclosed in double quotes (""). (The string is truncated to fit the display screen; this varies among cameras.) Usually, they will be fairly short in order to make them more readable. If a script uses the same label as another script in the same menu, both labels will appear under the menu. As this is confusing to users, it is useful to label each of your scripts differently.

The format of the label statement is:

```
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.

Exitscript Statement

The last statement in a script should be an exitscript statement. It tells the script interpreter to close the script and return the camera to the standard operating mode.

The format of the exitscript statement is:

```
exitscript
```

An exitscript statement may also be placed elsewhere in a script. For example, the display could show a menu that asks the user whether or not they want to capture more images. If they don't, the script can execute an exitscript statement to terminate the script without requiring a jump to the end of the script.

Sample Script

Below is an example of the minimum elements necessary for a valid script. Note that this script does nothing except appear in the capture mode in the Sample Scripts menu with the label Do Nothing.

```
name "Do nothing script example"  
mode 0  
menu "Sample Scripts"  
label "Do Nothing"  
exitscript
```

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. If you have a long comment, place it after the name, mode, menu, and label statements.

Additional Statements

The rest of the script between the label and exitscript statements consists of a series of statements. These statements tell the camera to actually do something, such as get information from the user, write information to a file, or display information to the user. The following additional statements are supported:

- assignment statements
- command statements
- comment statements
- conditional statements
- declarative statements
- marker statements
- goto statements

Assignment Statements

Assignment statements are used to give values to identifiers. An assignment statement consists of a variable name followed by an equal sign (=) followed by a value or another identifier which has already had a value assigned.

The format of an assignment statement is:

```
variable1 = 2
```

This example sets the value of variable1 equal to 2.

Note that the right side of an assignment statement can include arithmetic operations such as addition, subtraction, multiplication, and division for identifiers of type u, i, and f. For example, the statement:

```
variable1 = dataValue + 2
```

sets the value of variable1 equal to the value of dataValue plus 2.

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 – DigitaScript 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 – DigitaScript 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 – DigitaScript 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 DigitaScript language commands. A command statement generally includes one or more parameters enclosed within parentheses ().

An example of a command statement is:

```
Display ("Hello, world")
```

This example would display the words “Hello, world” on the camera’s display.

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", uWbal )
```

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 – DigitaScript 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 – DigitaScript 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

Conditional statements allow scripts to make decisions. A conditional statement consists of the word “if” followed by a conditional expression. The following lines contain the statements to be executed if the conditional expression is true. Finally, the conditional statement ends with the word `end`.

For example, the statement:

```
if variable1 == 2
    variable1 = variable1 + 1
end
```

first determines whether or not the value of `variable1` is equal to 2. If it is, then it adds 1 to the value of `variable1`. If not, then the script continues on the first line following the line containing the word `end`.

Other comparisons that can be used in the conditional expression include `>` (greater than), `<` (less than), and `!=` (not equal to).

The syntax for conditional statements is as follows:

```
if conditional-expression
    arithmetic-statements | bitwise-statement | string-statement |
    jump-statement | function-statements | UserInterface-statements
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 4 on page 13.

Table 2 shows the supported data types and operators that apply to each.

Table 2. 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 DigitaScript 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
```

Declaration Statements

Declaration statements allow you to tell the script interpreter what identifiers (variables) you will use in a script. Identifiers must be declared before they are used. FlashPoint recommends that you declare all of your variables near the beginning of the script. This makes them easier to find when reading a script. However, as long as the identifier is declared before it is used, it may be declared anywhere within the body of a script.

Declaration statements begin with the word “declare” and are then followed by an identifier type, a colon (:), and a list of identifier names.

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: uNum1, uNum2
declare    s: sStr1, sStr2
```

Identifiers

DigitaScript language identifiers are a series of characters that specify a value or other language element. The following rules apply to identifiers in the DigitaScript 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.
- All characters in the identifier name after the first may be any combination of letters, numbers, and underscore (_) characters.
- Upper and lower case letters are not interpreted as the same. Because of this, Foo and foo are two different identifiers.
- Identifiers may be any length, but DigitaScript only pays attention to the first 31 characters. Therefore, AReallyLongIdentifierNameThatYouAreUsing is seen by DigitaScript as AReallyLongIdentifierNameThatYo.

Marker Statements

Markers create a reference by identifying a specific location in your script. They identify a location in a script and can be used with goto statements to create loops. Marker statements consist of a marker identifier followed by a colon (:). 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 allowing goto statements to be used for loop functions. 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.

Script Language File Names

Script file naming conventions are as follows:

<name>.CSM DigitaScript file names conform to the DOS file format (8.3). Names can include letters, digits, or any of the following special characters: \$ % ' - _ @ { } ~ ! # ()

<name> A string of up to eight alphanumeric characters. Use upper case characters. The first character cannot be a number.

.CSM indicates that the script is a general “camera script module” loaded by the camera application.

An example script file name is MYTEST_1.CSM

Naming Conventions for Variables

If you look at DigitaScripts written by different people, you will see that everyone has their own style. Some will use the shortest name possible or use variables like; Var1, Var2, etc. This type of variable makes it much more difficult to figure out how a script works. Script modification becomes a major task for the originator if they haven't looked at the code for a while and for an outsider looking at the code. It's up to the scriptwriter to make the script user friendly. This helps both the writer and anyone who wants to modify or add to a script.

Variables can represent seven different types of data, (see Table 3). Knowing what type of variable you're looking at while reading a script makes understanding it much easier. Having to look at the declare statements at the top of the script to understand a variable four pages into the script is not user friendly. It also helps if you know whether you're setting a value or retrieving a value, although both functions are sometimes done using the same variable.

One approach to making scripts user-friendly is the use of a capital S for "Set", R for "Retrieve". A lower case character is used in front of the S or R to indicate the type of data the variable represents. These two characters are followed by the variable description/name. (Example: `uSetCameraStatus`)

Naming Variables

Always use variable names that have meaning and are descriptive. Keep this in mind: "What does the variable represent and/or how is it used." The more descriptive the variable name, the better. When your variables have meaning, your need for comment statements will be reduced. Comment statements are very important and it's good coding practice to use them when needed.

Using capitalization also helps when looking at a variable. (Example: `UserOptionChoice`)

Variable names should not exceed 31 characters.

Example Variables

| Variable | Description |
|-----------------------------|---|
| uSstimSelfTimer | u = unsigned integer S = set value stim = camera parameter, Self Timer Enable |
| declare u: UserOptionChoice | declare u: UserOptionChoice = Variable declaration |
| uRUserOptionChoice | u = unsigned integer R = retrieved value UserOptionChoice = Variable |

Table 3. Variable Data Types

| Data Type | Description |
|-----------|---|
| u | An unsigned 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 $+32767.9999$. <i>Note: Fixed point decimals are not displayed as “nn.nn”, but as signed long data, which is not useful for most purposes. To convert it to useful information, divide by 65, 536 to get the integer part of the number.</i> |
| 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 “0b111111100000011”, 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 DigitaScript V1.5 or greater.

Operators

The operators supported by the DigitaScript language are listed in Table 4.

Table 4. DigitaScript 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, ~ (ones 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 |

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

Data Type Assumptions

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

- A string of 3 to 8 characters that ends with a period and a 3 character extension is 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.

Error Codes

This section describes the error codes that may be passed back in response to a command. The error codes shown in Table 5 are common to all script commands.

Table 5. 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. |

Table 5. Error Codes

| Error Code | Explanation |
|-----------------|--|
| 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 Commands

This chapter provides details of the DigitaScript commands. The commands are listed in two ways: the first is by function as displayed in Table 6, the second is in alphabetical order.

Script Commands by Function

Table 6 groups the commands by functions.

Table 6. Script Commands Grouped by Function

| Command | Description | Location |
|---|--|----------|
| Product and Image Information Commands | | |
| GetProductInfo | Requests specific information about the product by parameter | page 36 |
| GetImageSpecifications | Returns hardware-related image information, including zone organization | page 30 |
| Status Commands | | |
| GetCameraStatus | Returns system status, capture status, and vendor status | page 23 |
| GetError | Clears error flag and returns last error and its description | page 28 |
| Option List Commands | | |
| SetOption | Displays an option list to the user | page 49 |
| GetOption | Prompts the user for an option selection | page 35 |
| Camera Capabilities And State Commands | | |
| GetCapabilityType | Returns capability type associated with the specified camera parameter | page 27 |
| GetCapabilitiesRange | Returns the min-max range information associated with the specified camera parameter | page 26 |
| GetCapabilitiesCount | Returns the number of capability list items defined for the specified camera parameter | page 25 |
| 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 26 |
| GetCameraState | Requests a current camera parameter setting | page 23 |
| SetCameraState | Updates a current camera parameter setting | page 47 |
| GetCameraDefault | Requests the user or factory default value of a camera parameter setting | page 22 |
| SetCameraDefault | Updates the user default value for a camera parameter setting | page 46 |
| RestoreCameraDefault | Restores either the user defaults or the factory defaults | page 42 |
| Power and Capture Commands | | |
| GetPowerMode | Determines the power level available to the camera | page 35 |

Table 6. Script Commands Grouped by Function

| Command | Description | Location |
|---|--|----------|
| SetPowerMode | Powers off the camera | page 49 |
| SetCaptureMode | Controls the type of capture sequence: still, group, or timelapse | page 47 |
| StartCapture | Starts the capture process | page 51 |
| EndCapture | Stops the capture process | page 19 |
| File Commands | | |
| GetFileCount | Returns the total number of captured images resident on the disk | page 29 |
| GetFileInfo | Accesses the list of all captured image files currently available | page 29 |
| GetNewFileCount | Returns the number of new images resident on the disk | page 33 |
| GetNewFileInfo | Accesses the list of all newly created image files currently available | page 34 |
| EraseFile | Delete image files or other files | page 20 |
| GetStorageStatus | Determines the status of storage available to the camera | page 37 |
| GetFileTag | Returns the value of the specified image file tag | page 30 |
| SetUserFileTag | Sets the value of the specified user image file tag | page 50 |
| MakeFolder | Creates a new directory | page 40 |
| Date and Time Commands | | |
| GetClock | Gets the current date and time | page 27 |
| SetClock | Sets the current clock value for date and time | page 47 |
| GetDateString | Returns the date as a formatted string | page 28 |
| GetTimeString | Returns the time as a formatted string | page 39 |
| Camera Script Execution Commands | | |
| GetSystemFileCount | Returns the number of system files resident in the SYSTEM folder | page 38 |
| GetSystemFileName | Returns a single system file name based on an index | page 39 |
| RunApp | Restarts the camera using the selected application | page 43 |
| RunScript | Runs a script file that is listed in the SystemFilesList | page 43 |
| GetScriptName | Retrieves the long name of a script | page 36 |
| Serial Processing Commands | | |
| SerialOpen | Opens the serial channel at the selected baud rate | page 43 |
| SerialSendReceive | Sets up for the next data exchange through the serial port | page 45 |
| SerialSend | Sends the indicated data and sets up the next receive cycle | page 45 |
| SerialReceive | Waits to receive an expected transmission | page 44 |
| SerialClose | Releases the serial port | page 43 |
| File and Read/Write Commands | | |
| FileOpen | Opens a file in the specified directory with the specified filename | page 21 |
| FileClose | Closes a file | page 21 |
| Read | Reads data from a text file | page 41 |
| ReadLine | Reads one line of data from a text file | page 42 |
| Set Delimiter | Sets the delimiter used by the Read and ReadLine commands | page 48 |
| Write | Appends the specified data to the current pointer location in an open file (no carriage return added at end) | page 55 |

Table 6. Script Commands Grouped by Function

| Command | Description | Location |
|------------------------------|--|----------|
| 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 56 |
| String Manipulation Commands | | |
| FindString | Places the pointer at the start of a substring within a source string | page 22 |
| NumberToString | Converts a numerical value to a string value | page 41 |
| StringToNumber | Converts a string to a numerical value | page 53 |
| Substring | Extracts and returns a substring from a source string | page 52 |
| Wait And Display Commands | | |
| Wait | Specifies the number of milliseconds for the script program to wait | page 54 |
| Display | Places feedback text in the LCD (liquid crystal display) | page 18 |
| DisplayLine | Places feedback text in the LCD and appends a carriage return | page 19 |
| DisplayClear | Clears the LCD | page 19 |
| GetString | Requests an input text string from the user | page 38 |
| WaitForShutter | Returns camera system control to the user while waiting for the shutter button to be pressed. Valid only in “capture” mode | page 55 |
| Alert | Places an alert on the LCD | page 17 |
| Image Commands | | |
| GetMarkedImageCount | Returns the number of images currently marked | page 33 |
| GetMarkedImage | Retrieves an image file | page 32 |
| MarkImage | Marks a specified image | page 40 |
| MarkAllImages | Marks all the images on the CompactFlash card | page 40 |
| UnMarkImage | Unmarks a specified image | page 54 |
| UnMarkAllImages | Unmarks all the images on the CompactFlash card | page 54 |

Script Commands 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 script commands 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 6 on page 15.

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

Refer to “The DigitaScript language supports marker and goto control statements, keywords, common data types (signed and unsigned integer, bitwise Boolean, fixed point, and strings) numeric and logical operators, declaration statements, and variables and constants for all data types.” on page 1 for details on the syntax conventions used in this document.

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. However, if the text string is too long the entire message will not fit on the screen. The amount of display space will vary from camera to camera.

Display

This command allows the script to place feedback text in the liquid crystal display (LCD) 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.

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: iStatus  
...  
DisplayLine ("Image Processing completed: ", iStatus)
```

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 image file 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: nGName  
EndCapture (nGName)
```

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, nFilename)
```

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 (uFileId)
```

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: uFileId
FileOpen (... , uFileId)
...
FileClose (uFileId)
```

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 DigitaScript 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 (uDriveNum, sPath/FileName, uFileMode, uFileId)
```

PARAMETER DEFINITIONS

| Parameter Name | Type | Description |
|----------------|------|--|
| DriveNo | u | The drive on which the file will reside. Currently 2 is all that is available and 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 DigitaScript 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 DigitaScript 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: tSource
declare s: sFind
declare u: uStart, uEnd
declare i: iLocation
source = "washington"
find="ton"
start=0
FindString(tSource, uStart, sFind, iLocation)
FindString(tSource, 0, "ton", iLocation)
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.

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 79 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 79 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: uAelk
GetCameraDefault (0, "aelk", uAelk)
```

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.

SYNTAX

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

PARAMETER DEFINITIONS

| Parameter Name | Type | Description |
|----------------|---------------------|--|
| PName | s | Specifies the camera parameter requested. "Camera Capabilities Parameters" starting on page 79 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 79 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: uHint
GetCameraState ("hint", uHint)
```

GetCameraStatus

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

SYNTAX

```
declare b: bSystemstat, bCapturestat, bVendstat
GetCameraStatus (bSystemstat, bCapturestat, bVendstat)
```

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 | <code>ed1a</code> | Expansion disk 1 is installed and available |
| 2 | <code>ed2a</code> | Expansion disk 2 is installed and available |
| 3 | <code>ramda</code> | RAM disk is installed and available |
| 4 | <code>ipip</code> | Image Processing In Progress If this bit is set, images are still being processed. |
| 5 | <code>memf</code> | 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 | <code>pwra</code> | Power alert indicates that the camera has entered a low-power state because of low batteries. |
| 7 | <code>flsc</code> | 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 | <code>trun</code> | 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 | <code>stst</code> | Self test is in progress |
| 10 | <code>stsc</code> | Self test is complete |
| 11 | <code>cerr</code> | 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 | <code>mcro</code> | 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 | <code>sbs1</code> | 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 | <code>sbs2</code> | 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 | <code>zoom</code> | Zooming indicates that you are currently zooming the camera manually. This bit reflects the real-time status of the zoom buttons. |
| 16 | <code>extp</code> | External power indicates that external power is being applied to the camera. |
| 17 | <code>edch</code> | 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 | <code>rsvd</code> | The rest of these bits are reserved |

The second bit field, `CaptureStatus`, contains 32 separate logical flags. Flags two and three 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 | <code>rsvd</code> | This bit is reserved. |
| 2 | <code>strc</code> | Strobe charging indicates that the strobe subsystem is charging. |
| 3 | <code>strr</code> | Strobe ready indicates that the strobe is charged and ready to fire. |
| 4-32 | <code>rsvd</code> | These bits are reserved. |

The third bit field, `VendorStatus`, contains 32 separate logical flags. These bit flags are defined by each individual vendor. At the current time, none of the `VendorStatus` flags are used, but `VendorStatus` must still be part of the `GetCameraStatus` command. 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.

EXAMPLE

```
declare b: bSystemstat, bCapturestat, bVendstat
GetCameraStatus (bSystemstat, bCapturestat, bVendstat)
```

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 type “enumerated list”, 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 |
|----------------------------|----------------|---|
| <code>parameter-tag</code> | <code>s</code> | Camera parameter whose capability list count is requested. |
| <code>list-count</code> | <code>u</code> | The number of items defined within the list. This value will be zero if the <code>parameter-tag</code> is invalid or if the capability type for the parameter is not a list. |
| <code>default-value</code> | <code>u</code> | 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: uCount, uValue
GetCapabilitiesCount ("wmod", uCount, uValue)
```

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: sDesc  
declare u: uValue  
GetCapabilitiesListItem ("wmod", 3, sDesc, uValue)
```

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. |

- 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.
- 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: iMin, iMax, iDefault  
GetCapabilitiesRange ("wbcb", iMin, iMax, iDefault)
```

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: sCvalue  
GetCapabilitiesValue ("cmne", sCvalue)
```

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: uCuptype  
GetCapabilityType ("wbcb", uCuptype)
```

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 219 and Table 9 on page 219. 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 7 on page 219, Table 8 on page 219 and Table 9 on page 219 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 (filecount)
```

PARAMETER DEFINITION

| Parameter Name | Type | Description |
|----------------|------|-----------------------------|
| filecount | 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 |
|-------------------|---|
| <code>ptyp</code> | Picture type. The value 1 indicates full size; the value 0 indicates thumbnail. |
| <code>piav</code> | Partial image available. The value 1 is true; 0 is false. |
| <code>ipcm</code> | Image processing complete. The value 1 is true; 0 is false. |
| <code>piwm</code> | Picture in working memory. The value 1 is true; 0 is false. |
| <code>prmd</code> | Picture in internal storage. The value 1 is true; 0 is false. |
| <code>ped1</code> | Picture in expansion disk 1. The value 1 is true; 0 is false. |
| <code>ped2</code> | 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 image file tag.

SYNTAX

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

PARAMETER DEFINITIONS

| Parameter Name | Type | Description |
|---------------------------|---------------------|---|
| <code>path</code> | s | The path of the file received by means of the <code>GetFileInfo</code> or <code>GetNewFileInfo</code> command. |
| <code>file-name</code> | n | The DOS name of the file received by means of the <code>GetFileInfo</code> or <code>GetNewFileInfo</code> command. |
| <code>tag</code> | s | Specifies which image file tag is requested. “Camera Capabilities Parameters” starting on page 79 lists all the parameters currently available. |
| <code>return-value</code> | varies ^a | Contains the item requested in the image file tag parameter. Refer to “Camera Capabilities Parameters” starting on page 79 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: uWmrc
GetFileTag ("CAMERA01/", "IM000001.JPG", "wmrc", uWmrc)
```

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 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:

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

| | |
|-------------------|---|
| <code>ttyp</code> | Thumbnail Type (a value of 4 indicates YCC 422) |
| <code>hpix</code> | Thumbnail pixels horizontal |
| <code>vpix</code> | Thumbnail pixels vertical |
| <code>fsiz</code> | 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: sFile_path
declare n: nFile_name
GetMarkedImage (1, sFile_path, nFile_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 after a `StartCapture` command has finished processing. 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. It is important to complete the image processing before executing this command. |

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.

After the `StartCapture` command has finished processing, 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: sPath
declare n: nName
declare u: uLength
declare b: bStatus
GetNewFileInfo (3, sPath, nName, uLength, bStatus)
```


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> . For more information on <code>SetOption</code> , refer to page 49. |

EXAMPLE

```
declare u: uChoice  
GetOption (uChoice)
```

GetPowerMode

This command retrieves the camera's current power level.

SYNTAX

```
GetPowerMode (PowerState)
```

PARAMETER DEFINITION

PowerState Value contains unsigned integers from 5 to 1 that indicate the power state of the camera. Not all states are implemented on all cameras. 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: uPower-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 65.

SYNTAX

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

PARAMETER DEFINITIONS

| Parameter Name | Type | Description |
|----------------|---------------------|---|
| PName | s | Specifies the product information parameter requested. “Product Information Parameters” starting on page 65 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 65 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: uStrobe-count
GetProductInfo ("scnt", uStrobe-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 GetSystemFileInfo 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: sScript-name  
GetScriptName ("TEST_01.CSM", sScript-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: 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. Current resolution and compression settings will affect the count. |
| 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: uTaken, uAvailable  
declare i: iRaw-count  
GetStorageStatus (1, uTaken, uAvailable, iRaw-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: sFeedback  
GetString ("Enter desired value:", sFeedback)
```

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: uSystem-files  
GetSystemFileCount (uSystem-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: nFile-name  
GetSystemFileName (3, nFilename)
```

GetTimeString

This command returns a string with the time formatted as specified in Table 7 on page 219 and Table 8 on page 219. 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: sTime  
GetTimeString (sTime)
```

MakeFolder

Note: This command is only recognized by products compatible with DigitaScript 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 DigitaScript 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 DigitaScript 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 DigitaScript 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 DigitaScript 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

If you have a text file that contains data in the form of: 13, 15, 2.34, 1, roger, the Read command syntax would resemble the following:

```
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 DigitaScript 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. Leaves the pointer at the beginning of the next line.

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

SYNTAX

```
ReadLine(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. This effect is not persistent through a power cycle. To make it persistent, use SetCameraDefault. 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 DigitaScript 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 79 lists all the parameters currently available. |

EXAMPLE

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


RunApp

This command powers down the camera. When restarted, it uses the selected .CAM file. 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. The total characters must match the serial send/receive size.

Warning! If more characters are received than specified, the extra characters may be added to other serial send/receive commands.

***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: sReceive-data  
SerialReceive (sReceive-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. The total characters must match the serial send/receive size.

Warning! If more characters are received than specified, the extra characters may be added to other serial send/receive commands.

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. |

| Parameter Name | Type | Description |
|----------------|--------|---|
| 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: sReceive-data
SerialSendReceive (6, 6, "test02", sReceive-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: This command only works with parameters that are stored to EEPROM (the cameras permanent memory).

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 79 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.

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 ("aelk", 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)
```

(Possible values are, `still`, `group`, and `tlap` (time lapse))

EXAMPLE

```
SetCaptureMode (still|group|tlap)
```

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 DigitaScript 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 (mode)
```

(Possible values are comma, whitespace, tab, and 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. Use of this command can shorten up your code.

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)
GetOption (uLabel)
```

These values can also be used to set parameters as shown in the following example:

```
SetOption (120, "Two minute Battery Sleep", 0)
SetOption (240, "Four minute Battery Sleep", 0)
SetOption (600, "Ten minute Battery Sleep", 0)
GetOption (uTime)
SetCameraState ("bttc", uTime)
```

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 DigitaScript 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(title)
```

| Parameter Name | Type | Description |
|----------------|------|----------------------------|
| title | s, t | Establishes the title bar. |

EXAMPLE

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

SetUserFileTag

This command sets the value of the specified user image file tag. This applies to the tags that start with “u” from “Image File Tags” found in *Appendix C*. Only certain file tags are user writable (“u”). In order for this command to work properly, the image has to be completely done processing.

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 image file tag of the item the user wants to set. Refer to “Image File Tags” starting on page 177. |
| value | varies ^a | The value of the item the user wants to set. Refer to “Image File Tags” starting on page 177. |

- 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.

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 image file 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 image file 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 image size for capture. The range is 1 to 8. For sizes supported by your camera, see <i>ssiz</i> , <i>tsiz</i> and <i>bsiz</i> parameters using <i>GetCapabilities...</i> commands. |

EXAMPLE FOR TIME LAPSE CAPTURE

- DigitaScript V1.5 and V1.6 compatible

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

- Earlier versions of DigitaScript

```
StartCapture (2, 60, 1)
```

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

- 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. It is recommended to check status returned by the *StartCapture* command.

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.

Note: For more information on capturing images refer to “Capturing Images” on page 58.

SubString

Note: This command is only recognized by products compatible with DigitaScript 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: sSource, sOutput
declare u: uStart, uEnd
source = "washington"
start = 0
finish = 5
Substring (sSource, uStart, finish, sOutput)
Substring (sSource, 0, 15, sOutput)
```

StringToNumber

Note: This command is only recognized by products compatible with DigitaScript V1.5 or greater.

Converts a string to a numerical value.

SYNTAX

```
StringToNumber(String, Number)
```

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 | i, u, f, b | 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 DigitaScript 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 only “2” is used and it is 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 DigitaScript 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 (1000 milliseconds = 1 second). |

EXAMPLE

```
Wait (5000)
```

(5000 milliseconds = 5 seconds)

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)  
StartCapture ()
```

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, t | 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: uFileId  
declare i: iCount, iTotat  
FileOpen (... , fileId)  
Write (uFileId, "Processing image", iCount, "of", iTotat, ".")
```

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, t | 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.

This chapter describes the following scripting techniques:

- error checking
- capturing images
- startup scripts
- extracting bits from a bitfield
- watermark placement
- Japanese text usage in scripts

Error Checking

DigitaScript provides a convenient method of error checking. Every command statement returns an error status. This status can be used to determine whether an issued command succeeded or failed. If a command succeeds, an error status of 0 is returned. If the command fails, the error status will be a number which indicates the reason for failure. The error code returned will be a signed integer. Most often, the codes returned are positive integers.

To access the returned status code, simply call the command statement as part of a variable assignment. For example, the command:

```
declare i: status
status = 0
status = StartCapture ()
```

attempts to capture an image. If the StartCapture command is successful, the value of `status` will be 0. If unsuccessful, `status` will contain an error code that may be checked for various failure conditions.

Note that the status variable, `status`, is declared as a signed integer. The name of the status variable does not matter. This example uses the name “status” simply to clarify the purpose of the variable.

It is important that the script set the value of a status variable to 0 before using it. Not all commands return 0 when they execute successfully. This is because commands which can allow user input have to temporarily suspend execution of the script while waiting for input. The script then resumes at the line following that command and does not have the opportunity to return a value if it executed successfully. The commands for which this is true include `WaitForShutter()`, `StartCapture()`, `GetOption()`, `GetString()`, and `Wait()`. It is good practice to always set the status variable to 0 before using it. That way, it doesn't matter whether the successful value is returned. If an error occurs, it will be returned promptly.

It is important to check the error status of critical operations in scripts. Errors, other than syntax errors, should not cause the script to stop operation. For example, if a script tries to capture an image and is unsuccessful, the script will continue running as if the image had been captured.

The short example below demonstrates how to check for the success or failure of an image capture. The script attempts to capture an image. If it succeeds, it tells the user that it was successful. If it fails, it informs the user of the value of the error code.

```

name "Error test"
mode 0
menu "Sample scripts"
label "Error Test"

declare i: status
status = 0

SetCaptureMode (still)

status = StartCapture ( )

if status == 0
  DisplayLine ("Image capture succeeded")
  Wait (2000)
end

if status != 0
  DisplayLine ("Image capture failed. Error ", status)
  Wait (2000)
end

exitscript

```

Capturing Images

DigitaScript provides a mechanism for capturing images via a script. However, if the camera is not ready to capture an image and the script tries to do so, the script will continue running as if the image capture was successful. There are several methods that can be used to determine if the camera is ready to capture an image. In all cases, it is a good idea to check the error status of a StartCapture command to determine whether or not it was successful. Refer to “Error Checking” on page 57 for more details on error checking.

When a script attempts to capture images as rapidly as possible, there are two cases that need to be checked for. The first is capturing images under user control (the user presses the shutter button) and the second is capturing images automatically via script control.

When prompting the user to capture images, the quickest way to determine whether the camera is ready to capture new images is to attempt to issue a WaitForShutter() command. The camera will not successfully execute the WaitForShutter() command until it is ready to capture another image. The following code snippet checks for this:

```

declare i: status, SYSTEM_BUSY

# Set SYSTEM_BUSY equal to the error status for a system busy error.
SYSTEM_BUSY = 12

Pict1:
  # Clear the status variable before using it
  status = 0
  status = WaitForShutter ( "Picture of front of house" )
  if status == SYSTEM_BUSY
    DisplayLine ( "Processing Pictures..." )
    # Give camera time to process images before checking again
    Wait ( 5000 )
    goto Pict1
  end
  if status != 0
    # Some other error occurred. Additional error checking could occur

```



```

DisplayLine ( "Insufficient disk space.", status )
    goto Done
end
SetCaptureMode (still)
StartCapture ( )

```

This particular code snippet just checks for a system busy error. This error will occur if the camera is busy processing images and isn't yet ready to capture another image. If a system busy error occurs, it waits 5 seconds then attempts to issue a WaitForShutter() command again. This method is illustrated again in the CAPTURE.CSM example script which is part of the DigitaScript SDK. It may be useful to check for other errors based on what is likely to occur during the execution of a particular script.

In the case where a script is automatically capturing images as rapidly as possible, another method must be used. If you use the above method, the camera will wait until the user presses the shutter button to capture the next image. This obviously won't work when the camera is supposed to capture images automatically. In this case, the best method is to check for the ipip bit of the SystemStatus bitfield to be clear. The SystemStatus bitfield is returned by the GetCameraStatus() command. This bit will be cleared when the camera has finished processing all images.

This method is slower than checking for WaitForShutter(), and so it should not be used when capturing images manually. The reason that it is slower is that most Digita-based cameras are capable of capturing more than one image at a time. If the script checks for the ipip bit to be cleared, it will not be able to capture another image until all processing of the first image is completed. If the WaitForShutter() check is used, the camera will be able to capture another image as soon as possible, even though the camera may still be processing the first image.

The following code snippet illustrates this method. The technique used here for extracting a specific bit from a bitfield is discussed elsewhere in this guide.

```

declare u: IPIP, processing
declare b: systemStatus, captureStatus, vendorStatus

# set IPIP equal to the bitmask needed to extract the ipip bit from the
# SystemStatus bitflag returned by GetCameraStatus()
IPIP = 0x10000000

Pict1:
    # Retrieve camera status information, including ipip bit.
    GetCameraStatus ( systemStatus, captureStatus, vendorStatus )
    processing = systemStatus & IPIP

    if processing == IPIP
        DisplayLine ( "Processing pictures..." )
        # Give camera time to process images before checking again
        Wait ( 5000 )
        goto Pict1
    end

SetCaptureMode (still)
StartCapture ( )

```

Startup Scripts

Digita has the capability of running a script automatically when a Digita-based camera first starts up. Such a script is known as a startup script. The basic requirement of a startup script is that it be named STARTUP.CSM. The script itself is otherwise identical to other scripts.

One important issue to be aware of is that in order for a startup script to be executed properly, the camera must be already set in the operating mode specified by the script before powering on the camera. Otherwise, the script will abort when the camera sets the operating mode. This happens right before the user gains control of the camera when it is starting up.

Note that if a Kodak[®] DC220 or DC260 running a startup script has firmware 1.0.2 or older, the startup script will always abort. This was fixed in firmware versions 1.0.3 and later. This issue should not affect other cameras.

Extracting Bits From a Bitfield

Some information within Digita-based cameras is stored in bitfields. In particular, the data returned by the DigitaScript command GetCameraStatus() is stored in bitfields. The bitfields used are all variables declared with type 'b'. This kind of variable holds 32 pieces, or bits, of information. Each bit is equal to either 0 or 1. If a bit is equal to 1, it is said to be set. If it is equal to 0, it is not set.

When a value is returned in a bitfield, a simple calculation needs to be performed to extract a particular bit from the bitfield. This calculation is performing an AND on the bitfield with a specific number. The number varies depending on which bit needs to be extracted from the bitfield. Understanding how this works is not important if you are unfamiliar with programming techniques. What is important is understanding how to use this procedure.

Two pieces of information are required in order to use this method. The first is the number used to AND the bitfield. This number is sometimes called a bitmask. Appendix E contains a handy listing of the bitmasks needed for each position of a bitfield. The second piece of information needed is the position of the bit needed within the bitfield. This will be a number from 1 to 32. For the purposes of DigitaScript, position 1 is the leftmost bit in the bitflag. This is sometimes called the Most Significant Bit (MSB). Position 32, or rightmost bit, is sometimes called the Least Significant Bit (LSB).

Once this information is obtained, simply AND them using the & operator. For example, use the following example script to determine whether the first bit of the bitfield 'testField' is set. When run "The bit is not set!" will be displayed.

```
# This script appears as the item "Bitmask"
# in the "Sample Scripts" menu.

name "bit set test"
mode 0
menu "Sample Scripts"
label "Bitmask"

declare b: testField
declare u: bitmask, result

# Set testField to some arbitrary number for this example
testField = 1

# Set bitmask equal to the appropriate number from table in
# Appendix E of the Script Reference
bitmask = 0x10000000

# Perform the AND of the two numbers and check to see if the bit is set
result = testField & bitmask

if result == bitmask
DisplayLine ( "The bit is set!" )
Wait ( 3000 )
end
```

```

if result == 0
DisplayLine ( "The bit is not set!" )
Wait ( 3000 )
end
exitscript

```

Watermark Placement

Placing watermarks on an image is a useful feature of Digita-based cameras. The method used to determine where a watermark is placed on an image requires a little bit of explanation.

When specifying where a particular watermark element should be placed on an image, a number from 0 to 100 is used for both the x and y positions. This number is the percentage of total distance from a specific corner. The x axis increases to the right and the y axis increases to the left. So, the upper left of the image is 0,0 and the lower right is 100,100.

When specifying the position of a watermark element, the following rules apply:

- for x or y percentages less than 50%, the number represents the percentage distance from the top left
- for x or y percentages greater than 50%, the number represents the percentage distance from the bottom right
- for x or y percentages equal to 50%, the element is centered on that axis.

The reason for this is that when starting at 0,0 the left/top corner of the watermark element is positioned in the upper left of the image. If the position increased in this way past the mid-point of a particular axis, the element might run off the edge of the screen on the right or bottom edge. To prevent this, when the percentage passes the mid-point, the camera starts calculating distance from the right/bottom corner of the watermark element. So, placing a text watermark at position 51,51 would put the right/bottom corner of the text just below and to the right of the center of the image.

This following example script places the text “Welcome to FlashPoint!” at the top-center of an image.

```

# This script appears as the item "Watermark"
# in the "Sample Scripts" menu.

name "Setting the default location for watermarks."
mode 0
menu "Sample Scripts"
label "Watermark"

declare u: wena, status

status = 0

# Set the bit mask to the value required to turn the watermark text on.
wena = 0x40000000

# Set the error checking variable to the return value from SetCameraState.

status = SetCameraState ( "wena", wena )

# If an error occurs, wait 3 seconds then exit.
if status != 0
DisplayLine ( "Error enabling watermarks: ", status )
Wait ( 3000 )
goto Done
end

```

```

# Set the horizontal and vertical position of the string watermark
# to (50,0).
# Set the watermark string to "Welcome to FlashPoint!"

status = SetCameraState ( "wsxp", 50 )
status = SetCameraState ( "wsyp", 0 )
status = SetCameraState ( "wstr", "Welcome to FlashPoint!" )
status = SetCameraState ( "wsop", 1 )

# Inform camera user that the "Welcome to FlashPoint!" watermark is
# is enabled, then wait 3 seconds to continue.
DisplayLine ( "FlashPoint watermark enabled." )
Wait ( 3000 )

Done:
exitscript

```

Japanese Text Usage In Scripts

The following method can be used to display Japanese characters from a script. This will only work on cameras which contain Japanese support. U.S. versions of some cameras do not contain such support.

Some important points to be aware of in the use of Japanese text with DigitaScript are:

- Japanese characters must be entered as hexadecimal values within a string using the format `\x11`, where `\x` tells DigitaScript that an extended character is being used and `11` is any valid hexadecimal number between 1 and 255. A table showing the mapping of these numbers to characters in the standard camera font can be found in Appendix F.
- Japanese text cannot be placed in the script menu or label.
- Japanese text may only be used with the `Display`, `DisplayLine`, `SetOption`, and `Alert` commands.
- If the two bytes following the `\x` cannot be converted to a numeric value (e.g. `\x1f`) a syntax error will be displayed.
- In order for Japanese text to be displayed, the region code parameter (`rgnc`) must be set to 8.
- Japanese and English text can be mixed in the same string.

The following script is an example of how to use Japanese text in a script. If the camera the script is loaded on supports Japanese characters, "Hello world" will be displayed in Japanese characters.

```

# This script appears as the item "Japanese Text"
# on the menu "Example Scripts"

name "Japanese Text Test"
mode 0
menu "Example Scripts"
label "Japanese Text"

declare u: status

# Set to the Japanese region code
# and set up an error checking variable.
status = SetCameraState ( "rgnc", 8 )

# Camera supports Japanese characters.
if status == 0
DisplayLine ( "\x1A\x9A\x8A\xC3\xBD\xC4\x83\xDE\x1D\xA1" )
Wait ( 6000 )
end

```

```
# Camera does not support Japanese characters.  
if status != 0  
DisplayLine (" Japanese characters not supported.")  
Wait ( 6000 )  
end  
  
exitscript
```


Appendix A Product Information Parameters

The following tables define the product information (`ProductInfo`) parameters supplied in the `PName` parameter.

All of these parameters are required which means that they must be provided in all implementations of Digita.

The following terms are used throughout this appendix:

Value Terms:

Fixed (Digita) - Value is not alterable, and is determined by Digita.

Fixed (Product) - Value is not alterable, and is determined on a product-by product basis.

Non-persistent- Parameter is either calculated or returns to its factory default when power to the device is cycled

Persistent (sticky) - Parameter is stored in EEPROM, so a user default may be set which will be loaded instead of the factory default when power is cycled.

cacn– Application Copyright Notice

| | |
|---------------------|---|
| Definition: | Provides the copyright notice for the current control application |
| Data Type: | String |
| Commands to Access: | <code>GetProductInfo</code> |
| Value: | Fixed (Product) |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|---|---------|--------------------------------|-----------|
| DigitaScript and all Digita enabled cameras | | (c) 2000 FlashPoint Technology | ROM |

carv – Control Application Version

| | |
|---------------------|--|
| Details: | Defines the version for the currently running Application. Defined by the Application itself. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Non-persistent |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------------|---|-----------------|-----------|
| DigitaScript | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a | 0x01000000 | ROM |
| Kodak DC 220/DC 260/DC 265/DC 290 | | 0x01000000 | ROM |
| Dimage EX 1500 | | | ROM |
| HP PhotoSmart C500 | Version 1.0.2 | 0x10003000 | ROM |
| HP PhotoSmart C618/ C912 | Version 1.0.3 | 0x10003000 | ROM |
| Pentax EI-200/ EI-2000 | Version 1.0.3 | 0x10003000 | ROM |

cccn – CCS Copyright Notice

| | |
|---------------------|---|
| Definition: | The copyright notice for the current Camera Control Subsystem. |
| Data Type: | String |
| Commands to Access: | GetProductInfo |
| Value: | Fixed (Product) |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|---|--|-----------------------------------|-----------|
| DigitaScript and all Digita enabled cameras | Only required for camera products The value should be returned by the Camera Control Subsystem | (c) 2000 FlashPoint Technology | ROM |

ccsv – CCS Subsystem Version

| | |
|----------------------|--|
| Definition: | The version of the current Camera Control Subsystem. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Fixed (Product) |
| Access: | Read only |
| Associated File Tag: | ccsv |

| Product/OS | Details | Factory Default | Stored In |
|---|--|-----------------|-----------|
| DigitaScript and all Digita enabled cameras | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a | 0x01000000 | ROM |

cpus – Processor Speed

| | |
|---------------------|--|
| Definition: | Specifies the speed of the processor, defined by the hardware. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo |
| Value: | Fixed(Product) |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|-------------------|-----------------|-----------|
| DigitaScript | Units: 0.01 MHz | 4909 | ROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Units: 0.01 MHz | 6600 | ROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Units: 0.01 MHz | 4909 (49.09MHz) | ROM |
| HP PhotoSmart C618 | Units: 0.01 MHz | 9600 (96 MHz) | ROM |
| HP PhotoSmart C912 | Units: 0.01 MHz | 9600 (96 MHz) | ROM |
| Pentax EI-200 | Units: 0.01 MHz | 9600 (96 MHz) | ROM |
| Pentax EI-2000 | Units: 0.01 MHz | 9600 (96 MHz) | ROM |

cput– Processor Type

| | |
|---------------------|--|
| Definition: | Specifies the speed of the processor, defined by the hardware. |
| Data Type: | Enum |
| Commands to Access: | GetProductInfo |
| Value: | Fixed(Product) |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 1=MPC823 (Motorola 823), 2=DCAM 101/103, 3=Coldfire | 1 | ROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | 1=MPC823 (Motorola 823) | 1 | ROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | 1=MPC823 (Motorola 823), 2=DCAM 101/103 | 2 | ROM |
| HP PhotoSmart C618 | 1=MPC823 (Motorola 823), 2=DCAM 101/103, 3=Coldfire | | ROM |
| HP PhotoSmart C912 | 1=MPC823 (Motorola 823), 2=DCAM 101/103, 3=Coldfire | | ROM |
| Pentax EI-200 | 1=MPC823 (Motorola 823), 2=DCAM 101/103, 3=Coldfire | | ROM |
| Pentax EI-2000 | 1=MPC823 (Motorola 823), 2=DCAM 101/103, 3=Coldfire | | ROM |

exsa – Expansion Slots Available

| | |
|---------------------|--|
| Definition: | Specifies the number of expansion slots available for plug-in removable or other type cards. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo |
| Value: | Fixed(Product) |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--|---|-----------------|-----------|
| DigitaScript and all Digita enabled cameras | Value is expressed in number of slots available | 1 | ROM |

fwcn – Digita Copyright Notice

Definition: Specifies copyright for the Digita firmware
Data Type: String
Commands to Access: GetProductInfo
Value: Fixed (Digita)
Access: Read only

| Product/OS | Details | Factory Default | Stored In |
|---|---|--------------------------------------|-----------|
| DigitaScript and all Digita enabled cameras | Provides the copyright notice for the Digita OE currently running | (c) 2000 FlashPoint Technology, Inc. | ROM |

fwrdr – Digita Release Date

Definition: Specifies the release date for the installed version of the Digita OE.
Data Type: UInt
Commands to Access: GetProductInfo
Value: Fixed (Digita)
Access: Read only

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|-------------------|-----------------|-----------|
| DigitaScript | Units=seconds | | ROM |
| Kodak DC 220/DC 260/DC 265 | Tag not supported | | |
| Kodak DC 290 | Units=seconds | | |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Units=seconds | 0x368Cb8E6 | ROM |
| HP PhotoSmart C618 | Units=seconds | 0x39370710 | ROM |
| HP PhotoSmart C912 | Units=seconds | 0x39370710 | ROM |
| Pentax EI-200 | Units=seconds | 0x39370710 | ROM |
| Pentax EI-2000 | Units=seconds | 0x39370710 | ROM |

fwrv – Digita Firmware Version

| | |
|----------------------|-----------------------------------|
| Definition: | Specifies Digita firmware version |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Fixed (Digita) |
| Access: | Read only |
| Associated File Tag: | fwrv |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|--|-----------------|-----------|
| DigitaScript | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a | 0x01000600 | ROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | | 0x1000000 | ROM |
| Dimage EX 1500 | | 0x168886272 | ROM |
| HP PhotoSmart C500 | | 0x01000600 | ROM |
| HP PhotoSmart C618 | | 0x01000000 | ROM |
| HP PhotoSmart C912 | | 0x01000400 | ROM |
| Pentax EI-200 | | 0x01000000 | ROM |
| Pentax EI-2000 | | 0x01000400 | ROM |

hprs– Host PRAM Size

| | |
|---------------------|--|
| Definition: | Number of EEPROM bytes available for use by Digita Desktop |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo |
| Value: | Fixed (Product) |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|--|-------------------------|-----------------|-----------|
| DigitaScript and all Digita enable cameras | Host PRAM size in bytes | 32 | ROM |

hwrv – Hardware Version

| | |
|----------------------|---|
| Definition: | Specifies the version number for the hardware |
| Data Type: | UInt |
| Commands to Access | GetProductInfo, GetFileTag |
| Value: | Persistent |
| Access: | Read only |
| Associated File Tag: | hwrv |

| Product/OS | Details | Factory Default | Stored In |
|---|--|-----------------|-----------|
| DigitaScript and all Digita enabled cameras | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a | 0x01000000 | EEPROM |

iirv – Image Info Version

| | |
|----------------------|--|
| Definition: | Specifies the version number used for image information file tags. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Fixed (Digita) |
| Access: | Read only |
| Associated File Tag: | iirv |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|--|-----------------|-----------|
| DigitaScript | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a | 0x01000100 | ROM |
| Kodak DC 220/DC 260/DC 265 | Tag not supported | | |
| Kodak DC 290 | Same as for DigitaScript | 0x01000100 | ROM |
| Dimage Ex 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Same as for DigitaScript | 0x01000100 | ROM |
| HP PhotoSmart C618 | Same as for DigitaScript | 0x01000100 | ROM |
| HP PhotoSmart C912 | Same as for DigitaScript | 0x01000100 | ROM |
| Pentax EI-200 | Same as for DigitaScript | 0x01000100 | ROM |
| Pentax EI-2000 | Same as for DigitaScript | 0x01000100 | ROM |

ipcn – IPM Copyright Notice

Definition: Copyright for the image processing chain
Data Type: String
Commands to Access: `GetProductInfo`
Value: Fixed (Product)
Access: Read only

| Product/OS | Details | Factory Default | Stored In |
|--|---------|--------------------------------|-----------|
| DigitaScript and all Digita enable cameras | | (c) 2000 FlashPoint Technology | ROM |

iprn – IPM Subsystem Version

Definition: Version number for the image processing chain
Data Type: UInt
Commands to Access: `GetProductInfo`
Value: Fixed (Product)
Access: Read only

| Product/OS | Details | Factory Default | Stored In |
|--|--|-----------------|-----------|
| DigitaScript and all Digita enable cameras | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a | 0x01000000 | ROM |

oerv – Digita OE Version

| | |
|---------------------|---|
| Definition: | Specifies version of the Digita OE currently running. |
| Data Type: | U hex |
| Commands to Access: | GetProductInfo |
| Value: | Fixed (Digita) |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Format = 0xAABBCCDD for version AA.BB.CCDD Values are binary coded decimal (BCD), from 0 to 9, a to f For example, version 2.0.3.a would appear as 0x200030a 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 | | |
| Kodak DC 290 | Same as DigitaScript | 0x01000000 | ROM |
| Dimage Ex 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Same as DigitaScript | 0X01000100 | ROM |
| HP PhotoSmart C618 | Same as DigitaScript | 0x1000200 | ROM |
| HP PhotoSmart C912 | Same as DigitaScript | 0x1000200 | ROM |
| Pentax EI-200 | Same as DigitaScript | 0x1000200 | ROM |
| Pentax EI-2000 | Same as DigitaScript | 0x1000200 | ROM |

pcnt – Picture Taken Count

| | |
|----------------------|--|
| Definition: | The number of pictures taken by the camera |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Persistent |
| Access: | Read only |
| Associated File Tag: | pcnt (This file tag is supported by Digita Script 1.5 and above only.) |

| Product/OS | Details | Factory Default | Stored In |
|---|--|-----------------|-----------|
| DigitaScript and all Digita enabled cameras | Acts as an “odometer” for the camera Shipped from the factory with a value of 0 Value is incremental every time Digita takes a picture Maximum value of 1048575 | 0 | EEPROM |

ptid – Product Name

| | |
|----------------------|----------------------------|
| Definition: | Name of the product |
| Data Type: | String |
| Access: | Read only |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Fixed (Product) |
| 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 |
| HP PhotoSmart C500 | | HP PhotoSmart C500 | ROM |
| HP PhotoSmart C618 | | HP PhotoSmart C618 | ROM |
| HP PhotoSmart C912 | | HP PhotoSmart C912 | ROM |
| Pentax EI-200 | | PENTAX EI-200 | ROM |
| Pentax EI-2000 | | PENTAX EI-2000 | ROM |

rams – Application RAM Size

| | |
|---------------------|--|
| Definition: | The amount of RAM available for an alternate application. This includes application code and all heap space. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo |
| Value: | Non-persistent |
| 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 |
| HP PhotoSmart C500 | | 12582912 | ROM |
| HP PhotoSmart C618 | | 12000000 | ROM |
| HP PhotoSmart C912 | | 12000000 | ROM |
| Pentax EI-200 | | 12000000 | ROM |
| Pentax EI-2000 | | 12000000 | ROM |

scnt – Strobe Pictures Taken

| | |
|---------------------|--|
| Definition: | Number of strobe images captured with this camera. Acts as an “odometer” for the strobe subsystem. |
| Data Type: | UInt |
| Commands to Access: | GetProductInfo |
| Value: | Persistent |
| Access: | Read only |

| Product/OS | Details | Factory Default | Stored In |
|---|---|-----------------|-----------|
| DigitaScript and all Digita enabled cameras | Shipped from the factory with a value of 0 Count is increment 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 |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Persistent |
| Access: | Read only |
| Associated File Tag: | sern |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---------------------------|-----------------|-----------|
| DigitaScript | Maximum length of 32 bits | | |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Maximum length of 24 bits | XXXXXXXX | EEPROM |
| Dimage EX 1500 | Maximum length of 24 bits | XXXXXXXX | EEPROM |
| HP PhotoSmart C500 | Maximum length of 24 bits | XXXXXXXX | EEPROM |
| HP PhotoSmart C618 | Maximum length of 24 bits | XXXXXXXX | EEPROM |
| HP PhotoSmart C912 | Maximum length of 24 bits | XXXXXXXX | EEPROM |
| Pentax EI-200 | Maximum length of 24 bits | XXXXXXXX | EEPROM |
| Pentax EI-2000 | Maximum length of 24 bits | XXXXXXXX | EEPROM |

ucrv – Universal Command Set Version

| | |
|---------------------|---|
| Definition: | Specifies the version number of the host command set. |
| Data Type: | UInt - hex |
| Commands to Access: | GetProductInfo, SetUserFileTag |
| Value: | Fixed (Digita) |
| Access: | Read only |

| Product OS | Details | Factory Default | Stored In |
|---|--|-----------------|-----------|
| DigitaScript and all Digita enabled cameras | 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 |
| Commands to Access: | GetProductInfo, GetFileTag |
| Value: | Fixed (Product) |
| 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 |
| HP PhotoSmart C500 | Vendor Name | Hewlett Packard | ROM |
| HP PhotoSmart C618 | Vendor Name | Hewlett-Packard | ROM |
| HP PhotoSmart C912 | Vendor Name | Hewlett-Packard | ROM |
| Pentax EI-200 | Vendor Name | PENTAX | ROM |
| Pentax EI-2000 | Vendor Name | PENTAX | ROM |

Camera Capabilities Parameters

The tables on the following pages 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. (Custom Parameters are located at the end of this Appendix.)

The following terms are used throughout this appendix:

Value Terms:

Fixed (Digita) - Value is not alterable, and is determined by Digita.

Fixed (Product) - Value is not alterable, and is determined on a product-by product basis.

Non-persistent - Parameter is either calculated or returns to its factory default when power to the device is cycled

Persistent (sticky) - Parameter is stored in EEPROM, so a user default may be set which will be loaded instead of the factory default when power is cycled.

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.*

Standard Parameters

aagc – Analog Signal Gain Control

| | |
|----------------------|--|
| Definition: | Sets the analog signal gain from the image sensor. Used in low light situations (value measured in ASA rating) with the Programmed and Gain Priority exposure modes only (see xmod parameter). |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | aagc |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=100, Maximum=400, Units=ASA rating | 100 | EEPROM |
| Kodak DC 220/DC 260/DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=100, Maximum=100 | 100 | EEPROM |
| Dimage EX 1500 | Minimum=600, Maximum=2400 (version 2.0 only) | 800 | EEPROM |
| HP PhotoSmart C500 | Minimum=100, Maximum=400 | 100 | EEPROM |
| HP PhotoSmart C618 | Minimum=100, Maximum=400 | 100 | EEPROM |
| HP PhotoSmart C912 | Minimum=25, Maximum=400 | 100 | EEPROM |
| Pentax EI-200 | Minimum=100, Maximum=400 | 100 | EEPROM |
| Pentax EI-2000 | Minimum=25, Maximum=400 | 100 | EEPROM |

acpd – AC Power Down Timeout

| | |
|---------------------|---|
| Definition: | Sets the number of seconds to sleep with no user activity before powering down, when running from external power. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--|---|-----------------|-----------|
| DigitaScript | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/ DC 290 | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |
| Dimage EX 1500 | Minimum=5, Maximum=65535, Units=seconds | 1800 | EEPROM |
| HP PhotoSmart C500 | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |
| HP PhotoSmart C618 | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |
| HP PhotoSmart C912 | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |
| Pentax EI-200 | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |
| Pentax EI-2000 | Minimum=5, Maximum=65535, Units=seconds | 600 | EEPROM |

acse – AC Sleep Timeout Enable

| | |
|---------------------|---|
| Definition: | Indicates whether or not the AC sleep timeout is enabled. If enabled, sleep time is set by the parameter <code>actc</code> . If disabled, the power down timer runs without an intermediate sleep cycle. The third option is to never sleep when externally powered (no power down in this case). |
| Data Type: | Enum List |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------|---|-----------------|-----------|
| DigitaScript | 1=Enable, 2=Disable, 3=Never Sleep | 1 | EEPROM |
| Kodak DC 220/ DC 260/ DC 265 | The value zero (0) indicates disabled. This parameter is always disabled for this camera. | 0 | EEPROM |
| Kodak DC 290 | 1=Enable, 2=Disable, 3=Never Sleep | 2 | EEPROM |
| Dimage EX 1500 | The value zero (0) indicates disabled. This parameter is always disabled for this camera. | 0 | EEPROM |
| HP PhotoSmart C500 | 1=Enable, 2=Disable, 3=Never Sleep | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Enable, 2=Disable, 3=Never Sleep | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Enable, 2=Disable, 3=Never Sleep | 1 | EEPROM |
| Pentax EI-200 | 1=Enable, 2=Disable, 3=Never Sleep | 1 | EEPROM |
| Pentax EI-2000 | 1=Enable, 2=Disable, 3=Never Sleep | 1 | EEPROM |

actc – AC Power 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 <code>acse</code> parameter is set to off. |
| Data Type: | UInt |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=5, Maximum =65535, Units=seconds | 600 | EEPROM |
| Kodak DC 220/ DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=5, Maximum =65535, Units=seconds | 600 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Minimum=5, Maximum =65535, Units=seconds | 1800 | EEPROM |
| HP PhotoSmart C618 | Minimum=5, Maximum =65535, Units=seconds | 120 | EEPROM |
| HP PhotoSmart C912 | Minimum=5, Maximum =65535, Units=seconds | 120 | EEPROM |
| Pentax EI-200 | Minimum=5, Maximum =65535, Units=seconds | 120 | EEPROM |
| Pentax EI-2000 | Minimum=5, Maximum =65535, Units=seconds | 120 | EEPROM |

aeIk – AE Lock Mode

Definition: Auto Exposure (AE) Lock: Pressing the shutter release button partway down holds the exposure settings until 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

Commands to Access: GetCameraState, SetCameraState, GetFileTag

Value: Persistent

Access: Read/Write

Associated File Tag: lkmd

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|-------------------------|-----------------|-----------|
| DigitaScript | 1=Continuous, 2=AE Lock | 1 | EEPROM |
| Kodak DC 220/DC 265/ DC 265/DC 290 | 1=Continuous, 2=AE Lock | 2 | EEPROM |
| Dimage EX 1500 | 1=Continuous, 2=AE Lock | 2 | EEPROM |
| HP PhotoSmart C500 | 1=Continuous, 2=AE Lock | 2 | EEPROM |
| HP PhotoSmart C618 | 1=Continuous, 2=AE Lock | 2 | EEPROM |
| HP PhotoSmart C912 | 1=Continuous, 2=AE Lock | 2 | EEPROM |
| Pentax EI-200 | 1=Continuous, 2=AE Lock | 2 | EEPROM |
| Pentax EI-2000 | 1=Continuous, 2=AE Lock | 2 | EEPROM |

aflk – AF Lock Mode

| | |
|----------------------|---|
| Definition: | AF Lock: Pressing the shutter release button partway down holds the focus settings until 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 |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | lkmd |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|-----------------------------|-----------------|-----------|
| DigitaScript | Continuous=1, AF Locked=F=2 | 1 | EEPROM |
| Kodak DC 290 | Continuous AF not supported | 2 | EEPROM |
| Dimage EX 1500 | Continuous AF not supported | 2 | EEPROM |
| HP PhotoSmart C500 | Continuous=1, AF Locked=2 | 1 | EEPROM |
| HP PhotoSmart C618 | Continuous=1, AF Locked=2 | 2 | EEPROM |
| HP PhotoSmart C912 | Continuous=1, AF Locked=2 | 2 | EEPROM |
| Pentax EI-200 | Continuous=1, AF Locked=2 | 2 | EEPROM |
| Pentax EI-2000 | Continuous=1, AF Locked=2 | 2 | EEPROM |

aper – Aperture (F Number)

| | |
|----------------------|--|
| Definition: | This parameter sets the lens aperture setting to control exposure and depth of field. Value is in .01 f/stop increments, i.e., 300 = f/3. If the “Aperture Priority” or “Programmed” exposure mode is set (see the xmod parameter), this parameter directly controls the aperture value used to take the picture. In all other exposure modes, the aperture is under the control of the auto-exposure system. In all cases, the value is transferred to the file tag is the one actually used to take the picture. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| 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 |
|---------------------|---|-----------------|-----------|
| DigitaScript | Minimum=200, Maximum=3200, Units= 0.01 (e.g. F2.8 = 280) | 300 | EEPROM |
| Kodak DC 220 | Aperture is fixed at F4.0 | 400 | EEPROM |
| Kodak DC 260/DC 265 | Minimum= 300, Maximum=1600 | 300 | EEPROM |
| Kodak DC 290 | Minimum= 300, Maximum=1600 | 560 | EEPROM |
| Dimage EX 1500 | Aperture is fixed at F5.6 | 560 | EEPROM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Minimum=200, Maximum=1880 (e.g. F3 = 300, F5.6 = 560) Returned values are rounded to the nearest available setting, zoom position affects the allowed value | 260 | EEPROM |
| HP PhotoSmart C912 | Minimum=200, Maximum=1880 (e.g. F3 = 300, F5.6 = 560, F8.0=800, etc.) Returned values are rounded to the nearest available setting, zoom position affects the allowed value | 260 | EEPROM |
| Pentax EI-200 | Minimum=200, Maximum=1880 (e.g. F3 = 300, F5.6 = 560) Returned values are rounded to the nearest available setting, zoom position affects the allowed value | 260 | EEPROM |
| Pentax EI-2000 | Minimum=200, Maximum=1880 (e.g. F3 = 300, F5.6 = 560) Returned values are rounded to the nearest available setting, zoom position affects the allowed value | 260 | EEPROM |

bccl – Burst Capture Color Mode

| | |
|----------------------|--|
| Definition: | Sets the color mode for burst capture. Required if burst capture is supported. Parameter is transferred to the file tags for burst capture only. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | impn |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|--|-----------------|-----------|
| DigitaScript | 1=24 bit color, 2= Gray scale, 3=Sepia | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | 1=24 bit color | 1 | EEPROM |
| Dimage EX 1500 | 1=24 bit color, 2= Gray scale | 1 | EEPROM |
| HP PhotoSmart C500 | 1=24 bit color, 2= Gray scale | 1 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

bcpn – Burst Capture Compression

| | |
|----------------------|---|
| Definition: | Sets the compression mode for burst capture images. Higher compression means lower image quality. If some are not supported, then the closest available case should be substituted. Required if burst capture is supported. Tags only burst images. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | cmpn |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 1=Maximum, 2=High, 3=Normal, 4=Low, 5=Very Low, 6=Minimum, 7=Lossless | 2 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Settings 2, 3, and 4 are supported | 4 | EEPROM |
| Kodak DC 290 | Settings 2 (Good), 3 (Better), 4 (Best), and 7 (Uncompressed) are supported | 4 | EEPROM |
| Dimage EX 1500 | Settings 2 (Good), 3 (Better), 4 (Best), 5 (Super Fine) are supported | 4 | EEPROM |
| HP PhotoSmart C500 | Settings 2, 3, and 4 are supported | 3 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

bsiz – Burst Capture Size

| | |
|---------------------|---|
| Definition: | Sets the size of the image for burst capture. Required if burst capture is supported. Value sent to CCS via StartCapture command. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------|--|-----------------|-----------|
| DigitaScript | 1=Large Size, 2=Medium Size, 3=Small Size, and 4=Custom Size | 2 | EEPROM |
| Kodak DC 220 | Settings 1 and 3 are supported | 3 | EEPROM |
| Kodak DC 260/DC 265 | Settings 1, 2, and 3 are supported | 3 | EEPROM |
| Kodak 290 | 1=High (1792x1200), 2=Medium (1440x960), 3=Standard (720x480), 4=Ultra (2240x1500) | 3 | EEPROM |
| Dimage EX 1500 | Settings 1 and 2 are supported | 1 | EEPROM |
| HP PhotoSmart C500 | 1=High (1600x1200), 2=Medium (1152x864), 3=Low (640x480) | 1 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

btspd – Battery Power Down Timeout

| | |
|---------------------|--|
| Definition: | Sets the number of seconds to sleep on battery power before powering down. (<i>btse</i> must be enabled). |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--|--|-----------------|-----------|
| DigitaScript | Minimum: 5, Maximum: 65535, Units: seconds | 480 | EEPROM |
| Kodak DC 220/ DC 260/ DC 265/DC 290 | Minimum: 5, Maximum: 65535, Units: seconds | 480 | EEPROM |
| Dimage EX 1500 | Minimum: 5, Maximum: 65535, Units: seconds | 360 | EEPROM |
| HP PhotoSmart C500 | Minimum: 5, Maximum: 65535, Units: seconds | 480 | EEPROM |
| HP PhotoSmart C618 | Minimum: 5, Maximum: 65535, Units: seconds | 300 | EEPROM |
| HP PhotoSmart C912 | Minimum: 5, Maximum: 65535, Units: seconds | 300 | EEPROM |
| Pentax EI-200 | Minimum: 5, Maximum: 65535, Units: seconds | 300 | EEPROM |
| Pentax EI-2000 | Minimum: 5, Maximum: 65535, Units: seconds | 300 | EEPROM |

btse – Battery Sleep Timeout Enable

| | |
|---------------------|---|
| Definition: | Controls the sleep timer when operating from battery power. If enabled (1), the delay to sleep is controlled by the parameter <code>bttc</code> . If disabled, the power down timer runs without an intermediate sleep cycle. |
| Data Type: | Enum |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Definition | Factory Default | Stored In |
|--------------------------------------|--------------------------------|-----------------|-----------|
| DigitaScript | 1=Enable, 2=Disable | 1 | EEPROM |
| Kodak DC 220/DC260/ DC 265/DC 290 | Setting 1 supported | 1 | EEPROM |
| Dimage EX 1500 | Enable value supported | 1 | EEPROM |
| HP PhotoSmart C500 | 1=Enable, 2=Disable | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Enable (on), 2=Disable (off) | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Enable (on), 2=Disable (off) | 1 | EEPROM |
| Pentax EI-200 | 1=Enable (on), 2=Disable (off) | 1 | EEPROM |
| Pentax EI-2000 | 1=Enable (on), 2=Disable (off) | 1 | EEPROM |

btcc – Battery Sleep Timeout

| | |
|---------------------|---|
| Definition: | Sets the number of seconds of inactivity before the camera enters sleep mode, assuming battery power is being used. This parameter is not applied to Capture mode when the LCD is off (btse must be enabled). |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum: 5, Maximum: 65535, Units: seconds | 120 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum: 5, Maximum: 65535, Units: seconds | 120 | EEPROM |
| Kodak DC 290 | Minimum: 5, Maximum: 65535, Units: seconds | 60 | EEPROM |
| Dimage EX 1500 | Minimum: 5, Maximum: 65535, Units: seconds | 180 | EEPROM |
| HP PhotoSmart C500 | Minimum: 5, Maximum: 65535, Units: seconds | 30 | EEPROM |
| HP PhotoSmart C618 | Minimum: 5, Maximum: 65535, Units: seconds | 120 | EEPROM |
| HP PhotoSmart C912 | Minimum: 5, Maximum: 65535, Units: seconds | 120 | EEPROM |
| Pentax EI-200 | Minimum: 5, Maximum: 65535, Units: seconds | 120 | EEPROM |
| Pentax EI-2000 | Minimum: 5, Maximum: 65535, Units: seconds | 120 | EEPROM |

camf – Camera Folder Name

| | |
|---------------------|---|
| Definition: | The name of the folder where images are stored. |
| Data Type: | DOS Filename |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|--|-----------------|-----------|
| DigitaScript | 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 | | DC260_ | EEPROM |
| Kodak DC 220 | | DC220_ | EEPROM |
| Dimage EX 1500 | | EX15_ | EEPROM |
| HP PhotoSmart C500 | | HP500_ | EEPROM |
| HP PhotoSmart C618 | | CAMERA | EEPROM |
| HP PhotoSmart C912 | | CAMERA | EEPROM |
| Pentax EI-200 | | CAMERA | EEPROM |
| Pentax EI-2000 | | CAMERA | EEPROM |

cmne – User-defined Camera Name

| | |
|---------------------|--------------------------------|
| Definition: | The user-defined camera name |
| Data Type: | String |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|--|-----------------|-----------|
| DigitaScript | A maximum of 31 characters can be entered, but no error will occur if more characters are entered, the string will be shortened. | Digita Camera | EEPROM |
| Kodak DC 290 | | DC290 | EEPROM |
| Kodak DC 265 | | DC265 | EEPROM |
| Kodak DC 260 | | DC260 | EEPROM |
| Kodak DC 220 | | DC220 | EEPROM |
| HP PhotoSmart C500 | | HP C500 | EEPROM |
| HP PhotoSmart C618 | | Digita Camera | EEPROM |
| HP PhotoSmart C912 | | Digita Camera | EEPROM |
| Pentax EI-200 | | Digita Camera | EEPROM |
| Pentax EI-2000 | | Digita Camera | EEPROM |

cppt– 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 |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag, SetUserFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | ucpt |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | | "" | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Can be 31 characters, the default value is the null string. The user copyright notice goes into a tag in the image information. | "" | EEPROM |
| Kodak DC 290 | Tag not supported. Use ucpt instead | | |
| Dimage EX 1500 | Can be 31 characters, the default value is the null string. The user copyright notice goes into a tag in the image information. | "" | EEPROM |
| HP PhotoSmart C500 | Tag not supported. Use ucpt instead | | |
| HP PhotoSmart C618 | Tag not supported. Use ucpt instead | | |
| HP PhotoSmart C912 | Tag not supported. Use ucpt instead | | |
| Pentax EI-200 | Tag not supported. Use ucpt instead | | |
| Pentax EI-2000 | Tag not supported. Use ucpt instead | | |

crst– Image Counter Mode

| | |
|---------------------|---|
| Definition: | The filename for each image incorporates a number to ensure that it is unique. This parameter controls how that number is determined. It sets the counter to roll or odometer mode. In roll mode, the counter resets to one when the disk is emptied the odometer mode uses the absolute image count taken by the camera. |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------------|--|-----------------|-----------|
| DigitaScript | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | EEPROM |
| Kodak DC 220/DC260/ DC 265/DC 290 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 1 | EEPROM |
| Dimage EX 1500 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | EEPROM |
| HP PhotoSmart C500 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | EEPROM |
| HP PhotoSmart C618 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | EEPROM |
| HP PhotoSmart C912 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | EEPROM |
| Pentax EI-200 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | EEPROM |
| Pentax EI-2000 | 1=Roll Count (reset when empty), 2=Odometer Count (continuous counter) | 2 | 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. |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|------------------------------------|-----------------|-----------|
| DigitaScript | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 1 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 1 | EEPROM |
| HP PhotoSmart C618 | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 1 | EEPROM |
| HP PhotoSmart C912 | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 1 | EEPROM |
| Pentax EI-200 | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 3 | EEPROM |
| Pentax EI-2000 | 1=MM/DD/YY, 2=DD/MM/YY, 3=YY/MM/DD | 3 | EEPROM |

dfmt – Date Display Format

| | |
|---------------------|--|
| Definition: | Specifies the display format for dates in Watermarks, or any other use of longer date formats. |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| Dimage EX 1500 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| HP PhotoSmart C500 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| HP PhotoSmart C618 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| HP PhotoSmart C912 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| Pentax EI-200 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |
| Pentax EI-2000 | 1=short date, 2=abbreviated date, 3=long date | 1 | EEPROM |

fbti – Burst Time Interval

| | |
|---------------------|--|
| Definition: | Specifies the time interval between captures for burst capture. Required if burst capture is supported. Value is sent to CSS via StartCapture command. Value is in milliseconds. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------|----------------------------|-----------------|-----------|
| DigitaScript | Minimum=100, Maximum=1000 | 250 | EEPROM |
| Kodak DC 220 | Minimum=320, Maximum=10000 | 320 | EEPROM |
| Kodak DC 260/DC 265 | Minimum=320, Maximum=10000 | 360 | EEPROM |
| Kodak DC 290 | Minimum=500, Maximum=10000 | 500 | EEPROM |
| Dimage EX 1500 | Minimum=30, Maximum=10000 | 266 | EEPROM |
| HP PhotoSmart C500 | Minimum=333, Maximum=1000 | 500 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

fdst – Focus Distance

| | |
|----------------------|--|
| Definition: | Specifies the focus distance for Programmed focus mode. Set by the fmod parameter. Transferred to file tag via CSS value only if programmed focus is selected. Units are in centimeters. |
| Data Type: | UInt |
| Access: | Read/Write |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Associated File Tag: | fdst |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum=30, Maximum (infinity) =-65535 | 1000 | EEPROM |
| Kodak DC 220 | Fixed Focus Camera - Tag not supported | | |
| Kodak DC 260/DC 265/ DC 290 | Minimum=50, Maximum (infinity) =-65535 | 300 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Minimum=10, Maximum (infinity) =-65535 units = centimeters | 1000 | EEPROM |
| HP PhotoSmart C912 | Minimum=2, Maximum (infinity) =-65535 units = centimeters | 1000 | EEPROM |
| Pentax EI-200 | Minimum=10, Maximum (infinity) =-65535 units = centimeters | 1000 | EEPROM |
| Pentax EI-2000 | Minimum=2, Maximum (infinity) =-65535 units = centimeters | 1000 | EEPROM |

flty – File Type

| | |
|----------------------|---|
| Definition: | Specifies the file type when capturing still images (files with only one image in them). The enumeration list is updated (extended) when a plug-in FEM is loaded. |
| Data Type: | Enum List |
| Access: | Read/Write |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Associated File Tag: | flty |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 1=JPEG, 2=EXIF, 3=FPIX, 4=TIFF | 2 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | FPIX=0x46505820, JPEG=0x4A504547 Associated file tag: imcn | 0x4A504547 | EEPROM |
| Kodak DC 290 | 2=EXIF, 4=TIFF | 2 | EEPROM |
| Dimage EX 1500 | JPEG=0x4A504547 Associated file tag: imcn | 0x4A504547 | EEPROM |
| HP PhotoSmart C500 | 2=EXIF | 2 | EEPROM |
| HP PhotoSmart C618 | 2=JPEG, 4=TIFF | 2 | EEPROM |
| HP PhotoSmart C912 | 2=JPEG, 4=TIFF | 2 | EEPROM |
| Pentax EI-200 | 2=JPEG, 4=TIFF | 2 | EEPROM |
| Pentax EI-2000 | 2=JPEG, 4=TIFF | 2 | EEPROM |

fmod – Focus Mode

| | |
|----------------------|--|
| Definition: | Specifies the camera's focus mode. |
| Data Type: | Enum List |
| Access: | Read/Write |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Associated File Tag: | fmod |

| Product | Definition | Factory Default | Stored In |
|--------------------------------|-------------------------------|-----------------|-----------|
| DigitaScript | 1=Auto, 2=Program, 3=Manual | 1 | RAM |
| Kodak DC 220 | Tag not supported | | RAM |
| Kodak DC 260/DC 265/ DC 290 | Setting 1 and 3 are supported | 1 | RAM |
| Dimage EX 1500 | Setting 1 and 3 are supported | 1 | RAM |
| HP PhotoSmart C500 | 1=Auto | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Auto, 2=Manual Focus | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Auto, 2=Manual Focus | 1 | EEPROM |
| Pentax EI-200 | 1=Auto, 2=Manual Focus | 1 | EEPROM |
| Pentax EI-2000 | 1=Auto, 2=Manual Focus | 1 | EEPROM |

fmod – Focus Method

| | |
|----------------------|---|
| Definition: | Specifies the scene data to be used to compute the focus point. Selection is not applicable for Programmed or Manual focus mode set by <code>fmod</code> parameter. |
| Data Type: | Enum List |
| Access: | Read/Write |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> , <code>GetFileTag</code> |
| Value: | Persistent |
| Associated File Tag: | fmod |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | 1=Auto Matrix, 2=Center Spot, 3=Multi Spot, 4=Center Weighted, 5=Custom Matrix | 1 | RAM |
| Kodak DC 220 | Tag not supported | | |
| Kodak DC 260/DC 265/ DC 290 | Settings 1 and 2 are supported | 1 | RAM |
| Dimage EX 1500 | Setting 2 is supported | 2 | RAM |
| HP PhotoSmart C500 | Setting 1 is supported | 1 | EEPROM |
| HP PhotoSmart C618 | Setting 4 is supported | 4 | EEPROM |
| HP PhotoSmart C912 | Settings 2 and 4 are supported | 4 | EEPROM |
| Pentax EI-200 | Setting 4 is supported | 4 | EEPROM |
| Pentax EI-2000 | Settings 2 and 4 are supported | 4 | EEPROM |

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 |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | hint |

| Product/OS | Definition | Factory Default | Stored In |
|---------------------------|--|-----------------|-----------|
| DigitaScript | 1=Auto, 2=Portrait, 3=landscape,4=Closeup, 5=Sports, 6=Evening, 7=Night and 8=Sow-Sync, 9=Panorama | 1 | RAM |
| Kodak DC 220/DC 260/DC265 | Tag not supported | | |
| Kodak DC 290 | 1=Auto, 2=Portrait, 3=landscape,4=Closeup, 5=Sports, 6=Evening, 7=Night and 8=Sow-Sync, 9=Panorama | 1 | RAM |
| Dimage EX 1500 | Settings 1, 2, 3, 6, 7, and 8 are supported. Setting 9 supported by the 3D1500 only | 1 | RAM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Settings 1, 2, 3, and 5 are supported | 1 | RAM |
| HP PhotoSmart C912 | Settings 1, 2, 3, 5 and 7 are supported | 1 | RAM |
| Pentax EI-200 | Settings 1, 2, 3, and 5 are supported | 1 | RAM |
| Pentax EI-2000 | Settings 1, 2, 3, 5 and 7 are supported | 1 | RAM |

ihhc – 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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--------------------------------------|-----------------|-----------|
| DigitaScript | 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 |
| HP PhotoSmart C500 | Minimum:2, Maximum=30, Units=seconds | 8 | EEPROM |
| HP PhotoSmart C618 | Minimum:0, Maximum=30, Units=seconds | 8 | EEPROM |
| HP PhotoSmart C912 | Minimum:0, Maximum=30, Units=seconds | 8 | EEPROM |
| Pentax EI-200 | Minimum:0, Maximum=30, Units=seconds | 8 | EEPROM |
| Pentax EI-2000 | Minimum:0, Maximum=30, Units=seconds | 8 | EEPROM |

irev – Instant Review Duration

| | |
|---------------------|--|
| Definition: | Sets the duration of Instant Review after capture. If set to zero, Instant Review is off and Live View continues asap after capture is complete. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read /Write |

| Product/OS | Details | Factory Default | Stored In |
|--|---|-----------------|-----------|
| DigitaScript | Minimum=0, Maximum=3000, Units=0.01 seconds | 300 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/ DC 290 | Minimum=0, Maximum=3000, Units=0.01 seconds | 300 | EEPROM |
| Dimage Ex 1500 | Minimum=0, Maximum=10000 | 1000 | EEPROM |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|---|-----------------|-----------|
| HP PhotoSmart C500 | Minimum=0, Maximum=3000, Units=0.01 seconds | 0 | EEPROM |
| HP PhotoSmart C618 | Minimum=0, Maximum=3000, Units=0.01 seconds | 300 | EEPROM |
| HP PhotoSmart C912 | Minimum=0, Maximum=3000, Units=0.01 seconds | 300 | EEPROM |
| Pentax EI-200 | Minimum=0, Maximum=3000, Units=0.01 seconds | 300 | EEPROM |
| Pentax EI-2000 | Minimum=0, Maximum=3000, Units=0.01 seconds | 300 | EEPROM |

isam – Image Sharpness Amount

| | |
|----------------------|---|
| Definition: | Controls sharpening performed when the image is captured. |
| Data Type: | Signed Int |
| Access: | Read /Write |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Associated File Tag: | isam |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|---|-----------------|-----------|
| DigitaScript | 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 220/DC 260/DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=-25, Maximum=100 Negative values indicate softening and can be used in conjunction with sharpening parameters for full unsharp mask. | 50 | EEPROM |
| Dimage Ex 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

isra – Image Sharpness Radius

| | |
|---------------------|--|
| Definition: | Sets sharpening radius for capture. |
| Data Type: | Fixed Point Range |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read /Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum=0.1, Maximum=10, Units=0.1 pixels | 0.1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=0.1, Maximum=10, Units=0.1 pixels | 0.1 | EEPROM |
| Dimage Ex 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

isth – Image Sharpness Threshold

| | |
|---------------------|--|
| Definition: | Sets sharpening threshold for capture |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read /Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|-------------------------------------|-----------------|-----------|
| DigitaScript | Minimum=0, Maximum=25, Units=pixels | 4 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=0, Maximum=25, Units=pixels | 4 | EEPROM |
| Dimage Ex 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

mcap – Media Type Capture Mode

| | |
|----------------------|---|
| Definition: | Sets the system to capture the specified media type |
| Data Type: | Enum |
| Access: | Read /Write |
| Commands to Access: | GetCameraState, SetCameraState, GatFileTag |
| Value: | Persistent |
| Associated File Tag: | mcap |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 1=Still, 2=Burst, 3=Timelapse | 1 | RAM |
| Kodak DC 220 | 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 |
| Kodak DC 290 | 1=Still, 2=Burst, 3=Timelapse | 1 | RAM |
| Dimage EX 1500 | The media type capture mode, where 0=still, 1=burst and 2=timelapse | 0 | RAM |
| HP PhotoSmart C500 | 1=Still, 2=Burst, 3=Timelapse | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Still, 3=Timelapse | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Still, 3=Timelapse | 1 | EEPROM |
| Pentax EI-200 | 1=Still, 3=Timelapse | 1 | EEPROM |
| Pentax EI-2000 | 1=Still, 3=Timelapse | 1 | EEPROM |

mcro – Macro Mode

| | |
|---------------------|---|
| Definition: | Controls whether the optical system is in Macro mode (set up for short distance focussing distances). Applicable for a camera with an electronically controlled macro setting. If none exists, the enumeration list should have only one value (1=normal). When the camera has macro mode engaged (either under software control or manually), the focus distance should accurately reflect the macro distance (see the <code>fdst</code> parameter). |
| Data Type: | Enum |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Persistent |
| Access: | Read /Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | 1=Normal, 2=Macro Focus distance should accurately reflect the macro distance (see the 'fdst' parameter) | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Tag not supported | 1 | EEPROM |
| Dimage EX 1500 | 1=Normal (macro off), 2=Macro (macro on) | 1 | EEPROM |
| HP PhotoSmart C500 | Tag not supported | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Normal (0.5m - infinity), 2=Macro (0.1m -0.6m) | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Normal (0.5m - infinity), 2=Macro (0.02m -0.6m) | 1 | EEPROM |
| Pentax EI-200 | 1=Normal (0.5m - infinity), 2=Macro (0.1m -0.6m) | 1 | EEPROM |
| Pentax EI-2000 | 1=Normal (0.5m - infinity), 2=Macro (0.02m -0.6m) | 1 | EEPROM |

mhbs – Maximum Host Buffer Size

| | |
|---------------------|--|
| Definition: | Specifies the largest buffer required for transferring images on the host side. Host can set this value. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read /Write |

| Product/OS | Details | Factory Default | Stored In |
|--|---|-----------------|-----------|
| DigitaScript | Minimum=10240, Maximum=32768, Units=Bytes | 19456 | EEPROM |
| Kodak DC 220/ DC 260/ DC 265/DC 290 | Minimum=10240, Maximum=32768, Units=Bytes | 19456 | EEPROM |
| Dimage EX 1500 | Minimum=10240, Maximum=32768, Units=Bytes | 19456 | EEPROM |
| HP PhotoSmart C500 | Minimum=10240, Maximum=65535, Units=Bytes | 32767 | EEPROM |
| HP PhotoSmart C618 | Minimum=10240, Maximum=32768, Units=Bytes | 19456 | EEPROM |
| HP PhotoSmart C912 | Minimum=10240, Maximum=32768, Units=Bytes | 19456 | EEPROM |
| Pentax EI-200 | Minimum=10240, Maximum=32768, Units=Bytes | 19456 | EEPROM |
| Pentax EI-2000 | 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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Volatile |
| Access: | Read /Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|--------------------------------------|-----------------|-----------|
| DigitaScript | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| Dimage EX 1500 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| HP PhotoSmart C500 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| HP PhotoSmart C618 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| HP PhotoSmart C912 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| Pentax EI-200 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |
| Pentax EI-2000 | Minimum=1, Maximum=60, Units=seconds | 10 | EEPROM |

pasw – User Password

| | |
|---------------------|--|
| Definition: | A storage area in EEPROM that can be used for any scripting purpose. |
| Data Type: | String |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | Maximum length of 31 characters. Case sensitive | "" | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Maximum length of 31 characters. Case sensitive | "" | EEPROM |
| Dimage Ex 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Maximum length of 31 characters. Case sensitive | "" | EEPROM |
| HP PhotoSmart C618 | Maximum length of 31 characters. Case sensitive | "" | EEPROM |
| HP PhotoSmart C912 | Maximum length of 31 characters. Case sensitive | "" | EEPROM |
| Pentax EI-200 | Maximum length of 31 characters. Case sensitive | "" | EEPROM |
| Pentax EI-2000 | Maximum length of 31 characters. Case sensitive | "" | EEPROM |

pbbr – Burst Playback Interval

| | |
|---------------------|---|
| Definition: | Sets the playback rate for burst captured data sets |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=1000, Maximum=30000 Units= 0.01 seconds | 5000 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum=1 Maximum=10 Units=seconds | 3 | EEPROM |
| Kodak DC 290 | Minimum=1000, Maximum=30000 | 3000 | EEPROM |
| Dimage EX 1500 | Minimum=1 Maximum=10 Units=seconds | 3 | EEPROM |
| HP PhotoSmart C500 | Minimum=3000, Maximum=60000 Units= 0.01 seconds | 3000 | EEPROM |
| HP PhotoSmart C618 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| HP PhotoSmart C912 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| Pentax EI-200 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| Pentax EI-2000 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |

pbgr – Group Playback Interval

| | |
|----------------------|---|
| Definition: | Sets the playback rate for user grouped data sets |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Affect from Upgrade: | Defaulted |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=1000, Maximum=30000 Units= 0.01 seconds | 5000 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum=1, Maximum=10 Units=seconds | 3 | EEPROM |
| Kodak DC 290 | Minimum=1000, Maximum=30000 Units= 0.01 seconds | 3000 | EEPROM |
| Dimage EX 1500 | Minimum=1, Maximum=10 Units=seconds | 3 | EEPROM |
| HP PhotoSmart C500 | Minimum=3000, Maximum=6000 Units= 0.01 seconds | 3000 | EEPROM |
| HP PhotoSmart C618 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| HP PhotoSmart C912 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| Pentax EI-200 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| Pentax EI-2000 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |

pbtr – Timelapse Playback Interval

| | |
|---------------------|---|
| Definition: | Sets the playback rate for Timelapse captured data sets |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=1000, Maximum=30000 Units= 0.01 seconds | 5000 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum=1, Maximum=10 Units=seconds | 3 | EEPROM |
| Kodak DC 290 | Minimum=1000, Maximum=30000 Units= 0.01 seconds | 3000 | EEPROM |
| Dimage EX 1500 | Minimum=1, Maximum=10 Units=seconds | 3 | EEPROM |
| HP PhotoSmart C500 | Minimum=30000, Maximum=60000 Units= 0.01 seconds | 30000 | EEPROM |
| HP PhotoSmart C618 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| HP PhotoSmart C912 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| Pentax EI-200 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |
| Pentax EI-2000 | Minimum=1000, Maximum=10000 Units= 0.001 seconds (milliseconds) | 3000 | EEPROM |

pcpy – Print Copy Count

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

| Product/OS | Details | Factory Default | Stored In |
|--------------------|---------------------------------------|-----------------|-----------|
| DigitaScript | Minimum=1, Maximum=50 Units=Copies | 1 | |
| HP PhotoSmart C500 | Minimum=1, Maximum=50 Units=Copies | 1 | EEPROM |
| HP PhotoSmart C618 | Minimum=1, Maximum=999 Units=Copies | 1 | EEPROM |
| HP PhotoSmart C912 | Minimum= 1, Maximum= 999 Units=Copies | 1 | EEPROM |
| Pentax EI-200 | Minimum=1, Maximum=999 Units=Copies | 1 | EEPROM |
| Pentax EI-2000 | Minimum= 1, Maximum= 999 Units=Copies | 1 | EEPROM |

pgms – Program Shift

| | |
|----------------------|--|
| Definition: | Used to 'bias' the exposure toward larger or smaller apertures. Controls shutter speed/aperture trade-off. |
| Data Type: | UInt |
| Access: | Read/Write |
| Commands to Access | GetCameraState, SetCameraState, GetFileTag |
| Value | Persistent |
| Associated File Tag: | pgms |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|---|-----------------|-----------|
| DigitaScript | Minimum=-300, Maximum=300, Units=0.01EV | 0 | |
| Kodak DC 290 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

qbti – Quarter Burst Time Interval

| | |
|--------------------|--|
| Definition: | Specifies the time interval between captures for 1/4 size burst capture. Value sent to CSS via StartCapture command. |
| Data Type: | UInt |
| Commands to Access | GetCameraState, SetCameraState |
| Value | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=100, Maximum=1000, Units=0.01 seconds | 250 | EEPROM |
| Kodak DC 220 | Minimum=80, Maximum=10000 | 80 | EEPROM |
| Kodak DC 260/DC 265/ DC 290 | Minimum=360, Maximum=10000 Units=0.001 seconds (milliseconds) | 360 | EEPROM |
| Dimage EX 1500 | Minimum=30, Maximum=10000 | 266 | EEPROM |
| HP PhotoSmart C500 | Minimum=333, Maximum=1000 | 333 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

rgnc – Region Code

| | |
|--------------------|---|
| Definition: | Specifies the current language selection used in the product. |
| Data Type: | Enum |
| Commands to Access | GetCameraState, SetCameraState, GetFileTag |
| Value | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|---|-------------------|-----------|
| DigitaScript | 1=US English, 2=UK English, 3=French, 4=Italian, 5=Spanish, 6=German, 7=Swedish, 8=Japanese, 9=Dutch, 10=Chinese (simplified) | 1 | EEPROM |
| Kodak DC 220/DC260/DC 265/ | Settings 1 and 8 are supported | 1 or 8 | EEPROM |
| Kodak 290 | Settings 1, 3, 6, 8, and 10 supported | 1 or 8 | EEPROM |
| Dimage EX 1500 | Settings 1, 3, 6, and 8 supported | depends on region | EEPROM |
| HP PhotoSmart C500 | Settings 1, 3, 4, 5 and 6 are supported | 1 | EEPROM |
| HP PhotoSmart C618 | Settings 1, 3, 4, 5, 6, and 8 supported | 1 | EEPROM |
| HP PhotoSmart C918 | Settings 1, 3, 4, 5, 6, and 8 supported | 1 | EEPROM |
| Pentax EI-200 | Settings 1, 3, 4, 5, 6, and 8 supported | 1 | EEPROM |
| Pentax EI-2000 | Settings 1, 3, 4, 5, 6, and 8 supported | 8 | EEPROM |

rmod – Auto Rotate Image

| | |
|----------------------|--|
| Definition: | Enables or disables automatic image rotation (if available). This parameter is required if the product supports a rotation sensor. |
| Data Type: | Enum |
| Commands to Access | GetCameraState, SetCameraState, GetFileTag |
| Value | Persistent |
| Access: | Read/Write |
| Associated File Tag: | rmod |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|------------------------------|-----------------|-----------|
| DigitaScript | 0=Off, 1=On | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | 1=Auto and 2=Landscape | 1 | EEPROM |
| Kodak DC 290 | 0=Off (landscape), 1=On | 1 | EEPROM |
| Dimage EX 1500 | Always ON, Off not supported | 1 | EEPROM |
| HP PhotoSmart C500 | 0=Off, 1=On | 1 | EEPROM |
| HP PhotoSmart C618 | 0=Off (landscape), 1=On | 1 | EEPROM |
| HP PhotoSmart C912 | 0=Off (landscape), 1=On | 1 | EEPROM |
| Pentax EI-200 | 0=Off (landscape), 1=On | 1 | EEPROM |
| Pentax EI-2000 | 0=Off (landscape), 1=On | 1 | EEPROM |

sccl – Still Capture Color Mode

| | |
|-------------|---|
| Definition: | Controls the color mode used for still capture. |
| Data Type: | Enum List |
| Value | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|--|-----------------|-----------|
| DigitaScript | 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 1 supported | 1 | EEPROM |
| Dimage EX 1500 | Settings 1 and 2 are supported | 1 | EEPROM |
| HP PhotoSmart C500 | 1=Full, 2=B&W | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Full, 2=Gray, 3=Tone | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Full, 2=Gray, 3=Tone | 1 | EEPROM |
| Pentax EI-200 | 1=Full, 2=Gray, 3=Tone | 1 | EEPROM |
| Pentax EI-2000 | 1=Full, 2=Gray, 3=Tone | 1 | EEPROM |

scmp – Strobe Compensation

| | |
|---------------------|---|
| Definition: | Biases the amount of additional light contributed to the exposure by the automatic strobe. Typically this affects either quench level, aperture, or both. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=-200, Maximum=200, Units=0.01EV | 0 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=-200, Maximum=200, Units=0.01EV | 0 | EEPROM |
| HP PhotoSmart C500 | Minimum=-200, Maximum=200, value in 0.01 f/stops | 0 | EEPROM |
| HP PhotoSmart C618 | Minimum=-200, Maximum=200 | 0 | EEPROM |
| HP PhotoSmart C912 | Minimum=-200, Maximum=200 | 0 | EEPROM |
| Pentax EI-200 | Minimum=-200, Maximum=200 | 0 | EEPROM |
| Pentax EI-2000 | Minimum=-200, Maximum=200 | 0 | EEPROM |

scpn – Still Capture Compression

| | |
|----------------------|---|
| Definition: | Sets the compression level for still capture images. Higher compression means lower image quality. It is transferred as a file tag only for still capture images. Any compression engine which is utilized must support ALL the enumerated values. If some are not available, then the closest available case should be substituted (favoring better quality/less compression). |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | cmpn |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------|---|-----------------|-----------|
| DigitaScript | 1=Maximum, 2=High, 3=Normal, 4=Low, 5=Very Low, 6=Minimum, 7=Lossless | 3 | EEPROM |
| Kodak DC 220/DC 260DC 265 | Settings 2, 3, and 4 are supported | 4 | EEPROM |
| Kodak DC 290 | Settings 2, 3, 4, and 7 are supported | 4 | EEPROM |
| Dimage EX 1500 | Settings 2, 3, 4, and 5 are supported | 3 | EEPROM |
| HP PhotoSmart C500 | Settings 2, 3, and 4 are supported | 3 | EEPROM |
| HP PhotoSmart C618 | Settings 2, 3, and 4 are supported | 3 | EEPROM |
| HP PhotoSmart C912 | Settings 2, 3, and 4 are supported | 3 | EEPROM |
| Pentax EI-200 | Settings 2, 3, and 4 are supported | 3 | EEPROM |
| Pentax EI-2000 | Settings 2, 3, and 4 are supported | 3 | EEPROM |

shut – Shutter Speed

| | |
|----------------------|--|
| Definition: | Controls the shutter speed in the Shutter Priority and Programmed exposure modes (see the xmod parameter). Value is transferred to a file tag when used for Shutter Priority and Programmed modes. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | shut |

| Product/OS | Details | Factory Default | Stored In |
|-----------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum=1000, Maximum=1000000 Units=0.000001 seconds (500000=0.5 seconds) | 10000 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC | Minimum=500000, Maximum=16000000 Units=0.000001 seconds (500000=0.5 seconds) | 500000 | EEPROM |
| Kodak DC 290 | Minimum=2840, 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 |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Minimum=1000, Maximum=4000000 Units=0.000001 seconds (500000=0.5 seconds) | 10000 | EEPROM |
| HP PhotoSmart C912 | Minimum=1000, Maximum=4000000 Units=0.000001 seconds (500000=0.5 seconds) | 10000 | EEPROM |
| Pentax EI-200 | Minimum=1000, Maximum=4000000 Units=0.000001 seconds (500000=0.5 seconds) | 10000 | EEPROM |
| Pentax EI-2000 | Minimum=1000, Maximum=4000000 Units=0.000001 seconds (500000=0.5 seconds) | 10000 | EEPROM |

smod – Strobe Mode

| | |
|----------------------|--|
| Definition: | Sets the operating mode for the strobe. |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | smod |

| Product/OS | Details | Factory Default | Stored In |
|---------------------|---|-----------------|-----------|
| DigitaScript | <p>1=Auto -Built-in flash fires when necessary</p> <p>2=Fill -Built-in flash fires in all situations This can be useful when the subject is backlit but not in the center of the frame, which may fool the auto-exposure system.</p> <p>3=Off -Built-in flash will not fire. Expect longer exposure times</p> <p>4=Slave -Built-in flash is used to trigger flash units not attached to the camera</p> <p>5=Sync -triggers a PC connected strobe (built-in flash will not fire)</p> | 1 | EEPROM |
| Kodak DC 220 | <p>1=Off -Built-in flash will not fire. Expect longer exposure times</p> <p>2=Auto -Built-in flash fires when necessary</p> <p>3=Fill -Built-in flash fires in all situations This can be useful when the subject is backlit but not in the center of the frame, which may fool the auto-exposure system.</p> | 2 | EEPROM |
| Kodak DC 260/DC 265 | <p>1=Off -Built-in flash will not fire. Expect longer exposure times</p> <p>2=Auto -Built-in flash fires when necessary</p> <p>3=Fill -Built-in flash fires in all situations This can be useful when the subject is backlit but not in the center of the frame, which may fool the auto-exposure system.</p> <p>5=Sync -triggers a PC connected strobe (built-in flash will not fire)</p> | 2 | EEPROM |
| Kodak DC 290 | Settings 1, 2, 3, and 5 are supported | 2 | EEPROM |
| Dimage EX 1500 | Settings 1, 2, and 3 are supported. | 2 | EEPROM |
| HP PhotoSmart C500 | Settings 1, 2, and 3 are supported. | 1 | EEPROM |
| HP PhotoSmart C618 | Settings 1, 2, 3, and 4 are supported | 1 | EEPROM |
| HP PhotoSmart C912 | Settings 1, 2, 3, and 5 are supported | 1 | EEPROM |
| Pentax EI-200 | Settings 1, 2, 3, and 4 are supported | 1 | EEPROM |
| Pentax EI-2000 | Settings 1, 2, 3, and 5 are supported | 1 | 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 |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | spre |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum =0, Maximum=3 | 0 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | 0=Off, 1=On Strobe preflashes according to manufacturer specifications | 0 | EEPROM |
| Dimage EX 1500 | 0=Off, 1=On Strobe preflashes according to manufacturer specifications | 0 | EEPROM |
| HP PhotoSmart C500 | Minimum =0, Maximum=1 | 0 | EEPROM |
| HP PhotoSmart C618 | Minimum =0, Maximum=1 | 0 | EEPROM |
| HP PhotoSmart C912 | Minimum =0, Maximum=1 | 0 | EEPROM |
| Pentax EI-200 | Minimum =0, Maximum=1 | 0 | EEPROM |
| Pentax EI-2000 | Minimum =0, Maximum=1 | 0 | EEPROM |

sscs – Slideshow Category String

| | |
|---------------------|--|
| Definition: | Specifies the category used to filter displayed images for the sect parameter. |
| Data Type: | String |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Volatile |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|--|-----------------|-----------|
| DigitaScript | ssct parameter must be set to category | first category | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |

ssct – Slideshow Content

| | |
|---------------------|--|
| Definition: | Selects whether all images will be shown during the slide show, or if they will be filtered by category. If the Category value is set, the Slide Show Category String (sscs) parameter indicates which category is selected. |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Volatile |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|---|-----------------|-----------|
| DigitaScript | 1=Category, 2=All Category set in sscs parameter | 2 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

ssdr – Slide Show Frame Interval

| | |
|---------------------|--|
| Definition: | Sets the image display time for a slide shows without a timetrack. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|-----------------------------------|--------------------------------------|-----------------|-----------|
| DigitaScript | Minimum=1, Maximum=99, Units=seconds | 5 | EEPROM |
| Kodak DC 220/DC 260/DC 265/DC 290 | Minimum=1, Maximum=99, Units=seconds | 3 | EEPROM |
| Dimage EX 1500 | Minimum=1, Maximum=10 | 3 | EEPROM |
| HP PhotoSmart C500 | Minimum=30, Maximum=60 | 30 | EEPROM |
| HP PhotoSmart C618 | Minimum=1, Maximum=10 | 2 | EEPROM |
| HP PhotoSmart C912 | Minimum=1, Maximum=10 | 2 | EEPROM |
| Pentax EI-200 | Minimum=1, Maximum=10 | 2 | EEPROM |
| Pentax EI-2000 | Minimum=1, Maximum=10 | 2 | EEPROM |

ssis – Slide Show Image Sound

| | |
|---------------------|---|
| Definition: | Turns image sound annotations on and off. This parameter does not control soundtrack sound. |
| Data Type: | Enum |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|-----------------------------------|------------------------------|-----------------|-----------|
| DigitaScript | 0=OFF, 1=ON | 1 | EEPROM |
| Kodak DC 220/DC 260/DC 265/DC 290 | 0=OFF, 1=ON | 1 | EEPROM |
| Dimage EX 1500 | Always off. On not supported | 0 | EEPROM |
| HP PhotoSmart C500 | Sound is not supported | 0 | EEPROM |
| HP PhotoSmart C618 | 0=OFF, 1=ON | 1 | EEPROM |
| HP PhotoSmart C912 | 0=OFF, 1=ON | 1 | EEPROM |
| Pentax EI-200 | 0=OFF, 1=ON | 1 | EEPROM |
| Pentax EI-2000 | 0=OFF, 1=ON | 1 | EEPROM |

ssiz – Still Capture Size

| | |
|---------------------|---|
| Definition: | Specifies the image size for still capture. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------|--|-----------------|-----------|
| DigitaScript | 1=Large Size, 2=Medium Size, 3=Small Size, and 4=Custom Size | 1 | EEPROM |
| Kodak DC 220 | Settings 1 and 3 are supported | 1 | EEPROM |
| Kodak DC 260/DC 265 | Settings 1, 2, and 3 are supported | 1 | EEPROM |
| Kodak DC 290 | 1=High (1792x1200), 2=Medium (1440x960), 3=Standard (720x480), 4=Ultra (2240x1500) | 1 | EEPROM |
| Dimage EX 1500 | Settings 1 and 2 are supported | 1 | EEPROM |
| HP PhotoSmart C500 | 1=High (1600x1200), 2=Medium (1152x864), 3=Low (640x480) | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Full (1600 x 1200), 2=1/4 (800 x 600) | 1 | EEPROM |
| HP PhotoSmart C912 | 1= Full (1600 x 1200), 2=1/4 (800 x 600) | 1 | EEPROM |
| Pentax EI-200 | 1=Full (1600 x 1200), 2=1/4 (800 x 600) | 1 | EEPROM |
| Pentax EI-2000 | 1= Full (1600 x 1200), 2=1/4 (800 x 600) | 1 | EEPROM |

sslp – Slide Show Loop Indicator

| | |
|---------------------|--|
| Definition: | Specifies ON or Off for automatic looping for slide shows. If OFF, slide show ends at last slide show. If ON, slide show continues indefinitely, returning to the first slide after the last slide has been shown. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|-------------|-----------------|-----------|
| DigitaScript | 0=Off, 1=On | 0 | EEPROM |
| Kodak DC 220/DC 260/DC 265 | 0=Off, 1=On | 0 | EEPROM |
| Kodak DC 290 | 0=Off, 1=On | 0 | EEPROM |
| Dimage EX 1500 | 0=Off, 1=On | 0 | EEPROM |
| HP PhotoSmart C500 | 0=Off, 1=On | 1 | EEPROM |
| HP PhotoSmart C618 | 0=Off, 1=On | 1 | EEPROM |
| HP PhotoSmart C912 | 0=Off, 1=On | 1 | EEPROM |
| Pentax EI-200 | 0=Off, 1=On | 1 | EEPROM |
| Pentax EI-2000 | 0=Off, 1=On | 1 | 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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | <p>Minimum=1, Maximum=10</p> <p>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. This mechanism allows the user to create sets of sounds, including use of existing sounds, and download to camera. This parameter selects which set to use.</p> | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Tag not supported | | |
| Dimage EX 1500 | 0=ON | 0 | EEPROM |
| HP PhotoSmart C500 | Minimum=1, Maximum=1 | 1 | ROM |
| HP PhotoSmart C618 | Minimum=1, Maximum=1 | 1 | ROM |
| HP PhotoSmart C912 | Minimum=1, Maximum=1 | 1 | ROM |
| Pentax EI-200 | Minimum=1, Maximum=1 | 1 | ROM |
| Pentax EI-2000 | Minimum=1, Maximum=1 | 1 | ROM |

ssvl – System Sound Volume

| | |
|--------------------|--|
| Definition: | Specifies the volume of the system sounds including shutter sound, if available. If set to zero, sound is off. If set to maximum value, sound is loudest. If there is only on and off then use min/max of 0,1. |
| Data Type: | UInt |
| Commands to Access | GetCameraState, SetCameraState |
| Value | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|----------------------------|-----------------|-----------|
| DigitaScript | 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 |
| HP PhotoSmart C500 | Minimum=0, Maximum=1 | 1 | EEPROM |
| HP PhotoSmart C618 | Minimum=0, Maximum=1 | 1 | EEPROM |
| HP PhotoSmart C912 | Minimum=0, Maximum=1 | 1 | EEPROM |
| Pentax EI-200/ | Minimum=0, Maximum=1 | 1 | EEPROM |
| Pentax EI-2000/ | Minimum=0, Maximum=1 | 1 | EEPROM |

stim – Self Timer Enable

| | |
|---------------------|--|
| Definition: | Enables the self timer mode. When enable, the timer delays the capture processing associated with S2. The amount of delay is set by the <code>mtdy</code> parameter. |
| Data Type: | Enum List |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Volatile |
| Access: | Read/Write |

| Product/OS | Definition | Factory Default | Stored In |
|--------------------|--|-----------------|-----------|
| DigitaScript | 0=OFF, 1=ON The amount of delay is set by the <code>mtdy</code> parameter | 0 | RAM |
| Kodak DC 290 | 0=OFF, 1=ON | 0 | RAM |
| HP PhotoSmart C500 | 0=OFF, 1=ON | 0 | EEPROM |
| HP PhotoSmart C618 | 0=OFF, 1=ON | 0 | EEPROM |
| HP PhotoSmart C912 | 0=OFF, 1=ON | 0 | EEPROM |
| Pentax EI-200/ | 0=OFF, 1=ON | 0 | EEPROM |
| Pentax EI-2000 | 0=OFF, 1=ON | 0 | EEPROM |

tccl – Timelapse Capture Color Mode

| | |
|---------------------|--|
| Definition: | Controls the color mode used for timelapse capture. Required if timelapse capture is supported. This parameter is transferred to file tags for timelapse capture only. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|--|-----------------|-----------|
| DigitaScript | 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 | Setting 1 is supported | 1 | EEPROM |
| Dimage EX 1500 | Settings 1 and 2 supported | 1 | EEPROM |
| HP PhotoSmart C500 | Settings 1 and 2 are supported | 1 | EEPROM |
| HP PhotoSmart C618 | 1=full, 2=Gray scale (B&W), 3=Tone | 1 | EEPROM |
| HP PhotoSmart C912 | 1=full, 2=Gray scale (B&W), 3=Tone | 1 | EEPROM |
| Pentax EI-200 | 1=full, 2=Gray scale (B&W), 3=Tone | 1 | EEPROM |
| Pentax EI-2000 | 1=full, 2=Gray scale (B&W), 3=Tone | 1 | EEPROM |

tcnt – Timelapse Capture Count

| | |
|---------------------|--|
| Definition: | Sets the maximum number of images to capture for a timelapse. Note that fewer images will be captured if insufficient memory is available. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=-2, Maximum=1000, Units=Images | 10 | EEPROM |
| Kodak DC 220/DC 260DC 265/DC 290 | Minimum=-2, Maximum=1000, Units=Images | 10 | EEPROM |
| Dimage EZ1500 | Minimum=-2, Maximum=200 | 10 | EEPROM |
| HP PhotoSmart C500 | Minimum=-2, Maximum=1000, Units=Images | 10 | EEPROM |
| HP PhotoSmart C618 | Minimum=-2, Maximum=1000, Units=Images | 2 | EEPROM |
| HP PhotoSmart C912 | Minimum=-2, Maximum=1000, Units=Images | 2 | EEPROM |
| Pentax EI-200 | Minimum=-2, Maximum=1000, Units=Images | 2 | EEPROM |
| Pentax EI-2000 | Minimum=-2, Maximum=1000, Units=Images | 2 | EEPROM |

tcpn – Timelapse Capture Compression

| | |
|----------------------|---|
| Definition: | Sets the compression level for timelapse capture images. Required if timelapse capture is supported. Tags only timelapse images. Any compression engine that is utilized must support ALL of the enumerated values. If some are not supported, then the closest available case should be substituted. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | cmpn |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------|---|-----------------|-----------|
| DigitaScript | 1=Maximum, 2=High, 3=Normal, 4=Low, 5=Very Low, 6=Minimum, 7=Lossless | 3 | EEPROM |
| Kodak DC 220/DC 260DC 265 | Settings 1,5, 6 and 7 not supported | 4 | EEPROM |
| Kodak DC 290 | Settings 1, 5 and 6 not supported | 4 | EEPROM |
| Dimage EX 1500 | Settings 1, 6 and 7 not supported | 3 | EEPROM |
| HP PhotoSmart C500 | 2=High, 3=Normal, 4=Low | 3 | EEPROM |
| HP PhotoSmart C618 | 2=High, 3=Normal, 4=Low, | 3 | EEPROM |
| HP PhotoSmart C912 | 2=High, 3=Normal, 4=Low, | 3 | EEPROM |
| Pentax EI-200 | 2=High, 3=Normal, 4=Low, | 3 | EEPROM |
| Pentax EI-2000 | 2=High, 3=Normal, 4=Low, | 3 | EEPROM |

tfmt – Time Format

| | |
|---------------------|---|
| Definition: | Time format used for entering and editing time values. Used for watermarks and some display screens. A display screen may be without seconds, at its option, to conserve space. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|-----------------------------------|--|-----------------|-----------|
| DigitaScript | 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 |
| HP PhotoSmart C500 | 1=hh:mm am/pm, 2=hh:mm:ss am/pm, 3=hh:mm, 4=hh:mm:ss | 1 | EEPROM |
| HP PhotoSmart C618 | 1=hh:mm am/pm, 2=hh:mm:ss am/pm, 3=hh:mm (24 hrs), 4=hh:mm:ss (24 hrs) | 1 | EEPROM |
| HP PhotoSmart C912 | 1=hh:mm am/pm, 2=hh:mm:ss am/pm, 3=hh:mm (24 hrs), 4=hh:mm:ss (24 hrs) | 1 | EEPROM |
| Pentax EI-200 | 1=hh:mm am/pm, 2=hh:mm:ss am/pm, 3=hh:mm (24 hrs), 4=hh:mm:ss (24 hrs) | 3 | EEPROM |
| Pentax EI-2000 | 1=hh:mm am/pm, 2=hh:mm:ss am/pm, 3=hh:mm (24 hrs), 4=hh:mm:ss (24 hrs) | 3 | EEPROM |

tldy – Timelapse Capture Interval

| | |
|---------------------|--|
| Definition: | Specifies the time interval between captures. Required if timelapse capture is supported. Value is sent to CSS via StartCapture command. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum=1500, Maximum=8640000, Units=0.01seconds | 6000 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum=60, Maximum=86400, Units=seconds | 60 | EEPROM |
| Kodak DC 290 | Minimum=6000, Maximum=8640000, Units=0.01seconds | 6000 | EEPROM |
| Dimage EX 1500 | Minimum=60, Maximum=86400, Units=seconds | 60 | EEPROM |
| HP PhotoSmart C500 | Minimum=1500, Maximum=8640000, Units=0.01second steps. (8640000 x 0.01=24 hours) | 6000 | EEPROM |
| HP PhotoSmart C618 | Minimum=1500, Maximum=8640000, Units=0.01seconds | 1500 | EEPROM |
| HP PhotoSmart C912 | Minimum=1500, Maximum=8640000, Units=0.01seconds | 1500 | EEPROM |
| Pentax EI-200 | Minimum=1500, Maximum=8640000, Units=0.01seconds | 1500 | EEPROM |
| Pentax EI-2000 | Minimum=1500, Maximum=8640000, Units=0.01seconds | 1500 | EEPROM |

tsiz – Timelapse Capture Size

| | |
|---------------------|--|
| Definition: | Sets the image size for timelapse Captures. Required if timelapse capture is supported. Value is sent to CCS via the StartCapture command. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product | Details | Factory Default | Stored In |
|---------------------|--|-----------------|-----------|
| DigitaScript | 1=Large Size, 2=Medium Size, 3=Small Size, and 4=Custom Size | 1 | EEPROM |
| Kodak DC 220 | Settings 1 and 3 are supported | 1 | EEPROM |
| Kodak DC 260/DC 265 | Settings 1, 2, and 3 are supported | 1 | EEPROM |
| Kodak DC 290 | 1=High (1792x1200), 2=Medium (1440x960), 3=Standard (720x480), 4=Ultra (2240x1500) | 1 | EEPROM |
| Dimage EX 1500 | Settings 1 and 2 are supported | 1 | EEPROM |
| HP PhotoSmart C500 | 1=High (1600x1200), 2=Medium (1152x864), 3=Low (640x480) | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Full Size (1600 x 1200), 2=1/4 Size (800 x 600) | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Full Size (1600 x 1200), 2=1/4 Size (800 x 600) | 1 | EEPROM |
| Pentax EI-200 | 1=Full Size (1600 x 1200), 2=1/4 Size (800 x 600) | 1 | EEPROM |
| Pentax EI-2000 | 1=Full Size (1600 x 1200), 2=1/4 Size (800 x 600) | 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 |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag, SetUserFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | ucpt |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|-------------------------------------|-----------------|-----------|
| DigitaScript | | "" | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported. Use crcp instead | | |
| Kodak DC 290 | | "" | EEPROM |
| Dimage EX 1500 | Tag not supported. Use crcp instead | | |
| HP PhotoSmart C500 | | "" | EEPROM |
| HP PhotoSmart C618 | | "" | EEPROM |
| HP PhotoSmart C912 | | "" | EEPROM |
| Pentax EI-200 | | "" | EEPROM |
| Pentax EI-2000 | | "" | EEPROM |

vmod – Video Output Mode

| | |
|---------------------|--------------------------------|
| Definition: | Sets the video output mode. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|------------------------|-----------------|-----------|
| DigitaScript | 1=NTSC, 2=PAL, 3=SECAM | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | 0=NTSC, 1=PAL | 0 | EEPROM |
| Kodak DC 290 | 1=NTSC, 2=PAL | 1 | EEPROM |
| Dimage EX 1500 | 0=NTSC, 1=PAL | 0 | EEPROM |
| HP PhotoSmart C500 | 1=NTSC, 2=PAL | 1 | EEPROM |
| HP PhotoSmart C618 | 1=NTSC, 2=PAL | 1 | EEPROM |
| HP PhotoSmart C912 | 1=NTSC, 2=PAL | 1 | EEPROM |
| Pentax EI-200 | 1=NTSC, 2=PAL | 1 | EEPROM |
| Pentax EI-2000 | 1=NTSC, 2=PAL | 1 | EEPROM |

wbcb – White Balance Blue

| | |
|-----------------------|---|
| Definition: | Controls the value for the blue component when the <code>wmod</code> parameter is set to Custom. The value transferred to the file tag always reflects the actual value used to process the image. If the white balance mode is set to “Custom”, then the value of this parameter will be used for processing the image and transferred to the file tag. For all other white balance modes, the CCS and image processing (IPU) will determine the white balance, and transfer the calculated value to the file tag. |
| Data Type: | Fixed Point Range |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> , <code>GetFileTag</code> |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tags: | wbcb |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|------------------|-----------|
| DigitaScript | Minimum=1.0, Maximum=3.99 | 1.0 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum =0.25 (expressed as 0x00004000) Maximum=4.00 (expressed as 0x00040000) | 1.0 (0x00010000) | EEPROM |
| Kodak DC 290 | Tag not supported | | |
| Dimage EX 1500 | Minimum=-0x4000, Maximum=0x40000 | 0x10000 | EEPROM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

wbcg – White Balance Green

| | |
|-----------------------|---|
| Definition: | Controls the value for the green component when the wmod parameter is set to Custom. The value transferred to the file tag always reflects the actual value used to process the image. If the white balance mode is set to “Custom”, then the value of this parameter will be used for processing the image and transferred to the file tag. For all other white balance modes, the CCS and image processing (IPU) will determine the white balance, and transfer the calculated value to the file tag. |
| Data Type: | Fixed Point Range |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tags: | wbcg |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|------------------|-----------|
| DigitaScript | Minimum=1.0, Maximum=3.99 | 1.0 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum =0.25 (expressed as 0x00004000) Maximum=4.00 (expressed as 0x00040000) | 1.0 (0x00010000) | EEPROM |
| Kodak DC 290 | Tag not supported | | |
| Dimage EX 1500 | Minimum=-0x4000, Maximum=0x40000 | 0x10000 | EEPROM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

wbcr – White Balance Red

| | |
|-----------------------|---|
| Definition: | Controls the value for the red component when the wmod parameter is set to Custom. The value transferred to the file tag always reflects the actual value used to process the image. If the white balance mode is set to “Custom”, then the value of this parameter will be used for processing the image and transferred to the file tag. For all other white balance modes, the CCS and image processing (IPU) will determine the white balance, and transfer the calculated value to the file tag. |
| Data Type: | Fixed Point Range |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tags: | wbcr |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|------------------|-----------|
| DigitaScript | Minimum=1.0, Maximum=3.99 | 1.0 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Minimum =0.25 (expressed as 0x00004000) Maximum=4.00 (expressed as 0x00040000) | 1.0 (0x00010000) | EEPROM |
| Kodak DC 290 | Tag not supported | | |
| Dimage EX 1500 | Minimum=-0x4000, Maximum=0x40000 | 0x10000 | EEPROM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

wena – Watermark Enable

| | |
|---------------------|---|
| Definition: | Sets which kind of watermark will be applied to an image during image capture processing. |
| Data Type: | BitFlags |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product OS | Details | Factory Default | Stored In |
|----------------------------|--|-----------------|-----------|
| DigitaScript | Minimum=0b0000, Maximum=0b1111 Bit functions are: Logo enable (0b1000), Text enable (0b0100), Time enable (0b0010), Date enable (0b0001) | 0b0000 | EEPROM |
| Kodak DC 220/DC 260/DC 265 | 0000 = disables all watermarks, 0001 = enables date only, 0010 = enables time only, 0100 = enables text only, 1000 = enables loog only, 1111 = enables time/date/text/logo, 0011 = enables time and date | 0000 | EEPROM |
| Kodak DC 290 | Minimum=0b0000, Maximum=0b1111 Bit functions are: Logo enable (0b1000), Text enable (0b0100), Time enable (0b0010), Date enable (0b0001) | 0b0000 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Minimum=0x0, Maximum=0xf | 0x0 | EEPROM |
| HP PhotoSmart C912 | Minimum=0x0, Maximum=0xf | 0x0 | EEPROM |
| Pentax EI-200 | Minimum=0x0, Maximum=0xf | 0x0 | EEPROM |
| Pentax EI-2000 | Minimum=0x0, Maximum=0xf | 0x0 | EEPROM |

wmod – White Balance Mode

| | |
|----------------------|--|
| Definition: | Sets the mode for white balance during image capture. In “Auto” mode the capture system will attempt to determine the correct values to use from the image that was captured. In “Off” mode, the three co-efficient used are 1.0/1.0/1.0, which results in no white balance adjustment being made to the image. In “Custom” mode, the values of parameters wbc, wbcg, and wbcg are used. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | wmod |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------|---|-----------------|-----------|
| DigitaScript | 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 | 1 | EEPROM |
| Kodak DC 220/DC 260/DC265 | Settings 1,3,6, 9 and 11 are supported (11= Custom) | 1 | EEPROM |
| Kodak DC 290 | Settings 1,3,6, 9 and 11 are supported (11= Off) | 1 | EEPROM |
| Dimage EX 1500 | Settings 1,3,6, 9 and 11are supported | 1 | EEPROM |
| HP PhotoSmart C500 | Settings 1, 3, 6, 9, and 10 are supported | 1 | EEPROM |
| HP PhotoSmart C618 | Settings 1, 3, 6, and 9 are supported | 1 | EEPROM |
| HP PhotoSmart C912 | Settings 1, 3, 6, and 9 are supported | 1 | EEPROM |
| Pentax EI-200 | Settings 1, 3, 6, and 9 are supported | 1 | EEPROM |
| Pentax EI-2000 | Settings 1, 3, 6, and 9 are supported | 1 | EEPROM |

wpic – Watermark Logo File Name

| | |
|---------------------|---|
| Definition: | The DOS filename of the watermark logo. |
| Data Type: | DosName |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|--|-----------------|-----------|
| DigitaScript | Requires wena parameter to be set to enable the logo | "" | EEPROM |
| Kodak DC 220 | Tag not supported | | |
| Kodak DC 260, DC265, DC290 | Requires wena parameter to be set to enable the logo | "" | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Requires wena parameter to be set to enable the logo | "" | EEPROM |
| HP PhotoSmart C912 | Requires wena parameter to be set to enable the logo | "" | EEPROM |
| Pentax EI-200 | Requires wena parameter to be set to enable the logo | "" | EEPROM |
| Pentax EI-2000 | Requires wena parameter to be set to enable the logo | "" | EEPROM |

wpop – Watermark Logo Operation Mode

| | |
|---------------------|---|
| Definition: | Sets the graphics mode for merging of the logo with the image. This may be set to either source copy (srcCopy) or source (srcOr). |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|-------------------------------|--|-----------------|-----------|
| DigitaScript | 0=SourceCopy (Opaque Logo), 1=SourceOR (Translucent Logo) | 0 | EEPROM |
| Kodak DC 220 | Tag not supported | | |
| Kodak DC 260, DC265, DC290 | 1= SourceOR, sets areas of the logo masked in alpha channel to transparent/translucent | 1 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | 0=SourceCopy (Opaque Logo), 1=SourceOR (Translucent Logo) | 0 | EEPROM |
| HP PhotoSmart C912 | 0=SourceCopy (Opaque Logo), 1=SourceOR (Translucent Logo) | 0 | EEPROM |
| Pentax EI-200 | 0=SourceCopy (Opaque Logo), 1=SourceOR (Translucent Logo) | 0 | EEPROM |
| Pentax EI-2000 | 0=SourceCopy (Opaque Logo), 1=SourceOR (Translucent Logo) | 0 | EEPROM |

wpxp – Watermark Logo X Position

| | |
|---------------------|--|
| Definition: | Sets the horizontal position and justification for the logo. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 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 | | |
| Kodak DC 260/DC 265/ DC 290 | Minimum =0, Maximum=100, in steps of 2 | 4 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Minimum=0, Maximum=100, Units=percent | 4 | EEPROM |
| HP PhotoSmart C912 | Minimum=0, Maximum=100, Units=percent | 4 | EEPROM |
| Pentax EI-200 | Minimum=0, Maximum=100, Units=percent | 4 | EEPROM |
| Pentax EI-2000 | Minimum=0, Maximum=100, Units=percent | 4 | EEPROM |

wpyp – Watermark Logo Y Position

| | |
|---------------------|--|
| Definition: | Sets the vertical position and justification for the logo. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 260/DC 265/ DC 290 | Minimum =0, Maximum=100, in steps of 2 | 4 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 96 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 96 | EEPROM |

wsbc – Watermark Text Background Color

| | |
|---------------------|---|
| Definition: | Sets the background color for watermark text. |
| Data Type: | Uhex |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 290 | Same as DigitaScript | 0x00800080 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 0x00800080 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 0x00800080 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 0x00800080 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 0x00800080 | EEPROM |

wsfc – Watermark Text Foreground Color

| | |
|---------------------|---|
| Definition: | Sets the text color for watermark text. |
| Data Type: | Uhex |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 0xFF80FF80 | 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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|------------------------------|--|-----------------|-----------|
| DigitaScript | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Kodak DC 220/ DC 260/ DC 265 | Setting 1=Opaque Setting 2 makes the background color transparent | 0 | EEPROM |
| Kodak DC 290 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Pentax EI-200 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Pentax EI-2000 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |

wstr– Watermark Text String

| | |
|----------------------|--|
| Definition: | Sets the user watermark text. |
| Data Type: | String |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | wstr |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | The watermark text string. The null string is "". | "" | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| Kodak DC 220/DC 260/ DC 265/DC 290 | The watermark text string. The null string is "". | "" | EEPROM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | The watermark text string. The null string is "". | "" | EEPROM |
| HP PhotoSmart C912 | The watermark text string. The null string is "". | "" | EEPROM |
| Pentax EI-200 | The watermark text string. The null string is "". | "" | EEPROM |
| Pentax EI-2000 | The watermark text string. The null string is "". | "" | 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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|---|-----------------|-----------|
| DigitaScript | 16=16 point, 18=18 point, 20=20 point, 24=24 point, 36=36 point | 36 | EEPROM |
| Kodak DC 220/DC 260/DC 265 | Tag not supported | | |
| Kodak DC 290 | 16=16 point, 18=18 point, 24=24 point, 36=36 point | 36 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |
| HP PhotoSmart C912 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |
| Pentax EI-200 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |
| Pentax EI-2000 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |

wsxp – Watermark Text X Position

| | |
|---------------------|--|
| Definition: | Sets the horizontal position and justification for the text. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Same as DigitaScript | 4 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 96 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 96 | EEPROM |

wsyp – Watermark Text Y Position

| | |
|---------------------|--|
| Definition: | Sets the horizontal position and justification for the text. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | 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/DC 260/ DC 265/DC 290 | Same as DigitaScript | 96 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 96 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 96 | EEPROM |

wtbc – Watermark Time Background Color

| | |
|---------------------|--|
| Definition: | Sets the background color for watermark date/time. |
| Data Type: | Uhex |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|--|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 290 | Same as DigitaScript | 0x00800080 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 0x800080 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 0x800080 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 0x800080 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 0x800080 | EEPROM |

wafc – Watermark Time Foreground Color

| | |
|---------------------|--|
| Definition: | Sets the foreground color for watermark date/time. |
| Data Type: | Uhex |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 290 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 0xFF80FF80 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 0xFF80FF80 | EEPROM |

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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------|--|-----------------|-----------|
| DigitaScript | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Kodak DC 220/ DC 260/ DC 265 | Settings 1 and 2 are supported | 0 | EEPROM |
| Kodak DC 290 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Dimage EX 1500 | Tag not supported | | EEPROM |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| HP PhotoSmart C912 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Pentax EI-200 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |
| Pentax EI-2000 | 1=Opaque, 2=Translucent, 3=Transparent | 1 | EEPROM |

wttS – 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 |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|----------------------------|--|-----------------|-----------|
| DigitaScript | 16=16 point, 18=18 point, 20=20 points, 24=24 point, 36=36 point | 36 | EEPROM |
| Kodak DC 220/DC 260/DC 265 | Tag not supported | | |
| Kodak DC 290 | 16=16 point, 18=18 point, 24=24 point, 36=36 point | 36 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |
| HP PhotoSmart C912 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |
| Pentax EI-200 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |
| Pentax EI-2000 | 16=16 point, 18=18 point, 20=point, 24=24 point, 36=36 point | 36 | EEPROM |

wtxp – Watermark Time X Position

| | |
|---------------------|---|
| Definition: | Sets the horizontal position and justification for the watermark date/time. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Same as DigitaScript | 96 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 96 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 96 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 96 | EEPROM |

wtyp – Watermark Time Y Position

| | |
|---------------------|---|
| Definition: | Sets the vertical position and justification for the watermark date/time. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|---|-----------------|-----------|
| DigitaScript | 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 |
| Kodak DC 220/DC 260/ DC 265/DC 290 | Same as DigitaScript | 96 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Tag not supported | | |
| HP PhotoSmart C618 | Same as DigitaScript | 4 | EEPROM |
| HP PhotoSmart C912 | Same as DigitaScript | 4 | EEPROM |
| Pentax EI-200 | Same as DigitaScript | 4 | EEPROM |
| Pentax EI-2000 | Same as DigitaScript | 4 | EEPROM |

xcmp – Exposure Compensation

| | |
|----------------------|--|
| Definition: | Sets the amount of forced change to the automatic exposure (AE) either by decreasing (negative number) or increasing (positive number) the exposure. |
| Data Type: | UInt |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | xcmp |

| Product/OS | Details | Factory Default | Stored In |
|---------------------------------------|--|-----------------|-----------|
| DigitaScript | 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 |
| HP PhotoSmart C500 | Minimum= -150, Maximum= 150 | 0 | EEPROM |
| HP PhotoSmart C618 | Minimum= -300, Maximum= 300. Units= 0.01 EV | 0 | |
| HP PhotoSmart C912 | Minimum= -300, Maximum= 300. Units= 0.01 EV | 0 | |
| Pentax EI-200 | Minimum= -300, Maximum= 300. Units= 0.01 EV | 0 | |
| Pentax EI-2000 | Minimum= -300, Maximum= 300. Units= 0.01 EV | 0 | |

xmod – Exposure Mode

| | |
|---------------------|--|
| Definition: | Sets the exposure mode for the camera. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState, GetFileTag |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag | xmod |

| Product/OS | Details | Factory Default | Stored In |
|---------------------|--|-----------------|-----------|
| DigitaScript | 1=Auto, 2=Shutter Priority, 3=Aperture Priority, 4=Gain Priority, 5=Programmed, 6=Manual | | EEPROM |
| Kodak DC 220 | 1=programmed, 2 =long exposure mode | 1 | EEPROM |
| Kodak DC 260/DC 265 | 1=programmed, 2 =long exposure mode, 3=external flash sync mode | 1 | EEPROM |
| Kodak DC 290 | 1=programmed, 2 =long exposure mode, 3=external flash sync mode | 1 | EEPROM |
| Dimage EX 1500 | Settings 1, 3, 4, are supported | 1 | EEPROM |
| HP PhotoSmart C500 | 1=programmed | 1 | EEPROM |
| HP PhotoSmart C618 | 1=Auto, 2=Shutter Priority, 3=Aperture Priority | 1 | EEPROM |
| HP PhotoSmart C912 | Settings 1, 2, 3, 5 and 6 are supported | 1 | EEPROM |
| Pentax EI-200 | 1=Auto, 2=Shutter Priority, 3=Aperture Priority | 1 | EEPROM |
| Pentax EI-2000 | Settings 1, 2, 3, 5 and 6 are supported | 1 | EEPROM |

xmtd– Exposure Method

| | |
|----------------------|---|
| Definition: | Specifies the metering pattern used to determine the exposure. Selection is not applicable for Programmed or Manual exposure mode settings made by <code>xmod</code> parameter. |
| Data Type: | Enum List |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> , <code>GetFileTag</code> |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | <code>xmtd</code> |

| Product/OS | Details | Factory Default | Stored In |
|-------------------------------------|--|-----------------|-----------|
| DigitaScript | 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 | 3 | EEPROM |
| Dimage EX 1500 | Setting 1 only is supported | 1 | EEPROM |
| HP PhotoSmart C500 | Settings 1 and 4 are supported | 1 | EEPROM |
| HP PhotoSmart C618 | 3=Center Weighted, 4=Center Spot, 5=Average | 5 | EEPROM |
| HP PhotoSmart C912 | 3=Center Weighted, 4=Center Spot, 5=Average | 5 | EEPROM |
| Pentax EI-200 | 3=Center Weighted, 4=Center Spot, 5=Average | 5 | EEPROM |
| Pentax EI-2000 | 3=Center Weighted, 4=Center Spot, 5=Average | 5 | EEPROM |

zmod– Zoom Mode

| | |
|----------------------|--|
| Definition: | Specifies the operational mode of the zoom function. Motorized mode is used when the user operates the zoom with zoom buttons. Programmed mode sets zoom control to <code>zpos</code> ; manual disengages any motors and allows user direct operation of the lens. |
| Data Type: | Enum List |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> , <code>GetFileTag</code> |
| Value: | Persistent |
| Access: | Read/Write |
| Associated File Tag: | <code>zmod</code> |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|----------------------------------|-----------------|-----------|
| DigitaScript | 1=Motorized, 2=Program, 3=Manual | 1 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Supports setting 2 only | 2 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Supports setting 2 only | 2 | EEPROM |
| HP PhotoSmart C618 | Supports setting 2 only | 2 | EEPROM |
| HP PhotoSmart C912 | Supports setting 2 only | 2 | EEPROM |
| Pentax EI-200 | Supports setting 2 only | 2 | EEPROM |
| Pentax EI-2000 | Supports setting 2 only | 2 | EEPROM |

zpos– Zoom Position

| | |
|----------------------|--|
| Definition: | Controls the position of the zoom lens when the <code>zmod</code> parameter is set to Programmed (2). Value is in 100x units, where 100 is the wide angle (1x) setting. A 3x zoom will have the range 100 to 300. If a camera does not have zoom functionality, the range should be set to a minimum and maximum of 100. |
| Data Type: | Enum List |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> , <code>GetFileTag</code> |
| Value: | Volatile |
| Access: | Read/Write |
| Associated File Tag: | <code>zpos</code> |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum=100, Maximum=300, Units=0.01 steps | 100 | RAM |
| Kodak DC 220 | Minimum=100, Maximum=200, Units=0.01 steps | 100 | RAM |
| Kodak DC 260/DC 265/ DC 290 | Minimum=100, Maximum=300, Units=0.01 steps | 130 | RAM |
| HP PhotoSmart C500 | Minimum=100, Maximum=300, Units=0.01 steps | 100 | RAM |
| HP PhotoSmart C618 | Minimum=100, Maximum=300 units = 0.01 steps | 100 | RAM |
| HP PhotoSmart C912 | Minimum=100, Maximum=300 units = 0.01 steps | 100 | RAM |
| Pentax EI-200 | Minimum=100, Maximum=300 units = 0.01 steps | 100 | RAM |
| Pentax EI-2000 | Minimum=100, Maximum=300 units = 0.01 steps | 100 | RAM |

zspd– Zoom Speed

| | |
|----------------------|---|
| Definition: | Sets the speed of the optical assembly adjusting its focal length or “zooming”. A value of 100 represents normal zoom speed, 50 represents half normal speed, and 200 represents twice normal speed. If the camera only supports one speed for zooming, the minimum and maximum should be set to 100. |
| Data Type: | Enum List |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Volatile |
| Access: | Read/Write |
| Associated File Tag: | zpos |

| Product/OS | Details | Factory Default | Stored In |
|--------------------------------|---|-----------------|-----------|
| DigitaScript | Minimum=50, Maximum=200, Units=percent 100 = normal zoom speed, 50 = half normal zoom speed, 200 = twice normal zoom speed | 100 | EEPROM |
| Kodak DC 220/DC 260/ DC 265 | Tag not supported | | |
| Kodak DC 290 | Minimum=100, Maximum=100 | 100 | EEPROM |
| Dimage EX 1500 | Tag not supported | | |
| HP PhotoSmart C500 | Minimum=50, Maximum=200, Units=percent | 100 | EEPROM |
| HP PhotoSmart C618 | Tag not supported | | |
| HP PhotoSmart C912 | Tag not supported | | |
| Pentax EI-200 | Tag not supported | | |
| Pentax EI-2000 | Tag not supported | | |

Custom Parameters

Some manufacturers have chosen to add extra parameters to allow script access to some camera-specific features. They are distinguished from the standard Digita parameters by the fact that they are written in UPPER CASE.

AUDV– Audio Volume Control

Definition: Controls the audio volume
Data Type: UInt
Commands to Access: GetCameraState, SetCameraState
Value: Persistent
Access: Read/Write

| Product/OS | Details | Factory Default | Stored In |
|--------------------|----------------------|-----------------|-----------|
| HP PhotoSmart C618 | Minimum=1, Maximum=7 | 4 | EEPROM |
| HP PhotoSmart C912 | Minimum=1, Maximum=7 | 4 | EEPROM |
| Pentax EI-200 | Minimum=1, Maximum=7 | 4 | EEPROM |
| Pentax EI-2000 | Minimum=1, Maximum=7 | 4 | EEPROM |

BOND– Born On Date

Definition: Date (in seconds since 1/1/1970) that the camera was first powered on, after shipping.
Data Type: UInt
Commands to Access: GetProductInfo
Value: Persistent
Access: Read Only

| Product/OS | Details | Factory Default | Stored In |
|--------------------|--|-----------------|-----------|
| HP PhotoSmart C618 | Date (in seconds since 1/1/1970) that the camera was first powered on, after shipping. | | EEPROM |
| HP PhotoSmart C912 | Date (in seconds since 1/1/1970) that the camera was first powered on, after shipping. | | EEPROM |
| Pentax EI-200 | Date (in seconds since 1/1/1970) that the camera was first powered on, after shipping. | | EEPROM |
| Pentax EI-2000 | Date (in seconds since 1/1/1970) that the camera was first powered on, after shipping. | | EEPROM |

COLR– Image Color Setting

| | |
|---------------------|--|
| Definition: | Controls the color mode used for image capture. Overrides the parameters <code>sccl</code> and <code>tccl</code> . |
| Data Type: | Enum list |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Volatile |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|------------------------------|-----------------|-----------|
| HP PhotoSmart C618 | 1=Full color, 2=B&W, 3=Sepia | 1 | RAM |
| HP PhotoSmart C912 | 1=Full color, 2=B&W, 3=Sepia | 1 | RAM |
| Pentax EI-200 | 1=Full color, 2=B&W, 3=Sepia | 1 | RAM |
| Pentax EI-2000 | 1=Full color, 2=B&W, 3=Sepia | 1 | RAM |

CONT– Continuous Mode

| | |
|---------------------|---|
| Definition: | Activates continuous still capture |
| Data Type: | Enum list |
| Commands to Access: | <code>GetCameraState</code> , <code>SetCameraState</code> |
| Value: | Volatile |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|-------------|-----------------|-----------|
| HP PhotoSmart C618 | 0=Off, 1=On | 0 | RAM |
| HP PhotoSmart C912 | 0=Off, 1=On | 0 | RAM |
| Pentax EI-200 | 0=Off, 1=On | 0 | RAM |
| Pentax EI-2000 | 0=Off, 1=On | 0 | RAM |

DBRT– Display Brightness

Definition: Display brightness control
 Data Type: UInt
 Commands to Access: GetCameraState, SetCameraState
 Value: Persistent
 Access: Read/Write

| Product/OS | Details | Factory Default | Stored In |
|--------------------|----------------------|-----------------|-----------|
| HP PhotoSmart C500 | Minimum=1, Maximum=7 | 4 | EEPROM |
| HP PhotoSmart C618 | Minimum=1, Maximum=7 | 4 | EEPROM |
| HP PhotoSmart C912 | Minimum=1, Maximum=7 | 4 | EEPROM |
| Pentax EI-200 | Minimum=1, Maximum=7 | 4 | EEPROM |
| Pentax EI-2000 | Minimum=1, Maximum=7 | 4 | EEPROM |

EYES–Eye Start

Definition: Eye start control
 Data Type: Enum list
 Commands to Access: GetCameraState, SetCameraState
 Value: Volatile
 Access: Read/Write

| Product/OS | Details | Factory Default | Stored In |
|--------------------|-----------------------------------|-----------------|-----------|
| HP PhotoSmart C500 | 0=On, 1=Off; Minimum=0, Maximum=1 | 0 | EEPROM |

EYEU–Eye Start User Control

Definition: User enable or disable eye start settings
 Data Type: Enum list
 Commands to Access: GetCameraState, SetCameraState
 Value: Volatile
 Access: Read/Write

| Product/OS | Details | Factory Default | Stored In |
|--------------------|-----------------------------------|-----------------|-----------|
| HP PhotoSmart C500 | 0=Off, 1=On; Minimum=0, Maximum=1 | 1 | EEPROM |

FOTO–Photo Mode Setting

Definition: Reads the current photo mode dial switch position
Data Type: Enum list
Commands to Access: `GetProductInfo`
Value: Read Only
Access: None

| Product/OS | Details | Factory Default | Stored In |
|--------------------|---|-----------------|-----------|
| HP PhotoSmart C912 | Auto=65, Portrait=66, Landscape=67, Action=68, Macro=69, Night=70, P=71, Av=72, Tv=73, M=74 | | |
| Pentax EI-2000 | Auto=65, Portrait=66, Landscape=67, Action=68, Macro=69, Night=70, P=71, Av=72, Tv=73, M=74 | | |

HAND–Hand-Held

Definition: Hand-held, when turned on, disallows long shutter speeds
Data Type: Enum list
Commands to Access: `GetCameraState`, `SetCameraState`
Value: Volatile
Access: Read/Write

| Product/OS | Details | Factory Default | Stored In |
|--------------------|-----------------------------------|-----------------|-----------|
| HP PhotoSmart C500 | 0=Off, 1=On; Minimum=0, Maximum=1 | 1 | EEPROM |

LVON–Liveview On

Definition: Controls whether LiveView is on or off at power-up and mode changes
Data Type: Enum list
Commands to Access: `GetCameraState`, `SetCameraState`
Value: Persistent
Access: Read/Write

| Product/OS | Details | Factory Default | Stored In |
|--------------------|-------------|-----------------|-----------|
| HP PhotoSmart C618 | 0=Off, 1=On | 0 | EEPROM |
| HP PhotoSmart C912 | 0=Off, 1=On | 0 | EEPROM |
| Pentax EI-200 | 0=Off, 1=On | 0 | EEPROM |
| Pentax EI-2000 | 0=Off, 1=On | 0 | EEPROM |

OVLY–Overlay Mode

| | |
|---------------------|--|
| Definition: | Controls the display of the 3rd line of information in the Play mode overlay |
| Data Type: | Enum list |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Persistent |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|--------------------|--------------------------------------|-----------------|-----------|
| HP PhotoSmart C912 | 1=Standard 2 line; 2=Expanded 3 line | 1 | EEPROM |
| Pentax EI-2000 | 1=Standard 2 line; 2=Expanded 3 line | 1 | EEPROM |

RSHT–Remote Shutter Enable

| | |
|---------------------|--|
| Definition: | Enables remote shutter release; requires a remote shutter release device, connected to the serial port |
| Data Type: | Enum list |
| Commands to Access: | GetCameraState, SetCameraState |
| Value: | Volatile |
| Access: | Read/Write |

| Product/OS | Details | Factory Default | Stored In |
|-------------|-----------------------------------|-----------------|-----------|
| Kodak DC290 | 0=Off, 1=On; Minimum=0, Maximum=1 | 0 | RAM |

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 | Tag not supported |
| 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. |
| HP PhotoSmart C500 | This field is variable and contains a value returned from the CCS. |
| HP PhotoSmart C618 | This field is variable and contains a value returned from the CCS. |
| HP PhotoSmart C912 | This field is variable and contains a value returned from the CCS. |
| Pentax EI-200 | This field is variable and contains a value returned from the CCS. |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | Tag not supported |
| HP PhotoSmart C618 | This field is variable in 0.01 unit, F4.0 = 400, F3.0=300, F5.6=560, F8.0=800 |
| HP PhotoSmart C912 | This field is variable in 0.01 unit, F4.0 = 400, F3.0=300, F5.6=560, F8.0=800 |
| Pentax EI-200 | This field is variable in 0.01 unit, F4.0 = 400, F3.0=300, F5.6=560, F8.0=800 |
| Pentax EI-2000 | This field is variable in 0.01 unit, F4.0 = 400, F3.0=300, F5.6=560, F8.0=800 |

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. |
| HP PhotoSmart C500 | Tag not supported Replaced by cngn |
| HP PhotoSmart C618 | Tag not supported Replaced by cngn |
| HP PhotoSmart C912 | Tag not supported Replaced by cngn |
| Pentax EI-200 | Tag not supported Replaced by cngn |
| Pentax EI-2000 | Tag not supported Replaced by cngn |

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 |
| HP PhotoSmart C500 | The firmware version for this CCS, e.g., Version 2.0.3.a is 0x0200030a |
| HP PhotoSmart C618 | The firmware version for this CCS, e.g., Version 2.0.3.a is 0x0200030a |
| HP PhotoSmart C912 | The firmware version for this CCS, e.g., Version 2.0.3.a is 0x0200030a |
| Pentax EI-200 | The firmware version for this CCS, e.g., Version 2.0.3.a is 0x0200030a |
| Pentax EI-2000 | The firmware version for this CCS, e.g., Version 2.0.3.a is 0x0200030a |

cmpn – Image Compression Level

Data Type: UInt

| Product | Definition |
|--------------------|---|
| Kodak DC 220 | The degree of image compression, where 2=good, 3=better and 4=best |
| Kodak DC 260 | The degree of image compression, where 2=good, 3=better and 4=best |
| Kodak DC 265 | The degree of image compression, where 2=good, 3=better, 4=best and 5=super |
| Kodak DC 290 | The degree of image compression, where 2=good, 3=better, 4=best and 7=Uncompressed |
| Dimage EX 1500 | The degree of image compression, where 2=economy, 3=standard, 4=fine and 5=super fine |
| HP PhotoSmart C500 | The degree of image compression, where 2=good, 3=better and 4=best |
| HP PhotoSmart C618 | The degree of image compression, where 2=good, 3=better and 4=best |
| HP PhotoSmart C912 | The degree of image compression, where 2=good, 3=better and 4=best |
| Pentax EI-200 | The degree of image compression, where 2=good, 3=better and 4=best |
| Pentax EI-2000 | The degree of image compression, where 2=good, 3=better and 4=best |

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 |
| HP PhotoSmart C500 | Tag not supported Replaced by rgnc |
| HP PhotoSmart C618 | Tag not supported Replaced by rgnc |
| HP PhotoSmart C912 | Tag not supported Replaced by rgnc |
| Pentax EI-200 | Tag not supported Replaced by rgnc |
| Pentax EI-2000 | Tag not supported Replaced by rgnc |

cmty – 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 |
| HP PhotoSmart C500 | 1="None", 2="JPEG" |
| HP PhotoSmart C618 | 1="None", 2="JPEG" |
| HP PhotoSmart C912 | 1="None", 2="JPEG" |
| Pentax EI-200 | 1="None", 2="JPEG" |
| Pentax EI-2000 | 1="None", 2="JPEG" |

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 | Folder name specified by the StartCapture command. This field is variable and contains a DOS file name without a DOS extension |
| Dimage EX 1500 | Tag not supported |
| HP PhotoSmart C500 | Folder name specified by the StartCapture command. This field is variable and contains a DOS file name without a DOS extension |
| HP PhotoSmart C618 | Folder name specified by the StartCapture command. This field is variable and contains a DOS file name without a DOS extension |
| HP PhotoSmart C912 | Folder name specified by the StartCapture command. This field is variable and contains a DOS file name without a DOS extension |
| Pentax EI-200 | Folder name specified by the StartCapture command. This field is variable and contains a DOS file name without a DOS extension |
| Pentax EI-2000 | Folder name specified by the StartCapture command. This field is variable and contains a DOS file name without a DOS extension |

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 | Data supplied to group capture command; this field is variable and contains a DOS file name |
| Dimage EX 1500 | Tag not supported |
| HP PhotoSmart C500 | Data supplied to group capture command; this field is variable and contains a DOS file name |
| HP PhotoSmart C618 | Data supplied to group capture command; this field is variable and contains a DOS file name |
| HP PhotoSmart C912 | Data supplied to group capture command; this field is variable and contains a DOS file name |
| Pentax EI-200 | Data supplied to group capture command; this field is variable and contains a DOS file name |
| Pentax EI-2000 | Data supplied to group capture command; this field is variable and contains a DOS file name |

date – Image Capture Date

Data Type: UInt

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | The image capture date For example, 0x00031595 is March 15, 95 |
| Kodak DC 260/DC 265 | The image capture date For example, 0x00031595 is March 15, 95 |
| Kodak DC 290 | The image capture date Value in MM/DD/YY format - Example 12/24/99 = 122499 |
| Dimage EX 1500 | The image capture date For example, 0x00031595 is March 15, 95 |
| HP PhotoSmart C500 | The image capture date Value in MM/DD/YY format - Example 12/24/99 = 122499 |
| HP PhotoSmart C618 | The image capture date Value in MM/DD/YY format - Example 12/24/99 = 122499 |
| HP PhotoSmart C912 | The image capture date Value in MM/DD/YY format - Example 12/24/99 = 122499 |
| Pentax EI-200 | The image capture date Value in MM/DD/YY format - Example 12/24/99 = 122499 |
| Pentax EI-2000 | The image capture date Value in MM/DD/YY format - Example 12/24/99 = 122499 |

eval – Exposure Value

Data Type: UInt

| eval | Definition |
|---------------------|---|
| Kodak DC 220 | The exposure value from the CCS Image information, in 0.01 EV units |
| Kodak DC 260/DC 265 | The exposure value from the CCS Image information, in 0.01 EV units |
| Kodak DC 290 | The exposure value from the CCS Image information, in 0.01 EV units |
| Dimage EX 1500 | The exposure value from the CCS Image information, in 0.01 EV units |
| HP PhotoSmart C500 | The exposure value from the CCS Image information, in 0.01 EV units |
| HP PhotoSmart C618 | The exposure value from the CCS Image information, in 0.01 EV units |
| HP PhotoSmart C912 | The exposure value from the CCS Image information, in 0.01 EV units |
| Pentax EI-200 | The exposure value from the CCS Image information, in 0.01 EV units |
| Pentax EI-2000 | The exposure value from the CCS Image information, 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 |
| HP PhotoSmart C500 | The extracted image height |
| HP PhotoSmart C618 | The extracted image height |
| HP PhotoSmart C912 | The extracted image height |
| Pentax EI-200 | The extracted image height |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | The extracted image width |
| HP PhotoSmart C618 | The extracted image width |
| HP PhotoSmart C912 | The extracted image width |
| Pentax EI-200 | The extracted image width |
| Pentax EI-2000 | The extracted image width |

fdst – Focus Distance

Data Type: UInt

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | Tag 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 | Tag not supported |
| HP PhotoSmart C500 | Tag not supported |
| HP PhotoSmart C618 | Focus distance — maximum value is 65535, which equals infinity |
| HP PhotoSmart C912 | Focus distance — maximum value is 65535, which equals infinity |
| Pentax EI-200 | Focus distance — maximum value is 65535, which equals infinity |
| Pentax EI-2000 | Focus distance — maximum value is 65535, which equals infinity |

flty – Image File Type

Data Type: Enum

| Product | Definition |
|---------------------|------------------------------|
| Kodak DC 220 | Tag not supported (see imcn) |
| Kodak DC 260/DC 265 | Tag not supported (see imcn) |
| Kodak DC 290 | 2=EXIF, 4=TIFF |
| Dimage EX 1500 | Tag not supported (see imcn) |
| HP PhotoSmart C500 | 2=EXIF |
| HP PhotoSmart C618 | 2=JPEG, 4=TIFF |
| HP PhotoSmart C912 | 2=JPEG, 4=TIFF |
| Pentax EI-200 | 2=JPEG, 4=TIFF |
| Pentax EI-2000 | 2=JPEG, 4=TIFF |

fmod – Focus Mode

Data Type: UInt

| Product | Definition |
|---------------------|---|
| Kodak DC 220 | Tag not supported |
| Kodak DC 260/DC 265 | The focus mode, where 1=Auto- Focus and 3=Manual Focus |
| Kodak DC 290 | The focus mode, where 1= Auto Focus and 3= Manual Focus |
| Dimage EX 1500 | The focus mode, where 1=Auto Focus and 3= Manual Focus |
| HP PhotoSmart C500 | The focus mode, where 1=Auto Focus |
| HP PhotoSmart C618 | The focus mode, where 1= Auto Focus and 2=Manual Focus |
| HP PhotoSmart C912 | The focus mode, where 1= Auto Focus and 2=Manual Focus |
| Pentax EI-200 | The focus mode, where 1= Auto Focus and 2=Manual Focus |
| Pentax EI-2000 | The focus mode, where 1= Auto Focus and 2=Manual Focus |

fmtd – Focus Method

Data Type: Enum

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | Tag 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 |
| HP PhotoSmart C500 | The focus method, where 1=Auto Matrix |
| HP PhotoSmart C618 | The focus method, where 4=Center Weighted, |
| HP PhotoSmart C912 | The focus method, where 2=center-spot and 4=Center Weighted, |
| Pentax EI-200 | The focus method, where 4=Center Weighted, |
| Pentax EI-2000 | The focus method, where 2=center-spot and 4=Center Weighted, |

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. |
| HP PhotoSmart C500 | Tag not supported Replaced by aper |
| HP PhotoSmart C618 | Tag not supported Replaced by aper |
| HP PhotoSmart C912 | Tag not supported Replaced by aper |
| Pentax EI-200 | Tag not supported Replaced by aper |
| Pentax EI-2000 | Tag not supported Replaced by aper |

fwrv – Digita Firmware Version

Data Type: UInt

| Product | Definition |
|---------------------|---|
| Kodak DC 220 | The Digita firmware version number, e.g., Version 1.0 is 0x01000000 |
| Kodak DC 260/DC 265 | The Digita firmware version number, e.g., Version 1.0 is 0x01000000 |
| Kodak DC 290 | The Digita firmware version number, e.g., Version 1.0.2 is 0x01000200 |
| Dimage EX 1500 | The Digita firmware version number, e.g., 168886272 |
| HP PhotoSmart C500 | The Digita firmware version number, e.g., Version 1.0.6 is 0x01000600 |
| HP PhotoSmart C618 | The Digita firmware version number, e.g., Version 1.0 is 0x01000000 |
| HP PhotoSmart C912 | The Digita firmware version number, e.g., Version 1.0 is 0x01000000 |
| Pentax EI-200 | The Digita firmware version number, e.g., Version 1.0 is 0x01000000 |
| Pentax EI-2000 | The Digita firmware version number, e.g., Version 1.0 is 0x01000000 |

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 cpjn |
| Dimage EX 1500 | Capture group folder DOS name, with no extension |
| HP PhotoSmart C500 | Tag not supported Replaced by cpjn |
| HP PhotoSmart C618 | Tag not supported Replaced by cpjn |
| HP PhotoSmart C912 | Tag not supported Replaced by cpjn |
| Pentax EI-200 | Tag not supported Replaced by cpjn |
| Pentax EI-2000 | Tag not supported Replaced by cpjn |

hint – Image Hint Mode

Data Type: UInt

| Product | Definition |
|---------------------|---|
| Kodak DC 220 | Tag not supported |
| Kodak DC 260/DC 265 | Tag 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 |
| HP PhotoSmart C500 | Tag not supported |
| HP PhotoSmart C618 | Hint mode, where 1=Auto, 2=Portrait, 3=Landscape, 5=Sport |
| HP PhotoSmart C912 | Hint mode, where 1=Auto, 2=Portrait, 3=Landscape, 4=Closeup, 5=Sport and 7= Night |
| Pentax EI-200 | Hint mode, where 1=Auto, 2=Portrait, 3=Landscape, 5=Sport |
| Pentax EI-2000 | Hint mode, where 1=Auto, 2=Portrait, 3=Landscape, 4=Closeup, 5=Sport and 7= Night |

hwrv – Hardware Version

Data Type: UInt

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | Tag not supported |
| Kodak DC 260/DC 265 | Tag not supported |
| Kodak DC 290 | Hardware version number 0x01000000 = version 1.0 |
| Dimage EX 1500 | Hardware version number 0x01000000 = version 1.0 |
| HP PhotoSmart C500 | Hardware version number 0x01000000 = version 1.0 |
| HP PhotoSmart C618 | Hardware version number 0x01000000 = version 1.0 |
| HP PhotoSmart C912 | Hardware version number 0x01000000 = version 1.0 |
| Pentax EI-200 | Hardware version number 0x01000000 = version 1.0 |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | Tag not supported |
| HP PhotoSmart C618 | Tag not supported |
| HP PhotoSmart C912 | Tag not supported |
| Pentax EI-200 | Tag not supported |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | Version number 0x01000100 = version 1.0.1 |
| HP PhotoSmart C618 | Version number 0x01000100 = version 1.0.1 |
| HP PhotoSmart C912 | Version number 0x01000100 = version 1.0.1 |
| Pentax EI-200 | Version number 0x01000100 = version 1.0.1 |
| Pentax EI-2000 | Version number 0x01000100 = version 1.0.1 |

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) |
| HP PhotoSmart C500 | Tag not supported Replaced by flty |
| HP PhotoSmart C618 | Tag not supported Replaced by flty |
| HP PhotoSmart C912 | Tag not supported Replaced by flty |
| Pentax EI-200 | Tag not supported Replaced by flty |
| Pentax EI-2000 | Tag not supported Replaced by flty |

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) |
| Kodak DC 260/DC 265 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| Kodak DC 290 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| Dimage EX 1500 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| HP PhotoSmart C500 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| HP PhotoSmart C618 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| HP PhotoSmart C912 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| Pentax EI-200 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |
| Pentax EI-2000 | Any of the values 0b00 to 001101100 (underexposure, overexposure, not supported, macro on, flash fired, not supported) |

imht – Image Height

Data Type: UInt

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | Image height - different from Extract Height if resizing was done after processing |
| Kodak DC 260/DC 265 | Image height - different from Extract Height if resizing was done after processing |
| Kodak DC 290 | Image height - different from Extract Height if resizing was done after processing |
| Dimage EX 1500 | Image height - different from Extract Height if resizing was done after processing |
| HP PhotoSmart C500 | Image height - different from Extract Height if resizing was done after processing |
| HP PhotoSmart C618 | Image height - different from Extract Height if resizing was done after processing |
| HP PhotoSmart C912 | Image height - different from Extract Height if resizing was done after processing |
| Pentax EI-200 | Image height - different from Extract Height if resizing was done after processing |
| Pentax EI-2000 | Image height - different from Extract Height if resizing was done after processing |

imis – Image Info Size

Data Type: UInt

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | 1024 bytes - size of file tag data block |
| Kodak DC 260/DC 265 | 1024 bytes - size of file tag data block |
| Kodak DC 290 | 1024 bytes - size of file tag data block |
| Dimage EX 1500 | 1024 bytes - size of file tag data block |
| HP PhotoSmart C500 | 1024 bytes - size of file tag data block |
| HP PhotoSmart C618 | 1024 bytes - size of file tag data block |
| HP PhotoSmart C912 | 1024 bytes - size of file tag data block |
| Pentax EI-200 | 1024 bytes - size of file tag data block |
| Pentax EI-2000 | 1024 bytes - size of file tag data block |

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 |
| HP PhotoSmart C500 | Tag not supported Replaced by iirv |
| HP PhotoSmart C618 | Tag not supported Replaced by iirv |
| HP PhotoSmart C912 | Tag not supported Replaced by iirv |
| Pentax EI-200 | Tag not supported Replaced by iirv |
| Pentax EI-2000 | Tag not supported Replaced by iirv |

imprn – Image Color Depth

Data Type: Enum

| Product | Definition |
|---------------------|--------------------------|
| Kodak DC 220 | 1=color |
| Kodak DC 260/DC 265 | 1=color |
| Kodak DC 290 | 1=color |
| Dimage EX 1500 | 1=color, 2=black/white |
| HP PhotoSmart C500 | 1=Color, 2=Gray |
| HP PhotoSmart C618 | 1=Color, 2=Gray, 3=Sepia |
| HP PhotoSmart C912 | 1=Color, 2=Gray, 3=Sepia |
| Pentax EI-200 | 1=Color, 2=Gray, 3=Sepia |
| Pentax EI-2000 | 1=Color, 2=Gray, 3=Sepia |

imwd – Image Width

Data Type: UInt

| Product | Definition |
|---------------------|--|
| Kodak DC 220 | Image width - different from Extract Width if resizing was done after processing |
| Kodak DC 260/DC 265 | Image width - different from Extract Width if resizing was done after processing |
| Kodak DC 290 | Image width - different from Extract Width if resizing was done after processing |
| Dimage EX 1500 | Image width - different from Extract Width if resizing was done after processing |
| HP PhotoSmart C500 | Image width - different from Extract Width if resizing was done after processing |
| HP PhotoSmart C618 | Image width - different from Extract Width if resizing was done after processing |
| HP PhotoSmart C912 | Image width - different from Extract Width if resizing was done after processing |
| Pentax EI-200 | Image width - different from Extract Width if resizing was done after processing |
| Pentax EI-2000 | Image width - different from Extract Width if resizing was done after processing |

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 |
| HP PhotoSmart C500 | Tag not supported |
| HP PhotoSmart C618 | Field is variable Minimum=0x1999, Maximum=0xa0000 |
| HP PhotoSmart C912 | Field is variable Minimum=0x1999, Maximum=0xa0000 |
| Pentax EI-200 | Field is variable Minimum=0x1999, Maximum=0xa0000 |
| Pentax EI-2000 | Field is variable Minimum=0x1999, Maximum=0xa0000 |

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 |
| HP PhotoSmart C500 | Tag not supported |
| HP PhotoSmart C618 | Field is variable Minimum=0, Maximum=25 |
| HP PhotoSmart C912 | Field is variable Minimum=0, Maximum=25 |
| Pentax EI-200 | Field is variable Minimum=0, Maximum=25 |
| Pentax EI-2000 | Field is variable Minimum=0, Maximum=25 |

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) |
| HP PhotoSmart C500 | Auto-exposure/auto-focus mode Set to 1 (AE locked) |
| HP PhotoSmart C618 | Auto-exposure/auto-focus mode Set to 3 (AE and AF locked) |
| HP PhotoSmart C912 | Auto-exposure/auto-focus mode Set to 3 (AE and AF locked) |
| Pentax EI-200 | Auto-exposure/auto-focus mode Set to 3 (AE and AF locked) |
| Pentax EI-2000 | 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=Burst, 3=TimeLapse |
| Dimage EX 1500 | Tag not supported |
| HP PhotoSmart C500 | 1=Still, 2=Burst, 3=TimeLapse |
| HP PhotoSmart C618 | 1=Still and 3=TimeLapse |
| HP PhotoSmart C912 | 1=Still and 3=TimeLapse |
| Pentax EI-200 | 1=Still and 3=TimeLapse |
| Pentax EI-2000 | 1=Still and 3=TimeLapse |

ortn – 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 |
| HP PhotoSmart C500 | Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus |
| HP PhotoSmart C618 | Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus |
| HP PhotoSmart C912 | Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus |
| Pentax EI-200 | Capture orientation: 1=landscape, 2=portrait plus, 3=portrait minus |
| Pentax EI-2000 | 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 | Absolute image number (odometer) for this camera. May also be used to create file name. |
| Dimage EX 1500 | Tag not supported |
| HP PhotoSmart C500 | Absolute image number (odometer) for this camera. May also be used to create file name. |
| HP PhotoSmart C618 | Absolute image number (odometer) for this camera. May also be used to create file name. |
| HP PhotoSmart C912 | Absolute image number (odometer) for this camera. May also be used to create file name. |
| Pentax EI-200 | Absolute image number (odometer) for this camera. May also be used to create file name. |
| Pentax EI-2000 | Absolute image number (odometer) for this camera. May also be used to create file name. |

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 |
| HP PhotoSmart C500 | HP PhotoSmart C500 |
| HP PhotoSmart C618 | HP PhotoSmart C618 |
| HP PhotoSmart C912 | HP PhotoSmart C912 |
| Pentax EI-200 | PENTAX EI-200 |
| Pentax EI-2000 | PENTAX EI-2000 |

rgnc – Region Code

Data Type: SInt

| Product | Definition |
|---------------------|---|
| Kodak DC 220 | Tag not supported |
| Kodak DC 260/DC 265 | Tag not supported |
| Kodak DC 290 | 1=English, 3=French, 6=German, 8=Japanese |
| Dimage EX 1500 | Tag not supported |
| HP PhotoSmart C500 | 1=English, 3=French, 4=Italian, 5=Spanish, 6=German |
| HP PhotoSmart C618 | 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| HP PhotoSmart C912 | 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-200 | 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-2000 | 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |

rmod – Auto Image Rotation Mode

Data Type: UInt

| Product | Definition |
|---------------------|-------------------|
| Kodak DC 220 | Tag not supported |
| Kodak DC 260/DC 265 | Tag not supported |
| Kodak DC 290 | 0=OFF, 1=ON |
| Dimage EX 1500 | Tag not supported |
| HP PhotoSmart C500 | 0=OFF, 1=ON |
| HP PhotoSmart C618 | 0=OFF, 1=ON |
| HP PhotoSmart C912 | 0=OFF, 1=ON |
| Pentax EI-200 | 0=OFF, 1=ON |
| Pentax EI-2000 | 0=OFF, 1=ON |

scmp – Strobe Compensation

Data Type: SInt

| scmp | 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 | Values in increments of 0.01EV |
| HP PhotoSmart C500 | Values in increments of 0.01 f/stop steps |
| HP PhotoSmart C618 | Minimum = -200, Maximum =200 Values in increments of 0.01 f/stop steps |
| HP PhotoSmart C912 | Minimum = -200, Maximum =200 Values in increments of 0.01 f/stop steps |
| Pentax EI-200 | Minimum = -200, Maximum =200 Values in increments of 0.01 f/stop steps |
| Pentax EI-2000 | Minimum = -200, Maximum =200 Values in increments of 0.01 f/stop steps |

scom – Sound Compression Type

Data Type: UInt

| Product | Definition |
|---------------------|---------------------|
| Kodak DC 220 | 0=No Sound, 2=ADPCM |
| Kodak DC 260/DC 265 | 0=No Sound, 2=ADPCM |
| Kodak DC 290 | 0=No Sound, 2=ADPCM |
| Dimage EX 1500 | 0=No Sound, 2=ADPCM |
| HP PhotoSmart C500 | 0=No Sound, 2=ADPCM |
| HP PhotoSmart C618 | 0=No Sound, 2=ADPCM |
| HP PhotoSmart C912 | 0=No Sound, 2=ADPCM |
| Pentax EI-200 | 0=No Sound, 2=ADPCM |
| Pentax EI-2000 | 0=No Sound, 2=ADPCM |

sdsr – Sound Sampling Rate

Data Type: UInt

| Product | Definition |
|---------------------|-------------------------------------|
| Kodak DC 220 | 0=No Sound, 11025=Sound (11.025KHz) |
| Kodak DC 260/DC 265 | 0=No Sound, 11025=Sound (11.025KHz) |
| Kodak DC 290 | 0=No Sound, 11025=Sound (11.025KHz) |
| Dimage EX 1500 | 0=No Sound, 11025=Sound (11.025KHz) |
| HP PhotoSmart C500 | 0=No Sound, 11025=Sound (11.025KHz) |
| HP PhotoSmart C618 | 0=No Sound, 11025=Sound (11.025KHz) |
| HP PhotoSmart C912 | 0=No Sound, 11025=Sound (11.025KHz) |
| Pentax EI-200 | 0=No Sound, 11025=Sound (11.025KHz) |
| Pentax EI-2000 | 0=No Sound, 11025=Sound (11.025KHz) |

sdss – Sound Sampling Size

Data Type: UInt

| Product | Definition |
|---------------------|----------------------|
| Kodak DC 220 | 0=No Sound, 16=Sound |
| Kodak DC 260/DC 265 | 0=No Sound, 16=Sound |
| Kodak DC 290 | 0=No Sound, 16=Sound |
| Dimage EX 1500 | 0=No Sound, 16=Sound |
| HP PhotoSmart C500 | 0=No Sound, 16=Sound |
| HP PhotoSmart C618 | 0=No Sound, 16=Sound |
| HP PhotoSmart C912 | 0=No Sound, 16=Sound |
| Pentax EI-200 | 0=No Sound, 16=Sound |
| Pentax EI-2000 | 0=No Sound, 16=Sound |

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 |
| HP PhotoSmart C500 | Serial number |
| HP PhotoSmart C618 | Serial number |
| HP PhotoSmart C912 | Serial number |
| Pentax EI-200 | Serial number |
| Pentax EI-2000 | Serial number |

shut – Shutter Speed

Data Type: UInt

| Product | Definition |
|---------------------|--------------------------------|
| Kodak DC 220 | Shutter speed, units=microsecs |
| Kodak DC 260/DC 265 | Shutter speed, units=microsecs |
| Kodak DC 290 | Shutter speed, units=microsecs |
| Dimage EX 1500 | Shutter speed, units=microsecs |
| HP PhotoSmart C500 | Shutter speed, units=microsecs |
| HP PhotoSmart C618 | Shutter speed, units=microsecs |
| HP PhotoSmart C912 | Shutter speed, units=microsecs |
| Pentax EI-200 | Shutter speed, units=microsecs |
| Pentax EI-2000 | Shutter speed, units=microsecs |

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=Auto, 2=On, 3=Off, 5=External Sync |
| Dimage EX 1500 | Strobe mode, where 1=Off, 2=Auto, 3=Fill |
| HP PhotoSmart C500 | Strobe mode, where 1=Auto, 2=On, 3=Off |
| HP PhotoSmart C618 | Strobe mode, where 1=Auto, 2=On, 3=Off, 4=Slave |
| HP PhotoSmart C912 | Strobe mode, where 1=Auto, 2=On, 3=Off, 5=External Sync |
| Pentax EI-200 | Strobe mode, where 1=Auto, 2=On, 3=Off, 4=Slave |
| Pentax EI-2000 | Strobe mode, where 1=Auto, 2=On, 3=Off, 5=External Sync |

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 |
| HP PhotoSmart C500 | Strobe precount, where 1=red-eye reduction |
| HP PhotoSmart C618 | Strobe precount, where 1=red-eye reduction |
| HP PhotoSmart C912 | Strobe precount, where 1=red-eye reduction |
| Pentax EI-200 | Strobe precount, where 1=red-eye reduction |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | Image capture time, as for example 0x00211530 for 9:15:30 pm |
| HP PhotoSmart C618 | Image capture time, as for example 0x00211530 for 9:15:30 pm |
| HP PhotoSmart C912 | Image capture time, as for example 0x00211530 for 9:15:30 pm |
| Pentax EI-200 | Image capture time, as for example 0x00211530 for 9:15:30 pm |
| Pentax EI-2000 | 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. |
| HP PhotoSmart C500 | The default is 0 for no null string. This field can be set to any value by the user. |
| HP PhotoSmart C618 | The default is 0 for no null string. This field can be set to any value by the user. |
| HP PhotoSmart C912 | The default is 0 for no null string. This field can be set to any value by the user. |
| Pentax EI-200 | The default is 0 for no null string. This field can be set to any value by the user. |
| Pentax EI-2000 | 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, "" |
| HP PhotoSmart C500 | User caption text The default value is the null string, "" |
| HP PhotoSmart C618 | User caption text The default value is the null string, "" |
| HP PhotoSmart C912 | User caption text The default value is the null string, "" |
| Pentax EI-200 | User caption text The default value is the null string, "" |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | User copyright notice |
| HP PhotoSmart C618 | User copyright notice |
| HP PhotoSmart C912 | User copyright notice |
| Pentax EI-200 | User copyright notice |
| Pentax EI-2000 | 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. |
| HP PhotoSmart C500 | User copyright region code; defaults to 0 for no null string This field can be set to any value by the user. |
| HP PhotoSmart C618 | This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| HP PhotoSmart C912 | This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-200 | This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-2000 | This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |

uc01-uc16 – User Categories 01-16

Data Type: String

| 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, "" |
| HP PhotoSmart C500 | User category The default value is the null string, "" |
| HP PhotoSmart C618 | User category - 15 character "short" provided by the user The default value is the null string, "" |
| HP PhotoSmart C912 | User category - 15 character "short" provided by the user The default value is the null string, "" |
| Pentax EI-200 | User category - 15 character "short" provided by the user The default value is the null string, "" |
| Pentax EI-2000 | User category - 15 character "short" provided by the user 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 |
| HP PhotoSmart C500 | User data |
| HP PhotoSmart C618 | User supplied data |
| HP PhotoSmart C912 | User supplied data |
| Pentax EI-200 | User supplied data |
| Pentax EI-2000 | User supplied data |

uptd – User Protected

Data Type: UInt

| Product | Definition |
|---------------------|-----------------------------|
| Kodak DC 220 | 0=nonprotected, 1=protected |
| Kodak DC 260/DC 265 | 0=nonprotected, 1=protected |
| Kodak DC 290 | 0=nonprotected, 1=protected |
| Dimage EX 1500 | 0=nonprotected, 1=protected |
| HP PhotoSmart C500 | 0=nonprotected, 1=protected |
| HP PhotoSmart C618 | 0=nonprotected, 1=protected |
| HP PhotoSmart C912 | 0=nonprotected, 1=protected |
| Pentax EI-200 | 0=nonprotected, 1=protected |
| Pentax EI-2000 | 0=nonprotected, 1=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. |
| HP PhotoSmart C500 | User category region code; defaults to 0 for no null string This field can be set to any value by the user |
| HP PhotoSmart C618 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| HP PhotoSmart C912 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-200 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-2000 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |

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. |
| HP PhotoSmart C500 | User string region code; defaults to 0 for no null string This field can be set to any value by the user. |
| HP PhotoSmart C618 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| HP PhotoSmart C912 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-200 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |
| Pentax EI-2000 | User category region code; defaults to 0 for no null string This field can be set to any value by the user. 1=English, 3=French, 4=Italian, 5=Spanish, 6=German, 8=Japanese |

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, "" |
| HP PhotoSmart C500 | User string; the default value is the null string, "" |
| HP PhotoSmart C618 | User string; the default value is the null string, "" |
| HP PhotoSmart C912 | User string; the default value is the null string, "" |
| Pentax EI-200 | User string; the default value is the null string, "" |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | The user tag identifier |
| HP PhotoSmart C618 | The user tag identifier |
| HP PhotoSmart C912 | The user tag identifier |
| Pentax EI-200 | The user tag identifier |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | Hewlett Packard |
| HP PhotoSmart C618 | Hewlett-Packard |
| HP PhotoSmart C912 | Hewlett-Packard |
| Pentax EI-200 | PENTAX |
| Pentax EI-2000 | PENTAX |

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 |
| HP PhotoSmart C500 | Vendor data |
| HP PhotoSmart C618 | Vendor data |
| HP PhotoSmart C912 | Vendor data |
| Pentax EI-200 | Vendor data |
| Pentax EI-2000 | 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 |
| HP PhotoSmart C500 | White balance blue |
| HP PhotoSmart C618 | Tag not supported (0 x 0 returned) |
| HP PhotoSmart C912 | Tag not supported (0 x 0 returned) |
| Pentax EI-200 | Tag not supported (0 x 0 returned) |
| Pentax EI-2000 | Tag not supported (0 x 0 returned) |

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 |
| HP PhotoSmart C500 | White balance green |
| HP PhotoSmart C618 | Tag not supported (0 x 0 returned) |
| HP PhotoSmart C912 | Tag not supported (0 x 0 returned) |
| Pentax EI-200 | Tag not supported (0 x 0 returned) |
| Pentax EI-2000 | Tag not supported (0 x 0 returned) |

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 |
| HP PhotoSmart C500 | White balance red |
| HP PhotoSmart C618 | Tag not supported (0 x 0 returned) |
| HP PhotoSmart C912 | Tag not supported (0 x 0 returned) |
| Pentax EI-200 | Tag not supported (0 x 0 returned) |
| Pentax EI-2000 | Tag not supported (0 x 0 returned) |

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 |
| Dimage EX 1500 | Indicates the auto-white balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten, 11=off (set manually) |
| HP PhotoSmart C500 | White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten, 10=flash |
| HP PhotoSmart C618 | White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten |
| HP PhotoSmart C912 | White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten |
| Pentax EI-200 | White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten |
| Pentax EI-2000 | White balance mode, where 1=auto, 3=daylight, 6=fluorescent, 9=tungsten |

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 290 | 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 |
| HP PhotoSmart C500 | The region code, where 1=U.S. English, 3=French, 4=Italian, 5=Spanish and 6=German |
| HP PhotoSmart C618 | The region code, where 1=English, 3=French, 4=Italian, 5=Spanish, 6=German and 8=Japanese |
| HP PhotoSmart C912 | The region code, where 1=English, 3=French, 4=Italian, 5=Spanish, 6=German and 8=Japanese |
| Pentax EI-200 | The region code, where 1=English, 3=French, 4=Italian, 5=Spanish, 6=German and 8=Japanese |
| Pentax EI-2000 | The region code, where 1=English, 3=French, 4=Italian, 5=Spanish, 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 |
| HP PhotoSmart C500 | Watermark text string |
| HP PhotoSmart C618 | The watermark text string. The null string is "". |
| HP PhotoSmart C912 | The watermark text string. The null string is "". |
| Pentax EI-200 | The watermark text string. The null string is "". |
| Pentax EI-2000 | The watermark text string. The null string is "". |

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 |
| HP PhotoSmart C500 | Minimum=-150, Maximum=150, Default=0 |
| HP PhotoSmart C618 | Minimum=-300, Maximum=300, Default=0 |
| HP PhotoSmart C912 | Minimum=-300, Maximum=300, Default=0 |
| Pentax EI-200 | Minimum=-300, Maximum=300, Default=0 |
| Pentax EI-2000 | Minimum=-300, Maximum=300, Default=0 |

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=auto-exposure |
| HP PhotoSmart C500 | Exposure mode, where 1=auto-exposure |
| HP PhotoSmart C618 | Exposure mode, where 1=auto-exposure (program AE), 2=shutter priority and 3=aperture priority |
| HP PhotoSmart C912 | Exposure mode, where 1=auto-exposure (program AE), 2=shutter priority and 3=aperture priority, 5=programmed and 6= manual |
| Pentax EI-200 | Exposure mode, where 1=auto-exposure (program AE), 2=shutter priority and 3=aperture priority |
| Pentax EI-2000 | Exposure mode, where 1=auto-exposure (program AE), 2=shutter priority and 3=aperture priority, 5=programmed and 6= manual |

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 |
| HP PhotoSmart C500 | Exposure method, where 1=auto matrix and 4=centerspot |
| HP PhotoSmart C618 | Exposure method, where 3=centerweighted; 4=multi-spot, 5=average |
| HP PhotoSmart C912 | Exposure method, where 3=centerweighted; 4=multi-spot, 5=average |
| Pentax EI-200 | Exposure method, where 3=centerweighted; 4=multi-spot, 5=average |
| Pentax EI-2000 | Exposure method, where 3=centerweighted; 4=multi-spot, 5=average |

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 |
| HP PhotoSmart C500 | 2=Programmed mode.The “zpos” parameter controls the zoom position |
| HP PhotoSmart C618 | 2=Programmed mode The “zpos” parameter controls the zoom position |
| HP PhotoSmart C912 | 2=Programmed mode The “zpos” parameter controls the zoom position |
| Pentax EI-200 | 2=Programmed mode The “zpos” parameter controls the zoom position |
| Pentax EI-2000 | 2=Programmed mode The “zpos” parameter controls the zoom position |

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 |
| HP PhotoSmart C500 | Zoom position |
| HP PhotoSmart C618 | Zoom position |
| HP PhotoSmart C912 | Zoom position |
| Pentax EI-200 | Zoom position |
| Pentax EI-2000 | Zoom position |

Appendix D Date and Time Formats

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

Table 7. U.S. English Time Formats

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

Table 8 gives examples of U.S. English date formats.

Table 8. U.S. English Date Formats

| Date Format | Example |
|---------------|-------------------|
| 1 Short | 3/23/01 |
| 2 Abbreviated | Mar 23, 2001 |
| 3 Long | Thu, Mar 23, 2001 |

Table 9 shows time-hour-minute formats for different regions.

Table 9. Sample Time-Hour-Minute Formats by Region

Note: May vary by manufacturer

| 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 10 shows short-date formats for different regions

Table 10. Sample Short-Date Formats by Region

| Region | Example |
|---------------|------------|
| USEnglish = 1 | 3/23/01 |
| UKEnglish = 2 | 23/3/01 |
| French = 3 | 23/3/01 |
| Italian = 4 | 23-03-2001 |
| Spanish = 5 | 23/3/01 |
| German = 6 | 23.03.2001 |
| Swedish = 7 | 01-03-23 |
| Japanese = 8 | 01/3/23 |

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 | ＝ | 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 | 。 |
| 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 | 午前 |
| F8 | 前後 |
| F9 | 後 |
| FA | 上 |
| FB | 下 |
| FC | 左 |
| FD | 右 |
| FE | 撮 |
| FF | 影 |

Index

A

- Alert 18
- arithmetic assignment 5
- arithmetic operators 13
- assignment
 - arithmetic 5
 - bitwise 6
 - operators 13
 - statement syntax 5
 - string 5
- automatic
 - counting 51

B

- b data type 12
- backslash character 12
- backward reference 10
- batteries, low 24
- baud rate 44
- BitFlag tables 2
- bitmasks E-221
- bitwise assignment 6
- bitwise operators 13

C

- camera
 - capabilities commands 15
 - error 24
 - parameter name-value pair 26
 - parameter, current value 23
 - parameters, restore to defaults 42
 - power level 35
 - powering off 49
 - product information 36
 - restarting with app 43
 - script execution
 - commands 16
 - state, setting 47
 - storage status 37
- Camera Control Manager 30

- capability
 - list items, number of 25
 - type 27
 - value 27
- capture
 - commands 15
 - minimal 35
 - process, ending 19
 - reduced 35
 - sequence 47
 - sequence, starting 51
- captured
 - files, list of 29
 - files, number of 37
 - images, number of 29
- CaptureStatus 23, 25
- capturing images 58
- carriage return 2, 56
- CCD specifications 31
- cerr 24, 28
- clearing error flag 28
- clock
 - commands 16
 - current state of 27
 - setting 47
- close serial port 43
- closing a file 21
- command statement 6
- commands
 - conventions 2
 - error codes 13
 - format 2
- comment statement 7
- compressed files 37
- conditional statements 8
- continuation statements 2
- conventions for commands 2
- counting, automatic 51
- creating a file 21
- current date 27, 47
- current time 27, 47

D

- data
 - alignment 2
 - data type

- value assumptions 13
- date
 - formats, by region D-220
 - formatted 28
 - get current 27
 - setting 47
- default
 - param value, factory 42
 - param value, user 42
 - user values, setting 46
 - values, retrieving 22
- DEFAULT.CAM 43
- delay timer 24
- delete image files 20
- devices, external 44
- dfmt 28
- disk, number of captured images 29
- Display 18
- display an alert 18
- display commands 17
- display feedback text 18
- DisplayClear 19
- DisplayLine 19
- DOS
 - filenames, script files 10
 - name, image files 29

E

- ed1a 24
- ed2a 24
- edch 24
- EndCapture 20
- EraseFile 20
- error
 - camera 24
 - codes 13
 - description 28
 - flag, clearing 28
- expansion
 - disk 1 24
 - disk 1, file info 30, 34
 - disk 2 24
 - disk 2, file info 30, 34
 - disk changed 24
- exposure

- zone specification 32
- external devices 44
- external power status 24
- extp 24
- extracting bits from a bitfield 60

F

- f data type 12
- factory defaults 22, 42
- feedback text 18
- file
 - commands 16
- FileClose 21
- FileOpen 21
- files
 - captured, list of 29
 - captured, number of 37
 - commands 16
 - compressed 37
 - creating 21
 - list status change 24
 - name, system 39
 - names, image files 29
 - naming conventions,
 - script files 10
 - new, information on 34
 - number of new 33
 - system, number of 38
 - tag 30
 - unprocessed 37
 - writing to 55
- flsc 24, 29, 34
- focus
 - zone specifications 32
- format of commands 2
- formatted date 28
- forward reference 10
- free storage 37
- full operation 35
- full size picture 30, 34

G

- GetCameraDefault 22
- GetCameraState 23
- GetCapabilitiesCount 25
- GetCapabilitiesListItem 26
- GetCapabilitiesRange 26
- GetCapabilitiesValue 27
- GetCapabilityType 27
- GetClock 27

- GetDateStr 28
- GetError 28
- GetFileCount 29
- GetFileInfo 29
- GetFileTag 30
- GetImageSpecifications 31
- GetMarkedImage 32
- GetMarkedImageCount 33
- GetNewFileCount 33
- GetNewFileInfo 34
- GetOption 35
- GetPowerMode 35
- GetProductInfo 36
- GetScriptName 36
- GetStorageStatus 37
- GetString 38
- GetSystemFileCount 38
- GetSystemFileName 39
- GetTimeString 39
- goto statements 10
- group
 - capture 19
 - capture sequence 47
 - name 51
 - sequence number 51
- guidelines for script statements 2

H

- horizontal tabs 2
- horizontal zones 32

I

- i data type 12
- identifiers 10
- image
 - captured, number of 29
 - current number of
 - marked 33
 - file name format 29
 - file processing completed
 - flag 24
 - file retrieval 32
 - information commands 15
 - information, hardware-related 30
 - marked 32
 - number of new 33
 - number, in file name 29
 - partial 30, 34

- processing chain 24
- processing complete 30, 34, 51
- processing in progress 24, 51
- input text string 38
- internal storage 30, 34
- internal storage, deleting files 20
- ipcm 30, 34
- ipip 24, 51

J

- Japanese Text 62
- jump to (goto) 10

L

- label marker definition 10
- linefeed 2
- low-power state 24

M

- macro
 - button 24
 - position 24
- MakeFolder 40
- marked image commands 17
- marked images 32
- mcro 24
- memf 24
- memory full flag 24
- memory, working 30, 34
- minimal capture 35
- min-max range 26

N

- n data type 12
- name of script 36
- name-value pair 26
- naming conventions 10
- new files 34
- NULL strings 13
- number of captured images 29

O

- open serial channel 44

- opening a file 21
- operators 13
 - arithmetic 13
 - assignment 13
 - bitwise 13
 - relational 13
- option
 - list 35
 - list commands 15
 - list, displaying to user 49
 - selecting 35

P

- parameter
 - capability type 27
 - capability value 27
 - name-value pair 26
 - settings, updating 47
 - value, restore defaults 42
- partial image 30, 34
- ped1 30, 34
- ped2 30, 34
- piav 30, 34
- picture type 30, 34
- piwm 30, 34
- power
 - alert 24
 - commands 15
 - level, camera 35
 - low 24
- powering off the camera 49
- prmd 30, 34
- product
 - commands 15
 - information 36
- programmed sequence, terminating 19
- ptyp 30, 34
- pwra 24

Q

- quotes 12

R

- RAM
 - disk 24
- ramda 24
- range, min-max 26
- raw image data 37
- read

- commands 16
- receive cycle 45
- receive from serial port 44
- reduced capture 35
- region code 28
- relational operators 9, 13
- removable media 37
- removable media, deleting files 20
- RestoreCameraDefault 40, 41, 42
- retrieving image files 32
- rgnc 28, 39
- rsvd 24
- RunApp 43
- RunScript 43

S

- s data type 12
- sbs1 24
- sbs2 24
- screenail specifications 32
- script
 - command summary 15
 - commands, error codes 13
 - file name conventions 10
 - file, running 43
 - language operators 13
 - name, retrieving 36
 - statement guidelines 2
 - wait for shutter button 55
 - waiting during 54
- Scripting Techniques
 - Capturing Images 58
 - Error Checking 57
 - Extracting bits from a bitfield 60
 - Japanese Text usage 62
 - Startup Scripts 59
 - Watermark placement 61
- select option 35
- self test
 - complete 24
 - in progress 24
- sendlist 7
- separators 2
- serial channel, opening 44
- serial port
 - closing 43
 - opening 44
 - receiving data 44

- sending data 45
- serial processing commands 16
 - SerialClose 43
 - SerialOpen 44
 - SerialReceive 44
 - SerialSend 45
 - SerialSendReceive 45
 - set clock value 47
 - SetCameraDefault 46
 - SetCameraState 47
 - SetCaptureMode 47
 - SetClock 48
 - SetOption 48, 49
 - SetPowerMode 49
 - setting user tag 50
 - SetUserFileTag 50
- shutter
 - button 55
 - button status 24
 - wait for 55
- StartCapture 51
 - when not accepted 24
- Startup Scripts 59
- state commands 15
- status commands 15
- still
 - capture sequence 47
- storage status 37
- storage, internal 30, 34
- strc 25
- string
 - commands 17
- string assignment 5
- string commands 17
- strobe
 - charging flag 25
 - ready flag 25
- strr 25
- stsc 24
- stst 24
- system file name 39
- system files, number of 38
- SYSTEM folder 38, 39
- SystemStatus 23, 24

T

- tabs, horizontal 2
- terminate a programmed sequence 19
- text, input from user 38
- tfmt 39

- thumbnail 30, 34
 - specifications 31
- time
 - burst timer flag 24
 - format, by region D-219
 - get current 27
 - get formatted 39
 - setting 47
- time lapse
 - capture mode, side-effects 52
 - capture sequence 47
 - timer flag 24
- trun 24
- truncation 13

U

- u data type 12
- uninitialized variables 13
- unprocessed files 37
- user
 - default values 46
 - defaults 22, 42
 - input text 38
 - tag, setting 50

V

- variables, uninitialized 13
- VendorStatus 23, 25
- vertical zones 32

W

- Wait 54
- wait commands 17
- WaitForShutter 55
- working memory 30, 34
- Write 55
- write
 - commands 16
- WriteLine 56

Z

- zone
 - horizontal 32
 - organization 30
 - specification, exposure 32
 - specification, focus 32
 - vertical 32