DATA COMMUNICATION STANDARD
VERSION 3.10

OCTOBER 2014

DEVELOPED BY:

Data Communication Standard Committee
Lens Processing Division of The Vision Council

The Vision Council
225 Reinekers Lane Suite 700
Alexandria, VA 22314
thevisioncouncil.org

THE VISION COUNCIL

Contents

Page

Foreword

The Lens Processing Technology Division of the Vision Council (“VC”) developed Version 3.09 of the Standard for Data Communications. It supersedes all prior versions. The version is identified within the protocol by the data record OMAV=3.09. Version 3.03 is substantially identical to ISO standard 16284 that is based on this VC work

Introduction

This Standard is the result of a desire shared by manufacturers of optical laboratory equipment and producers of software used in optical laboratories to simplify the interconnection of their products.

The Standard defined herein provides:

- a method by which machines and computer systems conduct their exchanges of data;
- a method by which computer systems can initialize such parameters on machines as the manufacturers thereof allow;
- a method by which machines can initialize computer systems with information that the systems can use for various purposes;
- a method by which a machine can inform a computer system as to what information it wants to receive, thus allowing machines to define new interfaces dynamically.
- a standard set of records and Device types that are used to communicate agreed upon sets of information.

The last feature listed above requires that the Standard be amended on a regular basis, as the need for new data elements is inevitable.

Ophthalmic optics — Information interchange for ophthalmic optical equipment

1 Scope

This Standard establishes a method by which machines and computer software systems used in the fabrication of ophthalmic lenses can exchange information.

2 Normative Reference

The following normative documents contain provisions that, through reference in this text, constitute provisions of this Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document applies.

ISO 13666:1998, *Ophthalmic Optics – Spectacle Lenses – Vocabulary*.

3 Terms and definitions

For the purposes of this Standard, the terms and definitions given in ISO 13666:1998 and the following apply:

3.1 General

3.1.1
device
machine or instrument used in the fabrication of ophthalmic lenses that communicates with a computer system to send or receive job information

3.1.2
host
computer system providing information to or receiving information from a device

3.1.3
job
order for prescription ophthalmic lenses or spectacles

3.1.4
download
communication session in which the host system transmits data to the device

3.1.5
upload
communication session in which the device transmits data to the host

3.2 Reserved characters

3.2.1
code separator
reserved character used to delimit codes in a device record

3.2.2
CRC position character
reserved character marking the location of the end of the data records and the start of the optional CRC record within a packet

3.2.3
end character
reserved character marking the end of a packet

3.2.4
field separator
reserved character delimiting the fields in a record

3.2.5
label separator
reserved character separating the record label from the field(s) within a record

3.2.6
mandatory record flag
reserved character marking certain records as mandatory (deprecated)

3.2.7
start character
reserved character marking the beginning of a packet

3.2.8
record separators
reserved characters which delimit records

3.2.9
unknown data indicator
reserved character indicating that data required for a particular field is unknown to the host

3.2.10
ACK character
reserved character indicating successful transmission of a packet

3.2.11
NAK character
reserved character indicating failed transmission of a packet

3.2.12
control character
character having an ASCII value of less than 32

3.2.13
sub-field separator
reserved characters which delimit sub-fields

3.3 Data types

3.3.1
limited data
text data limited to a maximum length

3.3.2
literal data
text data limited to a maximum length and specified in the standard

3.3.3
numeric data
floating-point and integer numbers

3.3.4
text data
strings of characters that have no pre-defined meaning

3.3.5
integer data
data represented in whole number form, limited to the range -32768..32767.

3.3.6
binary data
data presented in a form usable by computer software with little or no translation.

NOTE Binary data requires special handling to avoid introduction of control characters

3.4 Messages

3.4.1
message
structured stream of data transmitted from a host to a device or from a device to a host

3.4.2
confirmation message
message sent by the receiver of a packet and comprised of a single character indicating that the transmission was successful

3.4.3
positive acknowledgement
single character message indicating successful reception of a sender's message

3.4.4
negative acknowledgement
single character message indicating unsuccessful reception of a sender's message

3.4.5
packet
structured message consisting of a start character and a series of records and terminated by an end character

3.4.5.1
data packet
packet sent from a device to a host, or from a host to a device, containing requested information

3.4.5.2
request packet
packet sent from a device to a host to initiate a session

3.4.5.3
response packet
packet containing status information

3.5 Records

3.5.1
record
structured stream of characters including a record label, a label separator, zero or more data fields separated by field separators which may contain multiple values separated by sub-field separators, and a terminating record separator

3.5.2
data field
single data element within a record

3.5.3
record label
a string of characters that identifies data contained in a record

NOTE A list of record labels is in Annex A.

3.5.4
ASCII record
record comprising ASCII characters and conforming to the structures defined herein

3.5.5
binary record
record comprising bytes encoded using the binary number system

3.5.6
chiral record
record with two fields, one for a data element for a right lens or eye, and one for a left, arranged in the order right, then left

3.5.7
CRC record
record at the end of any packet containing a CCITT CRC-16 cyclical redundancy check value calculated on the characters transmitted

3.5.8
device record
record containing job specific data elements conveyed between devices and hosts

3.5.9
interface record
record supporting the operation of the host-device interface and not containing job-specific data

3.5.10
process-control record
record controlling the operation of a device

3.5.11
structured datasets
groups of records which must appear in a specified order

3.6 Sessions

3.6.1
session
sequence of messages passed between a device and a host that serves to exchange information related to a single order or task

3.6.2
initialization session
specialized session allowing devices to provide hosts with information that would otherwise be included with each request, such as machine model, software version and operator ID

3.6.2.1
auto-format initialization
initialization session allowing devices to define sets of device records to be requested from hosts

3.6.2.2
preset initialization
initialization session allowing devices to transmit sets of identifying data to hosts

3.6.3
download session
session in which information is passed from a host to a device

3.6.4
upload session
session in which information is passed from a device to a host.

3.6.5
INFO session
upload request packet containing job status information used to indicate the completion of a job by a device.

3.6.6
MNT session
upload request packet containing vendor specific device information

3.7 Timeout

3.7.1
timeout
numeric value representing that period of time that a host or device shall wait for the arrival of data, after which it assumes that such data shall not be forthcoming

3.7.1.1
confirmation timeout
timeout which applies to the reception of the confirmation message

3.7.1.2
intercharacter timeout
timeout which applies to the interval between successive characters in a stream of data

3.7.1.3
packet timeout
timeout which applies to the reception of a packet

4 Overview

The strategy used in this Standard for the exchange of data between devices and hosts can be expressed as follows:

A machine used in the fabrication of ophthalmic lenses (a device) sends a request to a computer system (a host), indicating a need to do one of the following:

— Initialize information to identify the device, software versions, model numbers, etc.

— Upload to the host, information for it to store and/or use in the processing of ophthalmic prescription orders;

— Download from the host, information required by the device for it to perform its tasks.

Communication can be "initialized" in two ways. The device may begin an initialization session or the host can force the device to do so by refusing to accept a normal request and asking for initialization via a special error response. For upload requests, the host acknowledges the request and the device sends its data, the receipt of which the host acknowledges. For download requests, the host responds to the request with the data requested.

The variable-length packets of data that comprise this exchange consist of a series of records, each of which contains data and a label identifying the data. The Standard defines a set of labels and characterizes the data associated with each. This set of labels has been expanded on many occasions since its first publication, and shall continue to be expanded as needed in the future.

An exchange of packets related to a single job is called a session. The structure of these sessions and the packets of records of which they are composed is the substance of this Standard.

This Standard was initially implemented using point-to-point RS-232 serial links. In Version 3.04, specifications were included to facilitate communications on TCP/IP networks.

5 Requirements

NOTE In the examples in this document, in the interests of readability, the RECORD SEPARATORS may be omitted, the START CHARACTER may be placed on a separate line, and CRC RECORDS may be excluded. Remarks have been included as REM records. Comments are enclosed in parentheses ("**(…)** ") and are not part of the data stream. Ellipses (" … ") are used to indicate more data of the same type as precedes and follows the ellipses. Square brackets indicate data that is optional or which is record-dependent. SPACES have been inserted around record and field separators for readability; in practice these should not be included in packets as this needlessly decreases the efficiency of expression. In the descriptions below, REQUEST, RESPONSE, and DATA refer to packets.

5.1 Records

All records have the following form:

<record label><label separator><field value>[<sub-field separator><field value>][<field separator><field value> …]<record separator[s]>

The label separator is invariably the equals sign ("=") and the field separator is invariably the semi-colon (";"). Records contain one or more field values (as indicated by the square brackets in the example above); the number of permissible or required field values for a given record label is specified in Table A.1. Because of the chiral nature of spectacles (that is, “handed”, having right and left component parts), many records are specified to be “chiral”, which means that two field values shall nominally appear, the first applying to the right lens and the second to the left.

Fields may contain multiple values separated by a sub-field separator, which is invariably the pipe symbol (“l”). Records in which sub-fields are allowed are speicifed in Table A.1.

In every case, only as many field separators as are necessary to delineate included fields are required; that is, after the last included field in any record, no additional field separators are required. However, if additional field separators are included, they should be tolerated.

Parsers should strive to accommodate reasonable variations to the basic record format by tolerating a single value where chiral values are specified, applying the value received to both eyes, or to the eye being processed; similarly, the presence of a superfluous right-eye value on a left-eye-only order should be tolerated.

EXAMPLE : A chiral record:

SPH=2.50;2.75

5.1.1 Interface records

The Standard defines a set of interface records. These records contain information which the host and device use to communicate. They do not contain job specific data. These records are enumerated in A.2.

5.1.2 Device records

The Standard defines a set of device records which identify the data elements that might be required by any of the devices that might be required for the fabrication of a job. These records are enumerated in A.1.

5.1.3 Preset device types

The Standard further identifies subsets of device records that are deemed to be appropriate for specific types of devices. These records are enumerated in A.3. These sets of records are used only when auto-initialization is not performed, and are provided for the limited mode of operation available in the absence of initialization.

5.1.4 Records with unknown values

If the host is requested to send any record for which it has no information or partial information, it shall send the record with a question mark "?" in all the unknown data fields in order to indicate that the information is not available. Such records must be properly formatted according to the rules for chiral records.

NOTE Hosts or devices that do not support a particular record will not be aware of the correct structure of such record, therefore, the form “LABEL=?”, where “LABEL” is a record label unknown to the provider, is always a correct form.

5.1.5 Ignored records

Whenever a host or a device receives a record whose label it does not recognize, it shall ignore the record. This does not apply to records that appear in record label lists as described in 6.2.4.

5.1.6 Experimental records

When a machine vendor wishes to test new records prior to submitting them for inclusion in the Standard, such records should use labels that begin with an underscore character (ASCII "**_**", decimal 95). Excessive label length should be avoided, and labels may not include spaces or reserved characters defined in this standard.

5.1.7 Reserved characters

5.1.7.1 Control characters and the additional characters specified may not appear in transmitted data streams except as specified. The set of reserved characters is specified in Table 1.

5.1.7.2 Reserved characters shall appear in ASCII records only to provide the functionality they are assigned, as in the case of record and field separators. Reserved characters that conform to the definition of text data may also appear in text fields.

5.1.7.3 When a reserved character with a decimal value less than 32 appears in a binary record, it shall be "escaped" in the following manner. In place of such a character, two characters shall be sent. The first character shall be an ESC character followed by the original character with its high bit set, i.e., the character is OR'd with decimal 128, hex 0x80. The receiver, on receipt of an ESC character, shall discard the ESC character and clear the high bit of the following character. The CRC value, if present, shall be determined after such reserved characters are escaped, so that a receiver need not process packets prior to validating a received packet's CRC.

NOTE In other words, the transmitter encodes control characters before calculating the CRC, and the receiver calculates the CRC before decoding them.

EXAMPLE : A stream of bytes (a short tracing record in absolute binary form) before and after having been "escaped" as described above.

```
Before:

R=175   9  23  10  45  10 223   9  90   9 205   8  89   8 252   7 183   7 143   7
130   7 147   7 197   7  24   8 136   8  18   9 167   9  39  10  85  10  19  10
213   9 146   9  75   9  14   9 199   8 120   8  38   8 222   7 166   7 131   7
117   7 122   7 149   7 191   7 241   7  41   8  92   8 152   8 229   8  67   9 <CR/LF>

After:

R=175   9  23 27 138 45 27 138 223 9 90 9 205 8 89   8 252   7 183   7 143   7
130   7 147  7  197   7  24   8 136   8  18   9 167   9  39 27 138 85 27 138 27 147 27 138
213   9 146  9   75   9  14   9 199   8 120   8  38   8 222   7 166   7 131   7
117   7 122  7  149   7  191 7 241   7  41   8  92   8 152   8 229   8  67   9 <CR/LF>
```

5.1.7.4 Text data consists of character strings encoded in a standard encoding scheme. Standard encoding schemes are listed in Table A.2 in order of complexity. Support for the ASCII scheme is mandatory; support for any other standard scheme is optional. Support for an optional encoding scheme is negotiated during initialization, or, in the absence of initialization, during each session, by means of the TXTENC record. A device may include a TXTENC record in such packets, the data field of which comprises an ordered list of literals specifying the schemes it supports, separated by semi-colons. A host that supports optional encoding schemes shall include a TXTENC record in its response, indicating the most complex scheme it supports, which shall be the scheme used in sessions between that host and device. The absence of TXTENC or TXTENC=? in a request or response, indicates support for the ASCII encoding scheme only.

Limited and Literal data consist of ASCII character strings, further described below.

ASCII characters are constrained to the range of 32 to 127 decimal excluding 59 (semi-colon) in all data types.

In encoding schemes other than ASCII, the number of bytes of which a string is composed may exceed the number of characters expressed thereby; the length limitations below refer to the number of characters, not the number of bytes.

5.1.7.5 Text data is a character string expressed in a standard encoding scheme having no predefined meaning. Length is limited to 80 characters.

5.1.7.6 Limited data is a string of ASCII characters in the range of 32 to 127 decimal excluding 59 (semi-colon) constrained to a length of 12 characters.

5.1.7.7 Literal data is a string of ASCII characters in the range of 32 through 127 whose meaning is implied by the record type and specified in this standard. Length is limited to 12 characters unless otherwise noted in the record definition. Literal data shall not contain reserved characters defined by the interface (See Table 1). Literal data is case sensitive.

5.1.8 Record length

Non-binary records should not exceed 80 characters in length. As noted above, text expressed in encoding schemes other than ASCII may require more than one byte per character. Binary records may be longer than 80 characters.

5.1.9 Structured datasets

Generally, records are independent, and may appear in any sequence, except as specified herein. Certain kinds of data, however, are expressed in several different records, which must appear in the order specified. Tracing datasets (see section 5.4) and surface definition datasets (section 5.6.4) are examples of data expressed in datasets.

Table 1 – Reserved Characters

Character	Hexadecimal Value	Decimal Value	Control Key	Use
FS	0x1C	28	^\	Start of message
GS	0x1D	29	^]	End of message
DC1	0x11	17	^Q	Reserved (XOFF)
DC3	0x13	19	^S	Reserved (XON)
ACK	0x06	06	^F	Positive acknowledgement
NAK	0x15	21	^U	Negative acknowledgement
ESC	0x1B	27	^[	Escape
RS	0x1E	30	^^	CRC separator
SUB	0x1A	26	^Z	DOS End-of-file marker
CR	0x0D	13	^M	Record separator
LF	0x0A	10	^J	Record separator
;	0x3B	59	;	Field separator
=	0x3D	61	=	Label separator
,	0x2C	44	,	Code separator
*	0x2A	42	*	Mandatory record flag
?	0x3F	63	?	Unknown data indicator

5.2 Reference point records

Records are defined to indicate the horizontal and vertical distances between two reference points or to indicate an action a machine should take relative to a reference point. The following naming scheme will clarify all such reference records included in this Standard and can easily be extended for future ones.

5.2.1 The first two letters of the record label describe the first reference point, the second two letters of the record label describe the second reference point, and the last two letters indicate “IN” (horizontal) or “UP” (vertical) directions. The values indicate the position of the second reference point with respect to the first reference point.

5.2.2 A positive IN value indicates that the second reference point is towards the nasal relative to the first.

5.2.3 A negative IN value indicates that the second reference point is towards the temporal relative to the first.

5.2.4 A positive UP value indicates that the second reference point is above the first.

5.2.5 A negative UP value indicates that the second reference point is below the first.

Table 2 - Reference point identifiers

Identifier	Reference Point
BC	Lens Blank Center
ER	Engraving Reference Point; midpoint between semi-visible alignment marks
FB	Finish Block Location
FC	Frame Box Center
OC	Prism Reference Point (as defined in the VCA Lens Description Standard version 1.1)
SB	Surface Block Location
SG	Layout Reference Point (as defined in the VCA Lens Description Standard version 1.1)

Table 3 - Reference point records

Label	Meaning
FBFCIN, FBFCUP	Finish Block to Frame Center (see Table A.1 for usage)
FBSGIN, FBSGUP	Finish Block to Layout Reference Point (Segment/Fitting Cross/PRP)
FBOCIN, FBOCUP	Finish Block to Prism Reference Point (Optical Center)
SBBCIN, SBBCUP	Surface Block to Blank Center
BCSGIN, BCSGUP	Blank Center to Layout Reference Point
BCOCIN, BCOCUP	Blank Center to Prism Reference Point
SBSGIN, SBSGUP	Surface Block to Layout Reference Point
SBOCIN, SBOCUP	Surface Block to Prism Reference Point (see section 5.3.11 for use)
SBFCIN, SBFCUP	Surface Block to Frame Center
SGOCIN, SGOCUP	Layout Reference Point to Prism Reference Point (commonly known as “inset” and “B.O.C.”)
FCSGIN, FCSGUP	Frame center to Layout Reference Point (commonly known as “total inset” and “segment drop”)
FCOCIN, FCOCUP	Frame center to Prism Reference Point (commonly known as “decentration” and “O.C. drop”)

5.3 Generator records

5.3.1 The surface generator interface includes a number of records used to indicate adjustments that should be applied to the generator machine settings. Because the Standard provides for a complete dataset ("preset packet") to be sent to an "unknown" generator, it is necessary to clarify some of the relationships amongst these records, especially as relates to the "compensation" fields.

The position of a lens in a generator can be determined by the RNGH, RNGD, SAGRD, and SAGBD fields (ring height, ring diameter, lens sag at ring diameter, and lens sag at blank diameter, respectively). Some generators, especially those with exclusively mechanical components, may presume certain values for some of the above records and may be unable to effect the adjustments required by the mismatch in assumptions. The following compensation fields provide the data required to make these adjustments.

5.3.2 BLKCMP represents the change that is required to be made to the generator thickness setting that arises from a mismatch between the curvature of a block that has a curved contact surface with the lens and the curvature of the lens blocked thereon. The BLKB field contains the curvature of that surface of the block that contacts the lens; the BLKD field contains the diameter of the block.

5.3.3 When the blocks used for a job do not have a curved contact surface, the BLKB field is not necessary. If it is sent in such a case, its value should be equivalent to IFRNT.

5.3.4 RNGCMP represents the change that is required to be made to the generator thickness setting that arises from a mismatch between the blocking ring height and/or diameter, known to the host, and that which is presumed by the generator.

5.3.5 FINCMP indicates the amount of thickness that should be added to the generator setting to allow for material removed during fining (smoothing) and polishing.

5.3.6 THKCMP shall be used for such compensations to thickness as may be required, which are not otherwise handled by the records defined herein.

NOTE The above-enumerated thickness compensation fields apply equally to GTHK and OTHK.

5.3.7 EECMP indicates a diopter amount that shall be added to the GCROS and GCROSX values when elliptical error compensation is required.

5.3.8 A host can maintain complete control over the settings on a generator by including all of the compensation values that it believes to be required by a particular machine in the basic setting records and sending zero values in the compensation records; e.g., sum GCROS with EECMP, send the result as GCROS, and send EECMP with values equal to zero.

5.3.9 In compensation records, **zero values** indicate that the host does not want the machine to apply any compensation that it may be capable of being set to do.

5.3.10 The **ABSENCE of compensation fields** indicates that the generator should apply such compensations as it may be set to do.

5.3.11 Prism v. Decentration: the fields GPRVM and GPRVA represent prism magnitude and direction to be generated. These values should include all possible contributions to prism, including Rx prism, thinning prism, and prism for decentration. The fields SBOCIN and SBOCUP express vector distances to which the grinding center is to be offset from the surface block center, laterally and vertically. In order that both be expressed in a single packet, as would be desirable in the case of a "generic" GEN request, a set of prism records is defined, RPRVM and RPRVA, which represents prism to be ground exclusive of the decentration expressed in SBOCIN/SBOCUP. In addition, when decentration is expressed directly (via SBOCIN/SBOCUP), the OTHK value should be present. The following rules describe how these fields should be aggregated.

5.3.11.1 When expressing decentration to be generated as prism, the following fields are used together: GPRVM, GPRVA, and GTHK.

5.3.11.2 When expressing decentration to be generated directly, the following fields are used together: RPRVM, RPRVA, SBOCIN, SBOCUP, and OTHK. RPRVM differs from PRVM in that the former is expressed in PIND diopters or degrees, while the latter is expressed in LIND or “natural” diopters.

5.3.11.3 Because the method described in 5.3.11.1, in which decentration is expressed in the form of prism, is the more universal one for expressing decentration to be ground, the set of records described therein should be supported by all generators and hosts. The set of records described in 5.3.11.2, in which decentration is expressed as vectors, should be included in addition to the 5.3.11.1 set. Generators not supporting the vector method can safely ignore the vector fields.

5.3.12 Pad Thickness: The inclusion of the PADTHK record indicates that LAP curves cut by a generator should be compensated for the thickness of the pad. No compensation should be made to the lens curves expressed in GBASE/GCROS or GBASEX/GCROSX. It is the responsibility of the host to determine such compensations as shall be applied to generator curves, the need for which may result from the use of laps not compensated for pad thickness, or the desire to fine lenses from edge to center.

5.3.13 Curve Signs: Concave curves shall be expressed as negative numbers and convex curves as positive numbers. One implication of this is that for generators that cut laps, the lens curves and lap curves shall have opposite signs.

5.4 Tracing datasets

5.4.1 Expression of trace data

Trace data begins with a TRCFMT record that specifies the format in which the tracing is to be expressed, the number of points to be transmitted, whether the radii are equiangular, the orientation of the tracing and an indication of what was traced.

A device indicates its desire to upload or download trace data by including one or more TRCFMT records in either its initialization packet, or, if no initialization is done, its request packet. All supported formats and numbers of points are listed in the order of the device's preference. It is not necessary to include every combination of side specifiers (B, R, and L); because of the requirement specified in 5.4.13, only the device's preferred mode need be specified. Devices that neither upload nor download trace data do not include any TRCFMT records in their initialization or request packets. The same rules apply to sag data and ZFMT.

When trace data is expected by a device, but unavailable from the host, a special form of the TRCFMT record is sent, composed of a single field with the integer value 0. When trace data is available, but sag data is not, a similarly-formed ZFMT record (ZFMT=0) is sent by the host. When TRCFMT=0 is sent, ZFMT=0 is implied and not required. When two sides of radius data are sent, for which sag data are unavailable, a ZFMT=0 record should follow each set of radius data. In the case that two-side data has been negotiated, and only a single side is sent, TRCFMT=0 is not sent for the missing side.

EXAMPLE: Trace data requested by device, but unavailable from host:

```
TRCFMT=0
```

EXAMPLE: Trace data requested by device and available from host, Z data requested, but unavailable:

```
TRCFMT=1;400;E;R;F<CR/LF>
R=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
<etc.>
```

```
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
ZFMT=0
```

5.4.2 Radius data

These are contained in "R" records. Sag data is contained in "Z" records. Angle data for radius data is contained in "A" records, and for sag data in ZA records. For the ASCII format, all of the records follow the 80-character line limit rule, therefore there may (in fact, will likely) be multiple R, A, Z, and ZA records for each tracing. For BINARY data, the line limit rule will not apply and there will be only one R, A, Z and ZA record as needed. Radius and sag data is expressed hundredths of a degree with an implied decimal point.

EXAMPLE Radius data :

a) ASCII format

```
TRCFMT=1;400;E;R;F<CR/LF>
R=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
etc…
```

b) Binary formats :

```
TRCFMT=2;400;E;R;F<CR/LF>
R=<binary data stream><CR/LF>
```

5.4.3 Uneven angles

When these are specified in the TRCFMT record, angle data is required. It shall appear immediately after its corresponding radius data. Angle data is contained in one or more "A" records, and shall be expressed in the same format as the radius data to which it corresponds. Angle data is expressed in hundredths of a degree with an implied decimal point and shall be in the range 0-35999.

NOTE Angle data may also follow the sag data if uneven angles are specified for the sag data.

5.4.3.1 Two Types of Uneven Angles

Beginning with version 3.09, two distinct types of uneven (non-equiangular) tracing datasets are defined. To date, uneven angles have been specified by a “U” in the third field of the TRCFMT record. Although the angles specified for such tracings are unevenly spaced, they are expected to be incremental; that is, each angle is expected to refer to a position counter-clockwise from its predecessor. In order to allow more complex shapes to be produced, an additional angle mode is defined, “C” (for “creative”), in which case, angles may increase or decrease. By this means, shapes that incorporate included openings (such as a keyhole) can be specified. However, such datasets cannot be converted to equiangular datasets.

5.4.4 ZFMT

The sag data format record, ZFMT, has exactly the same definition as the TRCFMT record. The same rules apply to angle data for sag data as to angle data for radius data.

5.4.5 "R" records

These shall appear immediately after the TRCFMT record. All of the "R" records for a tracing (i.e., a single side, when two are provided) appear together. Each side has its own TRCFMT record.

5.4.6 "Z" records

These shall appear immediately after the ZFMT record. All of the "Z" records for a tracing (i.e., a single side, when two are provided) appear together. Each side has its own ZFMT record.

5.4.7 "A" or “ZA” records

These, when present, shall appear immediately after their corresponding set of "R" or "Z" records. ZFMT and "Z " records shall appear immediately after their corresponding TRCFMT and "R" records.

EXAMPLE A two-eye tracing dataset (abbreviated) showing the required sequence of records

```
TRCFMT=1;400;U;R;F<CR/LF>
R=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
…
R=1909;1914;1941;1983;2033;2089;2140;2200;2277;2371<CR/LF>
A=0;90;180;270;360;450;540;630;720;810<CR/LF>
A=900;990;1080;1170;1260;1350;1440;1530;1620;1710<CR/LF>
…
A=35100;35190;35280;35370;35460;35550;35640;35730;35820;35910<CR/LF>
ZFMT=1;100;U;R;F<CR/LF>
Z=322;331;342;328;314;308;300;295;288;280<CR/LF>
…
Z=316;318;324;328;333;343;349;352;357;362<CR/LF>
ZA=0;360;720;1080;1440;1800;2160;2520;2880;3240<CR/LF>
…
ZA=32400;32760;33120;33480;33840;34200;34560;34920;35280;35640<CR/LF>
TRCFMT=1;400;U;L;F<CR/LF>
R=2517;2450;2379;2318;2247;2168;2086;2014;1958;1923<CR/LF>
R=1909;1914;1941;1983;2033;2089;2140;2200;2277;2371<CR/LF>
…
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
A=0;90;180;270;360;450;540;630;720;810<CR/LF>
A=900;990;1080;1170;1260;1350;1440;1530;1620;1710<CR/LF>
…
A=35100;35190;35280;35370;35460;35550;35640;35730;35820;35910<CR/LF>
ZFMT=1;100;U;L;F<CR/LF>
Z=322;331;342;328;314;308;300;295;288;280<CR/LF>
…
Z=316;318;324;328;333;343;349;352;357;362<CR/LF>
ZA=0;360;720;1080;1440;1800;2160;2520;2880;3240<CR/LF>
…
ZA=32400;32760;33120;33480;33840;34200;34560;34920;35280;35640<CR/LF>
```

5.4.8 Centration of data

Radius data shall be centered geometrically when presented to the host. The host may decenter the shape in order to produce a certain effect on a device.

5.4.9 Lens sizing

5.4.9.1 BSIZ and CSIZ records indicate that devices shall modify the dimensions of the received trace by the amount specified.

5.4.9.2 Non-zero values shall not be supplied for both BSIZ and CSIZ.

5.4.9.3 HBOX, VBOX and CIRC records shall reflect the dimensions of the shape sent in R records.

5.4.9.4 In the particular case where BSIZ and CSIZ are absent or unknown and a single side tracing and two CIRC or CIRC3D values which differ are received, the CIRC or CIRC3D for the eye sent should reflect the shape. CSIZ is implied to be the difference between the two circumferences and the shape for the eye not sent shall be modified according to the implied CSIZ.

5.4.9.5 When CIRC3D is present, either z-data, or a FCRV value, shall be included. If z-data is included, the value of CIRC3D shall be calculated using the z-data; if only FCRV is included, the value of CIRC3D shall be calculated based on the FCRV value.

5.4.10 Rotational orientation

These shall be expressed so that the first radius is at zero degrees (3 o'clock on standard polar scale) and shall proceed anti-clockwise. When angle data are provided, the starting radius should be the first available meridian greater than or equal to zero.

5.4.11 Eye orientation

5.4.11.1 Eye orientation, right or left, should be viewed as a refractionist views a spectacle wearer; right-eye oriented data therefore start at the nasal side while left-eye data start at the temporal.

5.4.11.2 Eye orientation may be established during initialization. If it is not, it is specified in the device's request packet.

5.4.11.3 When eye R (Right) is specified, the device wants to send or receive a single set of trace data with right-eye orientation.

5.4.11.4 When eye L (Left) is specified, the device wants to send or receive a single set of trace data with left-eye orientation.

5.4.11.5 When eye B (Both) is specified, the device wants to send or receive both right and left sets of tracing data, each in the appropriate orientation.

5.4.12 Eye orientation during tracing transmission

5.4.12.1 When eye R (Right) or is specified in the TRCFMT or ZFMT records, the tracing will have RIGHT eye orientation.

5.4.12.2 When eye L (Left) is specified in the TRCFMT or ZFMT records, the tracing will have LEFT eye orientation.

5.4.12.3 B shall be specified only in request or initialization packets, never in data packets. In data packets in which both sides are included, there shall be two TRCFMT records, one for each side, labeled appropriately.

5.4.13 Trace orientation acceptance

Hosts and devices shall handle the reception of either orientation of tracing data without generating an error condition. Hosts shall make an effort to provide data in the quantity and orientation requested by the device when possible.

5.4.14 Sag Data

In the expression of sag data, the smallest number in the dataset shall be positive and shall represent the point located furthest toward the front of the frame or lens. The distances from this point to other points in the dataset have positive values in increments of 0.01 mm.

5.4.15 Tracing formats

Example Data

NOTE 1 It is strongly recommended that devices be consistent in the representations of data. While it is not expressly forbidden, representing radius data in one orientation and sag data in another might violate host system programmers' unwarranted assumptions and fail.

NOTE 2 The following small sample tracing, comprised of 40 radii, will be used for the examples below. The sample is not formatted in any particular way. It is just a list of the radius values used in the examples that follow.

24.79, 25.83, 26.05, 25.27, 23.94, 22.53, 21.37, 20.44, 19.75, 19.35,

19.22, 19.39, 19.89, 20.72, 21.84, 23.22, 24.71, 25.99, 26.45, 25.79,

25.17, 24.50, 23.79, 23.18, 22.47, 21.68, 20.86, 20.14, 19.58, 19.23,

19.09, 19.14, 19.41, 19.83, 20.33, 20.89, 21.40, 22.00, 22.77, 23.71

5.4.15.1 ASCII absolute format

In this format, each radius is presented as a 4-digit decimal number in hundredths of a millimeter with an implied decimal point. Each value is separated by a field separator (semi-colon). Data shall follow the 80-character line limit rule, therefore multiple "R" records are required for a single radius dataset.

EXAMPLE A tracing expressed in format #1, ASCII absolute, requires 196 bytes of radius data (including the semi-colons).

```
TRCFMT = 1;40;E;R;F<CR/LF>
R=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
R=2517;2450;2379;2318;2247;2168;2086;2014;1958;1923<CR/LF>
R=1909;1914;1941;1983;2033;2089;2140;2200;2277;2371<CR/LF>
```

5.4.15.2 Binary absolute format

In this format, each radius is expressed as a 16-bit integer with the radius in hundredths of a millimeter. The entire dataset is contained within a single radius data record.

NOTE Each binary 8-bit byte in the following examples is represented by 1 or more decimal digits separated from the next byte by a space. Semi-colons are used to separate the individual radii. **The semi-colons and spaces are not part of the actual binary record.**

EXAMPLE A tracing expressed in format #2, binary absolute, requires 86 bytes including the escape characters. The first example shows the data prior to the escape process (described in 5.1.7.3) having been applied, the second, afterwards. The escape sequences are shown in **bold**.

```
TRCFMT = 2;40;E;R;F
R=175   9; 23  10; 45  10;223   9; 90   9;205   8; 89   8;
  252   7;183   7;143   7;130   7;147   7;197   7; 24   8;
  136   8; 18   9;167   9; 39  10; 85  10; 19  10;213   9;
  146   9; 75   9; 14   9;199   8;120   8; 38   8;222   7;
  166   7; 131  7;117   7;122   7;149   7;191   7;241   7;
   41   8;  92  8;152   8;229   8; 67   9<CR/LF>

TRCFMT = 2;40;E;R;F
R=175   9;23 27 138;45 27 138;223   9; 90   9;205   8 ; 89   8;
```

```
252   7;183   7;143   7;130   7;147   7;197   7; 24   8;
136   8; 18   9;167   9; 39 27 138; 85 27 138;27 147 27 138;213  9;
146   9; 75   9; 14   9;199   8;120   8; 38   8;222   7;
166   7;131   7 117   7;122   7;149   7;191   7;241   7;
41    8; 92   8;152   8;229   8; 67   9<CR/LF>
```

5.4.15.3 Binary differential format

In this format, the starting radius is represented as a two-byte integer, as in format 2. Each radius thereafter is represented in a single signed byte as the difference between the current radius and the previous one. The data value expressed is therefore equal to the previous radius subtracted from the current. If the differential cannot be represented by signed byte, whose value must be between -127 and +127, a special value (hexadecimal 0x80, -128 decimal) is used to indicate that the next radius is expressed in absolute (i.e., 16-bit) form. The radius immediately following the 16-bit value is then represented in differential form.

EXAMPLE : A tracing expressed in format #3, binary differential, requires 54 bytes including the escape characters. As above, the first example shows the data prior to the escape process (described in 5.1.7.3) having been applied, the second, afterwards. In the first example, the absolute radii flags are shown in **bold**. In the second, the escape sequences are shown in **bold**.

```
TRCFMT = 3;40;E;R;F
R=175   9; 104; 22; -78;-128  90  9;-128 205 8;-116; -93;
  -69; -40; -13;  17;  50;  83; 112;-128 18 9;-128 167 9;
 -128  39  10;  46; -66; -62; -67; -71; -61; -71; -79; -82;
  -72; -56; -35; -14;  5; 27; 42; 50; 56; 51; 60; 77; 94<CR/LF>

TRCFMT = 3;40;E;R;F
R=175   9; 104; 22; -78 ;-128 90  9;-128 205 8;-116; -93;
  -69; -40; -13; 27 145; 50; 83; 112;-128 18 9;-128 167 9;
 -128  39 27 138; 46; -66; -62; -67; -71; -61; -71; -79; -82;
  -72; -56; -35; -14;  5; 27 155; 42; 50; 56; 51; 60; 77; 94<CR/LF>
```

5.4.15.4 Packed binary format

This is the most efficient, and most complex, method for the expression of shape information. To facilitate implementation of this format, a complete example is included in Annex B.

Three data types are defined :

- absolute, in which sixteen-bit "words" are used to express values in the range -327.67 mm to +327.67 mm ;
- differential, in which eight-bit "bytes" are used to express values in the range -1.26 mm to +1.27 mm ;
- incremental, in which four-bits "nibbles" are used to express values in the range –0.07mm to +0.07mm.

The first radius in a tracing is always expressed using the absolute data type. Radii subsequent to the first may be expressed as any of the three forms. Special values are reserved for each of the three types, which are inserted into the data stream to indicate that subsequent radii will be expressed in a different data type. These are shown in Table 4.

Table 4 – Packed binary type-shifting flag characters

Data Type	Hexadecimal Value	Decimal Value	Action	Label
Word	0x8000	-32768	Shift from absolute to differential form	AD
Byte	0x80	-128	Shift from differential to incremental form	DI
Byte	0x81	-127	Shift from differential to absolute form	DA
Nibble	0x8	-8	Shift from incremental to differential form	ID

Trace radii can always be expressed by the absolute data type, in which case the magnitude of the radius is simply encoded in a sixteen-bit word value in units of 0.01 mm, thus requiring sixteen bits per radius. Were this done for an entire record, the format would be indistinguishable from the binary absolute format.

NOTE The following examples use the sequences of radii : 25.40, 25.62, 25.97, 27.20, 28.00, 28.30, 28.35, 28.40, 28.43

EXAMPLE Radii expressed in absolute form.

```
Radius          Encoding            Value       Type        Size

First radius    (25.40 * 100)       2540        Absolute    word
Second radius   (25.62 * 100)       2562        Absolute    word
Third radius    (25.97 * 100)       2597        Absolute    word
Fourth radius   (27.20 * 100)       2720        Absolute    word
Fifth radius    (28.00 * 100)       2800        Absolute    word
Sixth radius    (28.30 * 100)       2830        Absolute    word
Seventh radius  (28.35 * 100)       2835        Absolute    word
Eighth radius   (28.40 * 100)       2840        Absolute    word
Ninth radius    (28.43 * 100)       2843        Absolute    word
```

If the difference in the magnitudes of two sequential radii is in the range which can be expressed by the differential data type, the magnitude of the second can be expressed in a byte (eight bits). An AD flag is inserted in the data stream indicating that subsequent radii are expressed in differential form unless and until another type-shifting flag appears in the stream. Each subsequent differential value is added to the decoded radius of its predecessor.

EXAMPLE Radii expressed in differential form.

```
Radius          Encoding                Value       Type          Size

First radius    (25.40 * 100)           2540        Absolute      word
AD Flag                                 -32768      Absolute      word
Second radius   (25.62-25.40 * 100)     22          Differential  byte
Third radius    (25.97-25.62 * 100)     35          Differential  byte
DA Flag                                 -127        Differential  byte
Fourth radius   (27.20 * 100)           2720        Absolute      word
AD Flag                                 -32768      Absolute      word
Fifth radius    (28.00-27.20 * 100)     80          Differential  byte
Sixth radius    (28.30-28.00 * 100)     30          Differential  byte
Seventh radius  (28.35-28.30 * 100)     5           Differential  byte
Eighth radius   (28.40-28.35 * 100)     5           Differential  byte
Ninth radius    (28.43-28.40 * 100)     3           Differential  byte
```

In the above example, it is necessary to revert to the absolute form to express the fourth radius because the difference between the fourth and third (27.20 – 25.62, or 1.58) exceeds the range which can be expressed using the differential form.

If the difference in the magnitude of the differential values of two sequential radii is in the range which can be expressed by the incremental data type, the second radius can be expressed in a nibble (four bits). A DI flag is inserted in the data stream indicating that subsequent radii are expressed in incremental form unless and until an ID flag appears in the stream. Each subsequent incremental value is added to the current differential value, which is then added to the decoded radius of its predecessor.

EXAMPLE Radii expressed in incremental form.

Radius	**Encoding**	**Value**	**Type**	**Size**
First radius	(25.40 * 100)	2540	Absolute	word
AD Flag		-32768	Absolute	word
Second radius	(25.62-25.40 * 100)	22	Differential	byte
Third radius	(25.97-25.62 * 100)	35	Differential	byte
DA Flag		-127	Differential	byte
Fourth radius	(27.20 * 100)	2720	Absolute	word
AD Flag		-32768	Absolute	word
Fifth radius	(28.00-27.20 * 100)	80	Differential	byte
Sixth radius	(28.30-28.00 * 100)	30	Differential	byte
Seventh radius	(28.35-28.30 * 100)	5	Differential	byte
DI Flag		-128	Differential	byte
Eighth radius	((28.35-28.30)-(28.40-28.35) * 100)	0	Incremental	nibble
Ninth radius	((28.40-28.35)-(28.43-28.40) * 100)	-2	Incremental	nibble

Because frame shapes usually comprise gentle arcs and most tracings are expressed using a large number of radii (four hundred or more), most shapes can be expressed using the incremental form for the entire packet, except for the starting radius.

EXAMPLE A tracing expressed in format #4, packed binary, requires 59 bytes including the escape characters. As above, the first example shows the data prior to the escape process (described in 5.1.7.3) having been applied, the second, afterwards. In the first example, the switch flags are shown in **bold**. In the second, the escape sequences are shown in **bold**.

TRCFMT = 4;40;E;R;F
R=af 09 **00** **80** 68 16 b2 **81** 5a 09 cd
08 **00** **80** 8c a3 bb d8 f3 11 32 53
70 **81** 12 09 a7 09 27 0a **00** **80** 2e
be **80** 4b c**8** c3 b9 b1 **80** d**8** b8 c8
dd f2 05 1b 2a 32 **80** 6b **8**3 c4 d5 e**0<**CR/LF**>**

TRCFMT = 4;40;E;R;F
R=af 09 00 80 68 16 b2 81 5a 09 cd
08 00 80 8c a3 bb d8 f3 **1b** 91 32 53
70 81 12 09 a7 09 27 **1b** 8a 00 80 2e
be 80 4b c8 c3 b9 b1 80 d8 b8 c8
dd f2 05 **1b** 9b 2a 32 80 6b 83 c4 d5 e0 <CR/LF>

Refer to the code example in Annex B for further details.

5.4.15 Special considerations for binary formats

5.4.15.4 Sixteen-bit numbers in binary data streams are represented in Intel 80X86 byte order in which the first byte is the low order or least significant byte and the second byte is the high order or most significant byte. Some computer systems (e.g., systems running on Motorola 680xx processors) will have to swap bytes internally in order to get the data to process correctly. An example of this is provided in the source code example in annex B.

5.4.15.5 Care must be taken to treat the data as signed or unsigned as called for by the data formats described herein.

5.4.15.6 It is important to observe the rules for encoding reserved characters described in 5.1.7.

5.4.15.7 Binary data shall begin immediately after the label separator and end immediately prior to the record separator; unlike for ASCII data, superfluous spaces would be catastrophic.

5.4.16 Trace Format Negotiation

Hosts and devices negotiate which trace format to use from amongst the formats mutually supported either during "initialization" (see section 6.2.8) or, in the absence of initialization, during each upload or download session (see example at section 6.4.3). In either case, the device proposes a list of trace formats, arranged in the device's order of "preference", and the host responds with its choice.

5.5 Drilling Records

5.5.1 General

5.5.1.1 DRILL superseded by DRILLE

The DRILL record introduced in version 3.03 is deprecated in favor of the DRILLE record introduced in version 3.04. The DRILLE record supports feature location by two new reference schemes in addition to the Cartesian reference supported by the DRILL record.

5.5.1.2 One DRILLE record per feature

Each feature to be produced on a lens is represented by a discrete DRILLE record, which means that packets containing any DRILLE records may contain multiple DRILLE records.

5.5.1.3 Field expression

The DRILLE record contains multiple fields. The first five – the minimum required to locate a simple hole – are required for all features. All or some of the remaining fields may be required to completely specify the feature. The field separators for fields which are not required may be present or absent, that is, the record may simply end at the last required field. In some cases, required fields may follow non-required ones, in which case, the intervening field separators must appear. When they do appear, the fields implicitly expressed thereby may contain nothing (an "empty" field, expressed as two adjacent field separators) or the unknown data indicator. Because whitespace is generally allowed in records, it is also permissible for spaces to appear in such fields.

5.5.1.4 Decentered tracings

Feature coordinates are unaffected by non-zero values in FBFCIN and FBFCUP fields, which indicate that the shape is to be decentered.

5.5.1.5 Auto-Format Initialization, DRILL, and DRILLE

Because multiple coordinate location reference modes are defined for use with the DRILLE record, hosts and devices must negotiate a mode acceptable to both as described in 5.5.3, either during Initialization, or during each request. It is therefore not necessary for the DRILLE record to appear in a Record Label List during Auto-Format Initialization. A device may, however, continue to support the now-deprecated DRILL record, which would appear in such Record Label Lists. The situation will arise, therefore, wherein a device will include both some number of DRLFMT records, and a DRILL label in a Record Label List, during the same Initialization session. In this instance, a host that supports the DRILLE record shall complete the DRLFMT negotiation as specified in 5.5.3, but shall not send a DRILL record in any download sessions subsequent to the Initialization session. In effect, such hosts shall disregard the DRILL record in the Record Label List. Hosts that do not yet support the DRILLE record simply ignore the DRLFMT records (pursuant to 5.1.5), and send DRILL records according to the rules that applied in prior versions of the Standard.

In a given packet, all features expressed in DRILLE records must use the same Feature Reference, described in 5.5.2.2.

In order to use DRILLE records in a session, DRLFMT negotiation must occur during initialization. Absent Initialization, DRILLE records cannot be used.

5.5.2 Constituent fields of the DRILLE record

5.5.2.1 Eye Side

The first field in the DRILLE record shall contain one of the following characters: “R”, signifying that the record contains instructions for a feature to be applied to the right lens; “L”, signifying that the record contains instructions for a feature to be applied to the left lens, or “B” signifying that the record contains instructions for a feature to be applied to both lenses. In the case of a feature to be applied to both lenses, the data describes locations for a right lens, which shall be mirrored by the device to apply the feature to the left lens. This field shall not be empty or absent.

A value of zero (“0”) may appear in this field, indicating that there are no features on either lens for this job. See 5.5.2.10.9.

5.5.2.2 Feature Location and Reference

The second field in the DRILLE record contains from one to three characters as described below. In each case, Cartesian coordinates are used; the x-axis is collinear with a line which bisects the vertical dimension of a box that circumscribes the frame trace and the y-coordinate is referenced to the x-axis. The x-origin is positioned as described below. This field may be empty, in which case Center Reference is implied. It shall not be unknown, nor absent.

The coordinates specified for a feature refer to the center of that feature. For an irregularly-shaped feature, the coordinates refer to the center of an implied rectangle circumscribing the feature.

The first character shall specify the location of the origin of the x-axis (“reference specifier”):

“**C**”, Center Reference, indicates that both the x and y coordinates of the feature are referenced to the origin of a Cartesian grid located at the box center of the frame trace;

“**E**“, Edge Reference, indicates that the x-coordinate of the feature is referenced to the edge of the lens at the y-coordinateof the feature and the y-coordinate is referenced to the x-axis;

“**B**”, Box Reference, indicates that the x-coordinate of the feature is referenced to the edge of a box circumscribing the shape and the y-coordinate is referenced to the x-axis.

“R”, Relative Reference, is used only in conjunction with Explicit Grouping, defined in 5.5.2.10.6. In this case, the x- and y-coordinates of controlled features are referenced to the controlling feature. Relative Reference can only be specified for controlled features; controlling features must use Center, Edge, or Box Reference.

“0” (zero) in the second field indicates that there are no features to be drilled on the side(s) specified in the first field.

For Edge Reference and Box Reference, the next character shall indicate the side of the lens on which the feature appears:

“**N**“ indicates that the feature is on the nasal side of the lens;

“**T**“ indicates that the feature is on the temporal side of the lens.

In all cases, an additional character may appear, the mounting-surface specifier, which specifies the surface of the lens onto which the mounting fixture attaches:

“F” indicates that the mounting fixture attaches to the front of the lens;

“R” indicates that the mounting fixture attaches to the rear of the lens.

In the absence of the mounting-surface specifier, F (front mounting) shall be assumed.

Therefore, the permissible combinations are: “C”, “CF”, “CR”, “EN”, “ENF”, “ENR”, “ET”, “ETF”, “ETR”, “BN”, “BNF”, “BNR”, “BT”, “BTF”, “BTR”, “R”.

5.5.2.3 Sign Convention

Center Reference coordinates are signed conventionally, with positive *x* values to the right of the origin and positive *y* above the origin. Edge and Box Reference coordinates are signed such that positive *x* values are invariably towards the center of the lens.

5.5.2.4 “Start” Coordinates

The third and fourth fields are the “starting” x and y coordinates respectively. This is the location of the center of the hole. When making a slot this is the location of one end of the slot; the location of the other end of the slot is described in the sixth and seventh fields. If Feature Type 2 is specified in the tenth field of this Record, the “start” coordinates specify one corner of a rectangular region (see field 9 below), and the “end” coordinates specify the diagonally-opposite corner. See Figure A.1. This field may not be empty, unknown, or absent.

NOTE The terms “start” and “end” are used descriptively herein and do not specify the manner in which a lens must actually be machined.

5.5.2.5 Diameter

5.5.2.6 The fifth field is the hole diameter (or, in the case of a slot, the width of the slot). When this field is empty, absent, or unknown, the hole shall be drilled to a default diameter determined at the device (which may be the diameter of the drilling tool).

5.5.2.7 “End” Coordinates

The sixth and seventh fields are the “ending” x and y coordinates, respectively, of a slot or rectangular feature. This field may be empty or absent in which case a round hole is drilled. If Feature Type 2 is specified in the ninth field, end coordinates must appear in fields 6 and 7 to specify the corner of the rectangular region diagonally opposite the one specified in the “start” coordinates (see field 9 below). See Figure A.1.

5.5.2.8 Depth

The eighth field is the depth in millimeters of the feature. When absent, empty, unknown, or zero, the feature is drilled through the entire thickness of the lens. When a feature is not drilled through the entire thickness of the lens, the feature is drilled on the side specified in the Feature Reference field (see 5.5.2.2). When no mounting-surface specifier appears, the front surface is assumed.

5.5.2.9 Feature Type

The ninth field in the DRILLE record contains the “Feature Type”, an integer values which differentiates the use of the coordinate fields in the record. In all cases, the coordinates are referenced as described in 5.5.2.2. If this field is empty, absent, or unknown, feature type 1 is implied.

5.5.2.9.1 Feature Type 1

A Feature Type value of 1 (or zero, see Note below) specifies a hole or a slot. The Start Coordinates (see 5.5.2.4) specify the location of a hole or the starting location of a slot. The End Coordinates (see 5.5.2.7) specify the ending location of the slot.

NOTE Developers should note that the Feature Type field may be absent, in which case the field value might reasonably be inferred to be zero. A value of zero should therefore be treated the same as a value of one; that is, either zero or one should be interpreted as specifying a hole or slot. Which of these two features is specified in a given record – hole or slot – depends on the presence of “End Coordinates” that differ from the specified “Start Coordinates”, in which case a slot is specified.

5.5.2.9.2 Feature Type 2

Feature Type 2 specifies a rectangular region to be milled out of the lens. The Start Coordinates (see 5.5.2.4) specify the upper outside corner of the region, and the End Coordinates (see 5.5.2.7) specify the lower inside corner of the region. “Outside” refers to the side of the milled region furthest from the center of the lens; “inside” refers to the side of the region closest to the center of the lens. When the fifth field is non-zero, the start and end coordinates refer to the center of the specified diameter. The coordinates refer to the center of a circle the diameter of which is specified in 5.5.2.5, A diameter of zero specifies sharp corners; a diameter greater than zero specifies corners radiused to one-half the diameter. Therefore, the dimensions of the feature will be the differences between the absolute values of the “start” and “end” coordinates plus the diameter specified in5.5.2.5.

5.5.2.10 Angular Orientation of Features

The Drill Reference Axis is the normal to the lens front surface at the lens box center. See Figures A.1, A.2, and A.3. The angle specified for a feature applies at the geometric center of an individual feature; when features are grouped according to 5.5.2.10.8, the angle applies at the geometric center of a rectangle circumscribing the features in the group.

5.5.2.10.1 Angle Mode

The tenth field specifies the angle at which the feature is to be drilled. It may contain one of the following characters: “B”, signifying that the feature is drilled normal to the lens back surface at the feature location; “F”, signifying that the feature is drilled normal to the lens front surface at the feature location; or “A”, signifying that the feature is drilled at the angles specified in the eleventh and twelfth fields. . If this field is absent or empty and the feature is not part of a group with an explicitly specified Angle Mode, Angle Mode “F” is assumed. If the Angle Mode is not specified and the feature is included in a group, the rules in 5.5.2.10.8apply.

When “F” or “B” is specified, either the Lateral Angle (see 5.5.2.10.2) or Vertical Angle (see 5.5.2.10.3) – but not both – may be specified. In that case, the angle specified is applied to the appropriate axis, and the angle not specified is the normal to the lens front or back surface (for modes “F” and “B” respectively).

5.5.2.10.2 Lateral Angle

The eleventh field specifies the lateral angle, relative to the Drill Reference Axis, at which the feature is drilled. If the Angle Mode specified in the tenth field is “A”, this field may not be absent, empty, or unknown. The lateral angle is specified in degrees and indicates an angular deviation from the Drill Reference Axis. A positive number signifies a deviation towards the nasal on a right lens, and towards the temporal on a left lens. See Figure A.2.

5.5.2.10.3 Vertical Angle

The twelfth field specifies the vertical angle, relative to the Drill Reference Axis, at which the feature is drilled. If the Angle Mode specified in the tenth field is “A”, this field may not be absent, empty, or unknown. The vertical angle is specified in degrees and indicates an angular deviation from the Drill Reference Axis. A positive number signifies a deviation towards the top of the lens. See Figure A.3.

5.5.2.10.4 Minimum Thickness

The thirteenth field specifies the minimum allowable lens thickness (in millimeters) at the feature location. This field may be absent, empty, or unknown. When this field is present in a packet together with MINDRL, this value shall take precedence.

5.5.2.10.5 Maximum Thickness

The fourteenth field specifies the maximum allowable lens thickness (in millimeters) at the feature location. This field may be absent, empty, or unknown. When this field is present in a packet together with MAXDRL, this value shall take precedence. This value reflects limitations of the mounting hardware to be used.

5.5.2.10.6 Explicit Group Indicator

The fifteenth, optional field allows the explicit grouping of features. When this field is empty or absent, implicit grouping applies as specified in 5.5.2.10.8. When present, this field shall contain a TEXT value comprising one or more upper-case alphabetic characters followed by one or more numeric characters forming a numeric string representing an ordinal number (examples: A1, BB99). The alphabetic character(s) indicate that the feature belongs to a group of features, all of which are identified as belonging to the group by virtue of containing the same alphabetic character(s) in this field (the “group indicator”). Group indicators shall start with “A” and proceed through the alphabet; in the event that more than 26 identifiers are required, a composite string of characters shall be used, starting with AA, and proceeding with AB, AC, and so on. The numeric portion of the field value indicates the ordinal position of the feature in the group (“sequence number”). The value of the controlling feature shall be “1”; subsequent, controlled features in the group shall have sequentially increasing values. All controlled features in a group are drilled at the same angle as the controlling feature. Controlled features may contain a reference specifier of “R” in which the start and end coordinates are referenced to the controlling feature.

5.5.2.10.7 Resizing

The sixteenth, optional field controls what happens to the position of a feature when a lens is resized. Features may remain fixed in either or both dimensions, or may move in accordance with the resizing. The default value of zero (0) indicates that the feature shall move proportionally with the change in size. A value of one (1) specifies that the x-cooridinate of the feature shall remain fixed, but the y-coordinate shall move proportionally with the change in the vertical dimension of the shape. A value of two (2) specifies that the y-coordinate shall remain fixed, but the x-coordinate shall move proportionally with the change in the horizontal dimension of the shape.

5.5.2.10.8 Feature Grouping

Features that appear on the same side of a lens, nasal or temporal, shall be implicitly grouped and shall be drilled at the same angle, except to the extent that features are explicitly identified as belonging to a particular group pursuant to 5.5.2.10.6. The Angle Mode for the group may be specified by assigning an Angle Mode to one, but not more than one, feature in the group. All features in a group are drilled parallel to one another. The angle at which a group of features is drilled is referenced to the geometric center of a rectangle circumscribing all the features in the group.

The presumption of grouping can be defeated by specifying the Angle Mode for each feature separately.

EXAMPLE – A minimal DRILLE record :

DRILLE=B ; C; -17.0 ; 10.32

EXAMPLE – A fully-populated DRILLE record :

DRILLE=B ; C; -17.0;10.32;2.3;-15.0;10.32;1.5 ;1; A;-15.0;5.0

5.5.2.10.9 No features present

When no drill features are present on any lenses for a job, an abbreviated form of the DRILLE record containing a single field with the value “0” (zero, nil) shall appear in the corresponding data packet.

If the Host does not handle drilling information, the unknown data indicator shall appear in this field in a single instance of the DRILLE record.

EXAMPLE – No drill features present for a job:

DRILLE=0

EXAMPLE – Host does not handle drilling information:

DRILLE=?

When drill features are present on one lens for a job, but not the other, DRILLE records will appear for the side to be drilled, and no DRILLE records appear for the side not to be drilled. There is no explicit indication of the absence of features for one side, other than the absence of DRILLE records for that side.

NOTE Following 5.1.4, hosts or devices that do not support DRILL may send “DRILL=?”. However, because the use of the DRILLE record requires Drill Format Negotiation (5.5.3), a device or host that does not support DRILLE will contain no references to DRLFMT or DRILLE, as the DRLFMT records would be ignored pursuant to 5.1.5.

5.5.3 Drill Format Negotiation

The Drill Feature Reference must be negotiated between hosts and devices in a manner similar to that required for tracing formats (see 6.2.8). The DRLFMT record is used for this purpose. DRLFMT appears only during Initialization, or, when Initialization is not used, in request packets; unlike TRCFMT, it never appears in data packets. The DRLFMT record contains a single field containing either the character “C”, signifying Center Reference; “E”, signifying Edge Reference; or “B”, signifying Box Reference. A device may support any number of Feature Location References, so, from zero to three DRLFMT records may appear.

NOTE – Because DRLFMT is an Interface Record, it never appears in the Record Label List that may be included in Initialization packets.

EXAMPLE – A Preset Initialization session in which the device proposes four Trace Formats and three Feature Reference Locations.

```
DEVICE                           HOST
<FS>REQ=INI<CR/LF>
<RS>
<GS>
                                 <ACK>
                                 <FS>ANS=INI<CR/LF>
                                 STATUS=0<CR/LF>
                                 <RS>
                                 <GS>
<ACK>
<FS>ANS=INI<CR/LF>
STATUS=0<CR/LF>
```

```
DEV=EDG<CR/LF>
VEN=GC<CR/LF>
MODEL=LE-3<CR/LF>
TRCFMT=4;400;E;R<CR/LF>
TRCFMT=4;512;E;R<CR/LF>
TRCFMT=1;400;E;R<CR/LF>
TRCFMT=1;512;E;R<CR/LF>
DRLFMT=C<CR/LF>
DRLFMT=E<CR/LF>
DRLFMT=B<CR/LF>
<RS>
<GS>
                           <ACK>
                           <FS>ANS=INI<CR/LF>
                           STATUS=0<CR/LF>
                           DEF=;1234<CR/LF>
                           TRCFMT=4;512;E;R<CR/LF>
                           DRLFMT=C<CR/LF>
                           <RS>
                           <GS>
<ACK>
```

EXAMPLE – A Preset edger request session in which the device proposes four Trace Formats and three Feature Reference Locations.

```
DEVICE                     HOST
<FS>REQ=EDG<CR/LF>
JOB=1234<CR/LF>
VEN=GC<CR/LF>
MODEL=LE-3<CR/LF>
TRCFMT=4;400;E;R<CR/LF>
TRCFMT=4;512;E;R<CR/LF>
TRCFMT=1;400;E;R<CR/LF>
TRCFMT=1;512;E;R<CR/LF>
DRLFMT=C<CR/LF>
DRLFMT=E<CR/LF>
DRLFMT=B<CR/LF>
<RS>
<GS>
                           <ACK>
                           <FS>ANS=EDG<CR/LF>
                           JOB=1234<CR/LF>
                           STATUS=0<CR/LF>
                           DO=B<CR/LF>
                           BEVP=7<CR/LF>
                           BSIZ=0;0<CR/LF>
                           CIRC=145.33;145.52<CR/LF>
                           CLAMP=?;?<CR/LF>
                           CSIZ=0.0;0.0<CR/LF>
                           DBL=14.3<CR/LF>
```

```
                    DIA=70.0;70.0<CR/LF>
                    DRILLE=B;C;-18.4;10.3<CR/LF>
                    DRILLE=B;C;-15.4;10.3<CR/LF>
                    EPRESS=?<CR/LF>
                    ERDRIN=-2.5;-2.5<CR/LF>
                    ERDRUP=5.0;5.0<CR/LF>
                    ERNRIN=2.5;2.5<CR/LF>
                    ERNRUP=14.0;14.0<CR/LF>
                    ERSGIN=0;0<CR/LF>
                    ERSGUP=2.0;2.0<CR/LF>
                    ETYP=1<CR/LF>
                    FBFCIN=0;0<CR/LF>
                    FBFCUP=0;0<CR/LF>
                    FBSGIN=3.5;3.5<CR/LF>
                    FBSGUP=2.0;2.0<CR/LF>
                    FCRV=5.5;5.3<CR/LF>
                    FPINB=0.5;0.5<CR/LF>
                    FTYP=1<CR/LF>
                    GDEPTH=?;?<CR/LF>
                    GWIDTH=?;?<CR/LF>
                    IPD=32.5;32.5<CR/LF>
                    LMATTYPE=1;1<CR/LF>
                    LMATID=?;?<CR/LF>
                    LTYPE=PR;PR<CR/LF>
                    MCIRC=?<CR/LF>
                    NPD=31.0;31.0<CR/LF>
                    PINB=1.0;1.0<CR/LF>
                    POLISH=1<CR/LF>
                    SEGHT=23;23<CR/LF>
                    TNORM=?<CR/LF>
                    ZTILT=3.5;3.2<CR/LF>
                    TRCFMT=4;512;E;R<CR/LF>
                    R=<… radius data …>
                    ZFMT=0
                    <RS>
                    <GS>
<ACK>
```

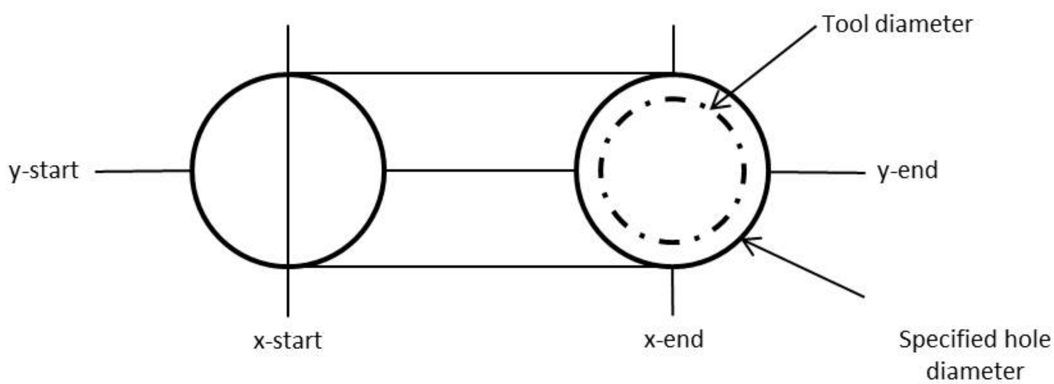

Figure A.1, showing Starting and Ending Coordinates

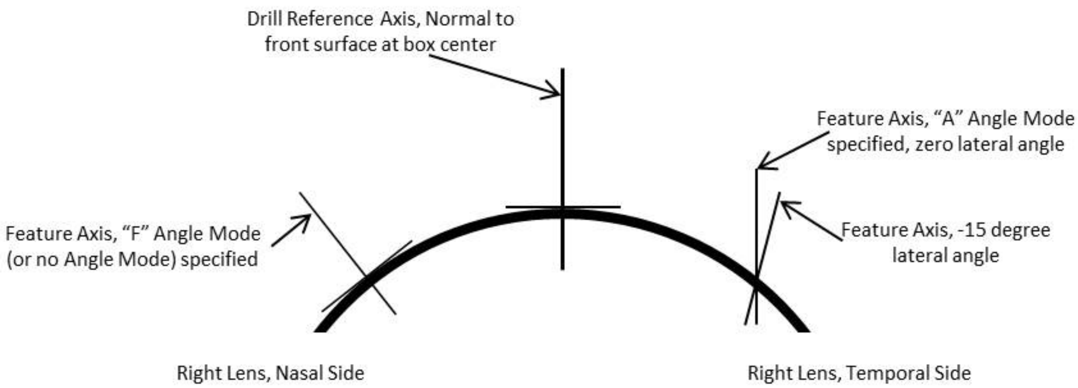

Figure A.2, Showing lateral angle

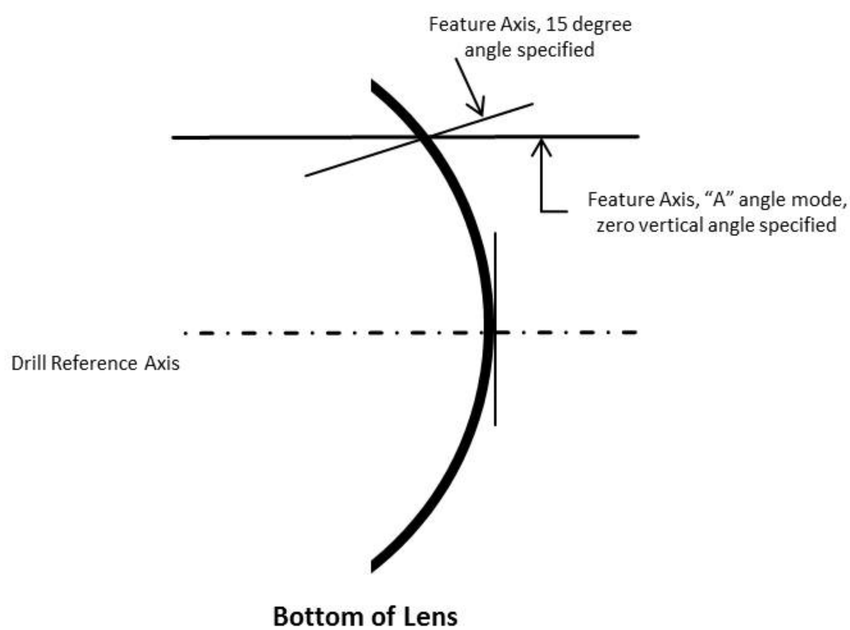

Figure A.3, Showing vertical angle

5.6 Side Drilling Records

Drill features applied to the edges of lenses must be described differently from those applied to the surfaces of lenses. The DRILLS record is used for this purpose.

5.6.1 Constituent fields of the DRILLS record

5.6.1.1 Eye Side

The first field in the DRILLS record shall contain one of the following characters: “R”, signifying that the record contains instructions for a feature to be applied to the right lens; “L”, signifying that the record contains instructions for a feature to be applied to the left lens, or “B” signifying that the record contains instructions for a feature to be applied to both lenses. In the case of a feature to be applied to both lenses, the data describes locations for a right lens, which shall be mirrored by the device to apply the feature to the left lens. This field shall not be empty or absent.

5.6.1.2 Feature Type

The second field in the DRILLS record contains an integer that indicates the “Feature Type”. Feature type “1” indicates a hole or a slot, and feature type “2” indicates a rectangle. If this field is empty, absent, or unknown, feature type 1 is implied.

5.6.1.3 Start Angle

The third field is the start angle for the feature as viewed from the lens front in degrees. When making a slot this is the location of one end of the slot as viewed from the lens front; the location of the other end of the slot is specified in the “End Angle” field. See Figure A-1. This field may not be empty, unknown, or absent.

NOTE: The terms “start” and “end” are used descriptively herein and do not specify the manner in which a lens must actually be machined.

5.6.1.4 Start Distance

The fourth field is the start distance of the feature from the lens front as viewed from the lens side in millimeters. When making a slot this is the location of one end of the slot from the lens front as viewed from the lens side; the distance location of the other end of the slot is specified in the “End Distance” field. See Figure A-4.

5.6.1.5 End Angle

The fifth field is the end angle for the feature as viewed from the lens front in degrees. When making a slot this is the location of the last end of the slot as viewed from the lens front; the location of the other end of the slot is specified in the “Start Angle” field. See Figure A.1. This field may not be empty, unknown, or absent. See Figure A-3.

5.6.1.6 End Distance

The sixth field is the end distance of the feature from the lens front as viewed from the lens side in millimeters. When making a slot this is the location of the last end of the slot from the lens front as viewed from the lens side; the distance location of the first end of the slot is specified in the “Start Distance” field. See Figure A-4.

5.6.1.7 Diameter

The seventh field is the hole diameter (or, in the case of a slot or rectangle, the width of the slot or rectangle). When this field is empty, absent, or unknown, the hole shall be drilled to a default diameter determined at the device (which may be the diameter of the drilling tool).

5.6.1.8 Depth

The eighth field is the depth in millimeters of the feature. This field is mandatory.

5.6.1.9 Lateral Angle

The ninth field specifies the lateral drilling of the drill feature in degrees. A positive number signifies a deviation from the normal towards the lens front, and a negative number signifies a deviation towards the lens back. See Figure A.2. If this field is absent, the drill angle will be normal to the side of the lens.

5.6.1.10 Rotational Angle

The tenth field specifies the rotational drilling angle of the feature in degrees. A positive number signifies a deviation from the normal in a counterclockwise direction as viewed from the lens front, and a negative number signifies a deviation from the normal in a clockwise direction as viewed from the lens front. See Figure A.3. If this field is absent, the rotational drill angle will be normal to the side of the lens.

5.6.1.11 No Side Drilling features present

When no side drilling features are present on any lenses for a job, an abbreviated form of the DRILLS record containing a single field with the value “0” (zero, nil) shall appear in the corresponding data packet.

EXAMPLE – No drill features present for a job:

DRILLS=0

When drill features are present on one lens for a job, but not the other, DRILLS records will appear for the side to be drilled, and no DRILLS records appear for the side not to be drilled. There is no explicit indication of the absence of features for one side, other than the absence of DRILLS records for that side.

Location of a Side Drilling Feature (Start Angle, End Angle)
Viewed from the Lens Front

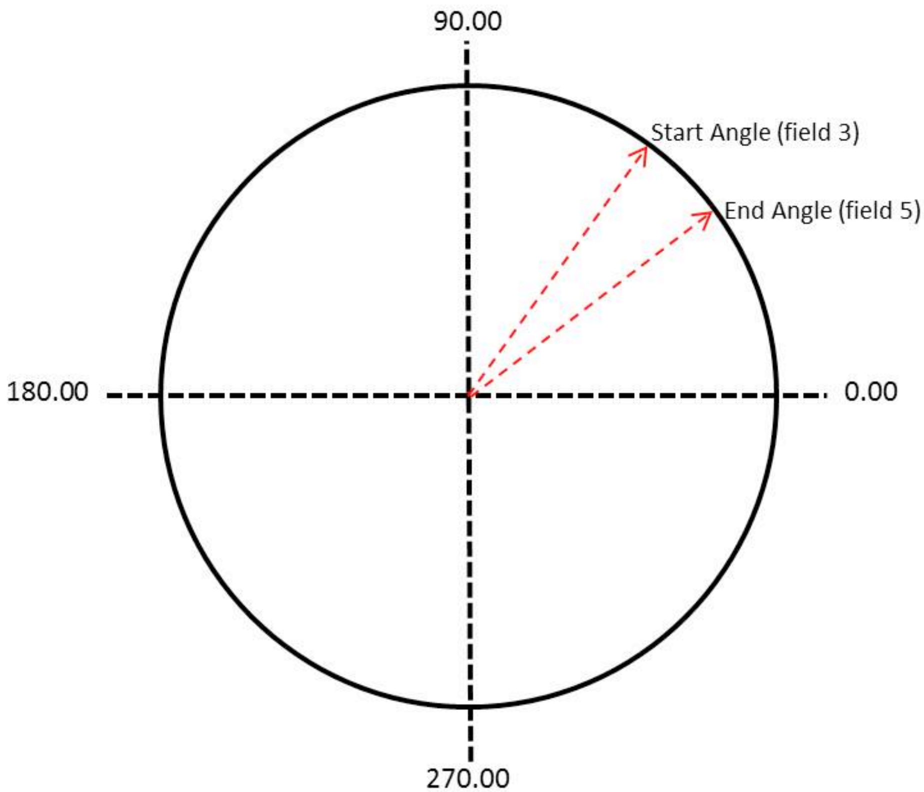

Figure A.4

Side Drilling Start Distance, End Distance and Lateral Angle Viewed from the Lens Side

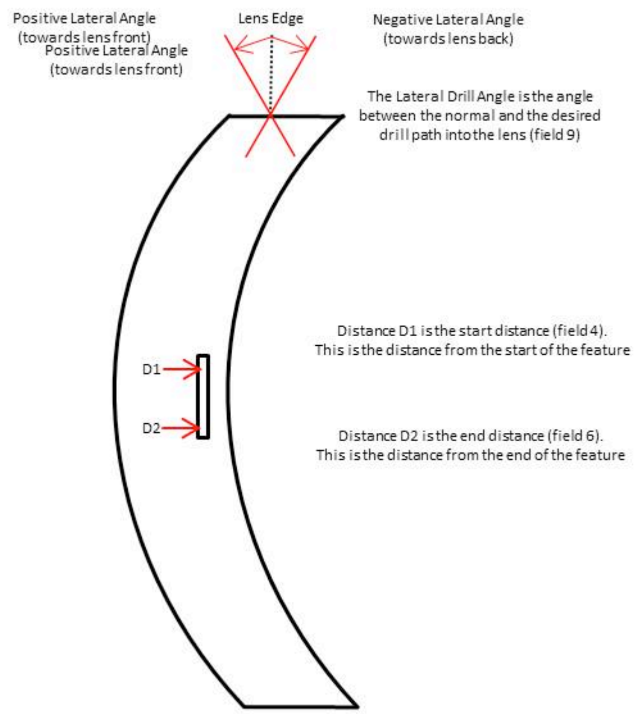

Figure A.5

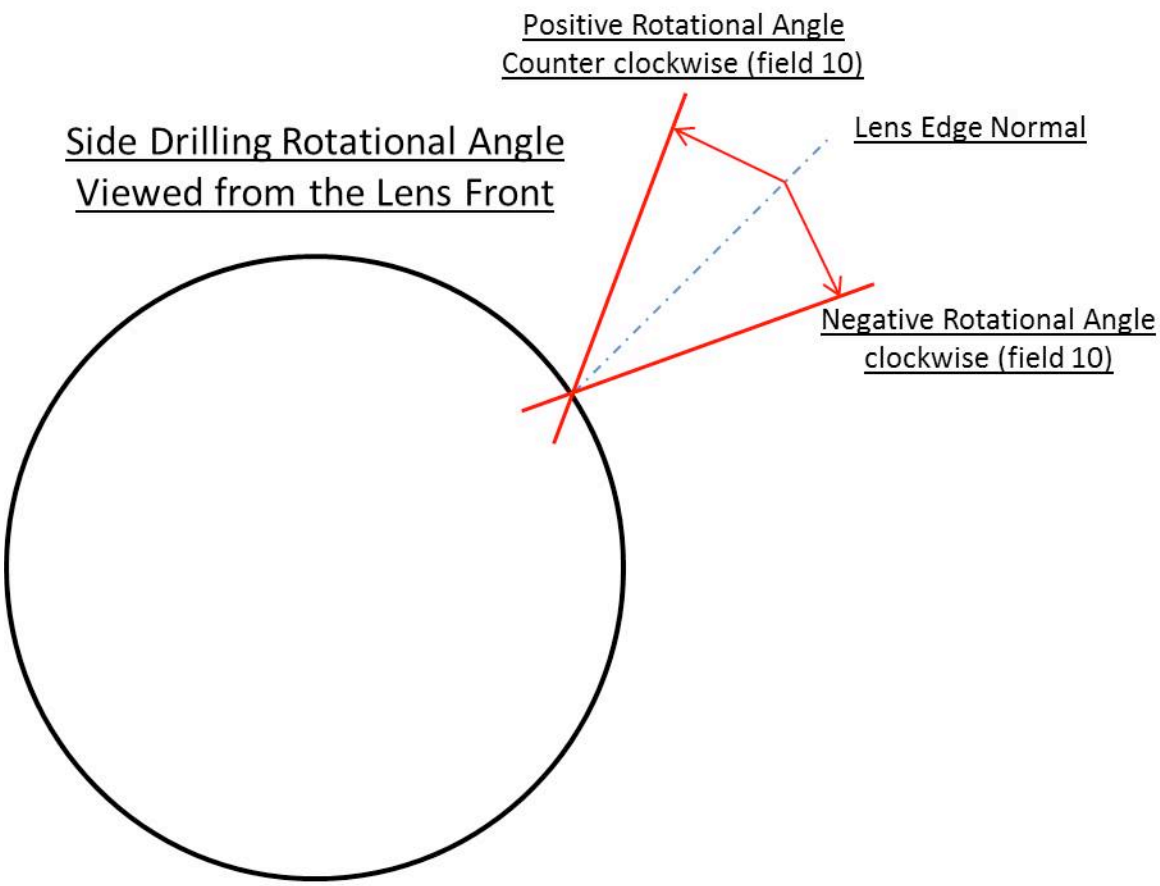
Positive Rotational Angle
Counter clockwise (field 10)
Side Drilling Rotational Angle
Viewed from the Lens Front
Lens Edge Normal
Negative Rotational Angle
clockwise (field 10)

Figure A.6

5.7 Bevel Profiles

The standard bevel profiles shown in Figure A.7, with the exception of “Shelf” are associated with field values for the ETYP record. See Table A.1.

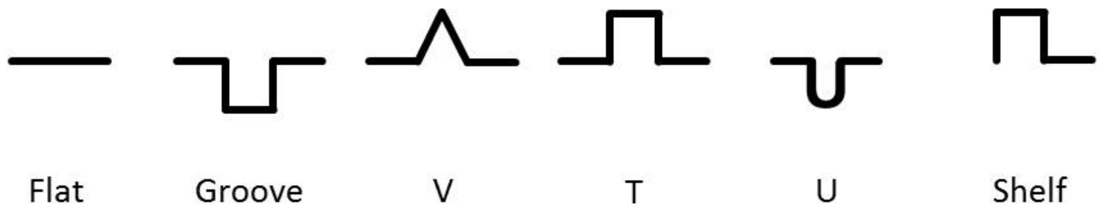

Figure A.7 Standard Bevel Profiles

5.8 Shelf Datasets

5.8.1 Expression of shelf data

Shelf data is used to create “shelves” on the edges of lenses, wherein the profile of the edge of the lens incorporates a reduction in the radius from the edging center to the edge, such that a flat surface is produced behind the front surface of the lens, or in front of the back surface of the lens, or both. See Figure A.7.

Shelf data begins with a SHLFFMT record that specifies the preferred format in which the shelf is to be expressed, the number of points to be transmitted, whether the radii are equiangular, and the orientation of the shelf. If more than one shelf is required on a lens, then each shelf shall be represented by a SHLFFMT record followed by all data associated with that shelf.

A device indicates its desire to upload or download shelf data by including one or more SHLFFMT records in either its initialization packet, or, if no initialization is done, its request packet. All supported formats and numbers of points are listed in the order of the device's preference. It is not necessary to include every combination of side specifiers (B, R, and L) or point number preferences; because of the requirement specified in 1.4.13, only the device's preferred mode need be specified. Devices that neither upload nor download shelf data do not include any SHLFFMT records in their initialization or request packets.

When shelf data is requested by a device, but a shelf is not present or the data is unavailable from the host, a special form of the SHLFFMT record is sent, composed of a single field with the integer value 0. In the case that two-side data has been negotiated, and only a single side is sent, SHLFFMT=0 is not sent for the missing side.

EXAMPLE: Shelf data requested by device, but shelf data is not present or unavailable from the host:

```
SHLFFMT=0
```

EXAMPLE: Shelf data requested by device and available from host:

```
SHLFFMT=1;400;E;R;B<CR/LF>
S=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
S=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
<etc.>
S=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
```

5.8.2 Shelf Radius Data

These are contained in "S" records. Angle data for shelf radius data is contained in "SA" records, and if available, shelf depth data is contained in “SD” records. For the ASCII format, all of the records follow the 80-character line limit rule, therefore there may (in fact, will likely) be multiple S, SA, SD, and SW records for each tracing. For BINARY data, the line limit rule will not apply and there will be only one S, SA, SD, and SW record as needed. Shelf radius and shelf depth data is expressed hundredths of a degree with an implied decimal point.

When even-angle shelf radius data is present, the areas of the lens that do not contain a shelf should be specified with radius values of zero, or they may be specified with radius values that are outside the finished lens edge. If the shelf goes all the way around the lens, there will be no shelf radius values present that are equal to zero.

EXAMPLE Shelf radius data :

a) ASCII format

```
SHLFFMT=1;400;E;R;B<CR/LF>
S=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
S=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
etc…
```

b) Binary formats :

```
SHLFFMT=2;400;E;R;B<CR/LF>
S=<binary data stream><CR/LF>
```

5.8.3 Uneven shelf angles

When these are specified in the SHLFFMT record, angle data is required. It shall appear immediately after its corresponding shelf radius data. Shelf angle data is contained in one or more "SA" records, and shall be expressed in the same format as the shelf radius data to which it corresponds. Shelf angle data is expressed in hundredths of a degree with an implied decimal point and shall be in the range 0-35999. When uneven shelf angles are present, the radius value immediately before and after each shelf segment must have a radius value of zero, or areas that do not contain a shelf may be specified with radius values that are outside the finished lens shape. If the shelf goes all the way around the lens, there will be no shelf radius values present that are equal to zero.

NOTE If shelf depth data is included and uneven angles are specified for the shelf data, the shelf depth data references the same angles as the shelf radius data.

5.8.4 "SD" records

These, when present, shall appear immediately after the corresponding set of “S” or “SA” records. All SD records for a shelf (i.e., a single side, when two are provided) appear together. Each side has its own SD records. SD values represent the depth of the shelf (the extent of the shelf from the front or back surface, as specified in the fifth field of SHLFFMT).

5.8.5 "SW" records

These, when present, shall appear immediately after their corresponding set of "S", "SA", or “SD” records. All SW records for a shelf (i.e., a single side, when two are provided) appear together. Each side has its own SW records. If no SW records are present, then the shelf is assumed to extend to the lens edge at any given point if the S record at that point is >0.

5.8.6 The number of values in SA, SD, and SW records shall equal the number of values in the corresponding S record. See Table A.1 for detailed definitions of S, SA, SD, and SW records.

EXAMPLE A two-eye shelf dataset (abbreviated) showing the required sequence of records

```
SHLFFMT=1;400;U;R;R<CR/LF>
S=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
S=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
…
S=1909;1914;1941;1983;2033;2089;2140;2200;2277;2371<CR/LF>
SA=0;90;180;270;360;450;540;630;720;810<CR/LF>
SA=900;990;1080;1170;1260;1350;1440;1530;1620;1710<CR/LF>
…
SA=35100;35190;35280;35370;35460;35550;35640;35730;35820;35910<CR/LF>
SD=322;322;322;324;324;325;325;326;326;326<CR/LF>
SD=326;327;327;328;328;328;327;327;327;327<CR/LF>
…
SD=325;325;324;324;324;324;323;323;323;322<CR/LF>
…


SHLFFMT=1;400;U;L;R<CR/LF>
S=2517;2450;2379;2318;2247;2168;2086;2014;1958;1923<CR/LF>
S=1909;1914;1941;1983;2033;2089;2140;2200;2277;2371<CR/LF>
…
S=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
SA=0;90;180;270;360;450;540;630;720;810<CR/LF>
SA=900;990;1080;1170;1260;1350;1440;1530;1620;1710<CR/LF>
…
SA=35100;35190;35280;35370;35460;35550;35640;35730;35820;35910<CR/LF>
SD=322;322;322;324;324;325;325;326;326;326<CR/LF>
SD=326;327;327;328;328;328;327;327;327;327<CR/LF>
…
SD=325;325;324;324;324;324;323;323;323;322<CR/LF>
…
```

5.9 Direct Surfacing Records

"Direct Surfacing" is the process of creating prescription lenses by machining lens surfaces more complex than simple tores. In this process, Lab Management Systems ("LMS") communicate prescription data to Lens Design Systems ("LDS") and receive results which include the identification of one or more Surface Definition Files ("SDF") which contain a matrix of points representing the surface (a "surface matrix") in the formats specified in this section.

5.9.1 File Exchange

Communication between the LMS and LDS is accomplished by an exchange of files written to and read from one or more shared storage resources. The LMS writes a file, having an extension of "LDS", containing prescription data required by the LDS. The records required by the LDS may be specified in an initialization file. The LDS retrieves the "LDS file" and produces a lens design. The LDS then writes a file having an extension of "LMS" (an "LMS file") that contains records related to the design, which is retrieved by the LMS. Additionally, the LDS may write a "surface definition file" having an extension of "SDF" (an "SDF file"). The path to the SDF file is specified to the LMS in an LDPATH record, which is retrieved by direct-surfacing-capable equipment when they request job data in download sessions. Alternative methods of delivering SDF data are described in section 5.9.6. There are two exchange modes, which differ in the number of data exchanges that occur to reach a conclusion.

In the simpler of the two, which was the first to be codified in this standard, the LMS submits data to the LDS in a file having a request type of “LDS”, and the LDS produces a file having a request type of “LMS” which contains information related to the lens design in its final form, usually including the location of the SDF file describing the surfaces to be produced.

In the second of the two modes, there is an initial exchange of files intended to allow the LDS to specify an appropriate base curve for the lens to be used. In this case, the LDS file contains a request type of “BRS” (meaning, “lens blank request”) and the LMS file contains a request type of “BAS” (“lens blank answer”). The BRS file may contain the same records as an LDS file would have. The BAS file shall contain at least the records MINFRT, MAXFRT, and OPTFRNT, indicating a range of acceptable true front curves (MINFRT..MAXFRT) and an ideal front curve (OPTFRNT). The BAS file may contain an OPC record, specifying the LDS’ preferred lens blanks for the order.

Regardless of the exchange mode, the file intended for consumption by the LDS has a file extension of “LDS”, and the file intended for consumption by the LMS has a file extension of “LMS”. The two modes are distinguished by the REQ records in the files.

5.9.1.1 File Naming Conventions

The file names (as opposed to extensions) to be used are not specified in this document. It is recommended that file names begin with a job identifier, equivalent to the content of the VCA ”JOB” record. It is suggested that files written by LMS, containing prescription data to be used by an LDS, have the form <job identifier>.LDS. (For example, “1234.LDS”). Files written by an LDS, containing data to be used by LMS should have the form <job identifier>.LMS. Files written by an LDS, containing Surface Data, should have the form <job identifier>.SDF.

Other files written by an LDS, containing data to be used by other devices, shall have the form <job identifier>.<ext> , where <ext> represents the particular extension required by the equipment for which the file is intended.

Responsibility for deletion of these files, at any particular installation, shall be assigned by agreement between the FSG, LDS, and LMS vendors.

5.9.1.2 Collision Avoidance

In order to prevent a receiving system from attempting to read a file while the sending system is still writing it, systems shall write files with extensions other than “LMS”, “LDS” or “SDF” and rename them to the appropriate extension after the files have been completely written and closed. The temporary extensions may consist of the last two characters of the permanent ones prefixed with a dollar sign (“$”).

5.9.1.3 Request Type records and record sequencing

The REQ record at the head of an LDS file shall contain the value “LDS” or “BRS”. The REQ record at the head of an LMS file shall contain the value “LMS” or “BAS”. The REQ record at the head of an SDF shall contain the value “SDF” Files follow the rule for request packets, so that REQ=<request type> must be the first record in every standard file, and JOB must be the second record in every standard file.

5.9.2 Initialization Files

5.9.2.1 The LDS may create an “LDS initialization” file, which shall be identified by an extension consisting of “LDI” (an “LDI” file). The initialization file shall contain a set of records similar to a device’s auto-initialization data packet as defined in section 7.2. The record label list is the set of records that the LDS requires from the LMS. The init file’s name and location shall be agreed upon by LMS and LDS.

Example Initialization file (records requested by LDS from LMS):

```
REQ=LDI<CR/LF>
DEV=LDS<CR/LF>
VEN=SHAMIR<CR/LF>
TRCFMT=1;400;E;R;F<CR/LF>
TRCFMT=1;200;E;R;F<CR/LF>
DEF=SHAMIRDATA<CR/LF>
D=LNAM;SPH;CYL;AX;ADD;MINCTR;MINEDG;OPC<CR/LF>
D=LMATID;PTOK;DBL;IPD;SEGHT;FTYP<CR/LF>
ENDDEF=SHAMIRDATA<CR/LF>
```

5.9.2.1.1 The presence of TRCFMT in the initialization file shall imply that TRCFMT and R records, and A records if needed, shall be created in the input data files. Multiple TRCFMT records may be specified in the initialization file if the LDS supports multiple options. The LMS shall use the first one in the list that it is capable of supporting.

Another example Initialization file:

```
REQ=LDI<CR/LF>
DEV=LDS<CR/LF>
VEN=SEIKO<CR/LF>
TRCFMT=1;24;E;R;F<CR/LF>
DEF=SEIKODATA<CR/LF>
D=LNAM;SPH;CYL;AX;ADD;PRVM;PRVA<CR/LF>
D=MINCTR;MINEDG;ADJSPH;ADJCYL;ADJAX<CR/LF>
D=ADJPRVM;ADJPRVA;PTOK;CRIB;LMATTYPE<CR/LF>
D=MBASE;FRNT;DIA;LIND;TIND;PTPRVM;PTPRVA;PIND<CR/LF>
D=MPD;IPD;FCSGUP;FCSGIN<CR/LF>
ENDDEF=SEIKODATA<CR/LF>
```

In the absence of an initialization file, the LMS shall create a data file containing the records defined in the LDS preset packet

5.9.2.2 The LMS may create an “LMS initialization” file, which shall be identified by a file extension consisting of “LMI” (an “LMI” file). The purpose of this file is to identify to the LDS, the set of direct-surfacing-capable machines for which it needs to produce SDF files, if the SDF produces different files for different machines. The LMI file shall contain one or more DSDEV records, which consists of the DEV, VEN, MODEL, and SN records that will be used to identify a particular machine’s, or type of machine’s, requests, together with a folder specifying the location where the LDS is to put the SDF files for that machine or machine type. It may optionally contain a record label list which indicates to the LDS, the set of data records that the LMS expects to receive from the LDS.

The DSDEV record has the form DSDEV=DEV;VEN;MODEL;SN;LDTYPE;<shared resource name>. An LDS may include an LDPATH record in LMS files for one, some, or all of the machines identified in the DSDEV records.

Example LMS initialization file

```
REQ=LMI
DSDEV=FSG;LOH;VFT;;SDF;\\SOMESERVER\SOMESHAREDFOLDER
DSDEV=FSG;SOM;HSC100;1234;SDF;\\ANOTHERSERVER\ANOTHERSHAREDFOLDER
DEF=INNOVADATA
D=LDSPH;LDCYL;LDAX;LDADD;LDIPD;LDNPD;LDDRSPH;LDDRCYL;LDDRAX
D=LDNRSPH;LDNRCYL;LDNRAX;LDSGSPH;LDSGCYL;LDSGAX;LDPTPRVA;LDPTPRVM;LDSEGHT
D=LDPRVA;LDPRVM;LDCTHK
ENDDEF=INNOVADATA
```

5.9.2.3 Initialization files are static and shall be replaced or changed only in response to changes in software revision or other requirements.

5.9.3 LDPATH Records

An LDPATH record may have one of three forms: a single field containing the fully qualified path name of the single file produced (the original form defined in this standard), a form comprising six or seven fields, or a form comprising six fields, the first of which is an integer identifier, as follows:

5.9.3.1 The original LDPATH form : LDPATH=\\SERVERNAME\SHARENAME\FILENAME.SDF

5.9.3.2 A device-specific form, or pre-standard LDPATH :

LDPATH=DEV;VEN;MODEL;SN; LDTYPE ;<fully qualified file specification>[;<left-side fully qualified file specification>]

The first field contains a device type;

The second field (optional) contains a vendor identifier;

The third field (optional) contains a model identifier;

The fourth field (optional) contains a serial number;

The fifth field contains a literal file type identifier. Type literals include:

SDF – The sixth field points to a standard file as defined herein.

STF, **HMF**, **XYZ**, **STA**, **OPT** – The sixth and seventh fields point to files formatted in accordance with a pre-standard, non-normative, device vendor's specification.

LDX – The sixth field contains private, non-normative information conveyed to a device, used by the device to obtain surface data directly from an LDS.

The Type field may not be unknown, but may be blank, in which case, SDF is presumed. The contents of the type field are included in the data sent to a device in an LDTYPE record.The sixth field contains the fully-qualified path name of the surface definition file. In the case of STF and HMF files, which are chiral, a sixth field contains the right-side surface definition file name, and a seventh field appears to contain the left-side file name.

Optional empty fields are preserved by contingous field separators.

When the multi-field (second) form is used, the LMS must determine which record is appropriate for a given machine's request, based on matching as much of the series of identifiers (DEV, VEN, MODEL, SN) as possible, from left to right. The LDPATH record sent to the machine shall not include the first five fields; that is, from a device's interface, there is only the original, single-field form. A given device only receives a single LDPATH (and LDTYPE) record.

An LDS may send a single LDPATH, in the original, single-field form, together with an LDTYPE record identifying the kind of file pointed to by LDPATH. When the LDTYPE record is absent, SDF is presumed.

In the case of file types other than SDF, LDPATH may contain two filename fields (one for each eye) or other such non-normative data as may be appropriate.

EXAMPLE – LDPATH records as may be included in an LMS file, specifying different file locations and file types for different devices.

```
LDPATH=FSG;LOH;VFT;;SDF;\\SOMESERVER\SOMESHAREDFOLDER\3141.SDF
LDPATH=FSG;SOM;HSC100;1234;SDF;\\ANOTHERSERVER\ANOTHERSHAREDFOLDER\3141.SDF
```

Note that in the communications to a device, the LDPATH record would include only the file specification:

```
LDTYPE=SDF
LDPATH=\\ANOTHERSERVER\ANOTHERSHAREDFOLDER\3141.SDF
```

or

```
LDTYPE=STF
LDPATH=\\SERVER\SHARE\3141R.STF;\\SERVER\SHARE\3141L.STF
```

This is true in all cases – no matter what is sent from the LDS to the LMS, the LDPATH record sent to a device consists of one or two fields (the latter being allowed only for non-standard, chiral surface data files).

5.9.3.3 The third form of LDPATH serves to connect the LDPATH with an encrypted dataset, described below. This form is only sent from an LDS to an LMS. The first field contains an integer, which cannot appear in either of the two other forms, making this form easily distinguishable from the other two. Subsequent fields after the first are the same as the first five fields in the device-specific LDPATH form. The seventh field may contain an identifier to be used by the LDS in a subsequent, discrete communications session with a device. In the same data packet as this record appears, a corresponding LDCRYPT record must also appear. The first field in an LDCRYPT record shall contain the same number as the number in the first field of its corresponding LDPATH record. Note that when sent to a device, the LDCRYPT record has a different form, see section 5.9.7.4.

EXAMPLE – LDPATH records used with encrypted datasets.

```
LDPATH=1;FSG;LOH;VFT;;SDF;\\SERVERNAME\SHARENAME\1234.SDF
LDPATH=2;FSG;SOM;HSC100;1234;SDF;ABCDEFG12345
LDPATH=3;POL;IGC;MDL01
…
LDCRYPT=1;0
LDCRYPT=2;1
LDCRYPT=3;2
EDATA=2;14459;H4sIAHW5VRYA/+3dd3SU1bvo8QQYMvM+kympzKQNaYAECKG3kA6hk4TQS0gCBEIS0gg
JvdfQO6EEoqLYQMWKiCKIoEgTFARBUVRUlA7K/b6TcOTwE36/ey53rfvHdZ3PSTKz3/08ez/PfmeyFvPL
xmN7X3dwSH2lW0zv580xcdaMLFt0QoMmrWwNbAlFtuTYhPi4+OjIpPju3VZ+Mid651czohzs/01ur4nM…
(14459 characters)
```

5.9.4 Surface Definition Datasets

Surface definition data is contained in files which have an extension "SDF" and which contain a request type record having the value "SDF". Some LDS systems produce, and some devices accept, surface definition files, the formats of which were defined by device and/or LDS providers prior to the creation of this standard (STF, HMF, XYZ, STA, and OPT files). Although literals have been added to the standard to provide for the identification of non-standard surface definition files, this has been done solely for the purpose of supporting systems designed and implemented prior to the development of this standard. The only normative surface definition file is the SDF file defined herein; it is the only format that is defined by and included in this standard. The older files can, however, be identified in the LDPATH record as described in section 5.9.3.

5.9.4.1 SURFMT record:

5.9.4.2 SURFMT=*format*;RIL;BIF;ncolumns;nrows;width;height[;slope format][;low-resolution indicator]

The first field indicates the format of the following data. Current support is for one standard format, "1", described below.

The second field contains R or L to indicate to which eye the following data applies.

The third field contains B or F to indicate which side – Back or Front – of the lens is to be surfaced

The fourth and fifth fields contain integers, n columns and n rows, representing the number of data points represented by the following ZZ records, the first applying to the horizontal (x) and the second to vertical (y).

The sixth and seventh fields contain numeric values representing the size of the matrix in millimeters, the sixth applying to the width of the matrix and the seventh to its height.

The eighth, optional field, indicates the presence of slope data at the perimeter of the matrix. Absence of this field, or a value of zero, means that slope data is not present. A non-zero value means that slope records are present in the format indicated by the value. Presently, only one format is defined (see section 5.9.4.4).

The ninth, optional field, indicates that the dataset contains low-resolution surface data. Normally, low-resolution data is contained in "SDL" files, and the flag is not needed to indicate that the data is low-resolution. When SDF data is delivered in a data packet, however (e.g., when SDFMODE=LMS), both low and high-resolution data may be present, so that the indicator is required to distinguish between the two. A value of 1 (one) in the ninth field indicates that the following dataset contains SDL data; a value of 0 (zero; nil), or an empty, absent, or unknown field, indicates SDF data.

"SURFMT=1;R;B;141;141;80.0;80.0" would therefore represent a rectangular array of points, 141 rows and 141 columns, representing data points that are approximately 0.571 mm apart horizontally and 0.571 mm apart vertically (80.0 / (141.0-1.0)) and would not contain peripheral slope data (defined below).

When the subject surface is a back surface, the first point in the first row represents the point at the lower nasal corner of the array for a right eye, and at the lower temporal corner for a left eye; in both cases, viewing the back surface directly.Points are ordered in both cases from left to right, bottom to top. When the subject surface is a front surface, the first point in the first row represents the point at the lower temporal corner of the array for a right eye, and the lower nasal corner for a left eye; in both cases, viewing the front surface directly. Points are ordered in both cases from left to right, bottom to top.

It should be noted that, if a point is desired to be located at the geometric center of the blank, an odd number of rows and columns must be specified (see below). Therefore, an odd number of points for both X and Y is strongly recommended.

5.9.4.3 Surface height matrix (ZZ records)

Following SURFMT, there shall be *nrows* lines of ZZ records, consisting of "ZZ=", each containing *ncolumns* of numbers, separated by semi-colons (";"), followed by a record terminator. Each numeric value represents the surface height at that point in the array. Subject to the minimum requirements specified in section 5.9.4.5, unknown or undefined points may be represented by "?". The 80-character line length limit shall not apply to ZZ records. Leading spaces are allowed, but not required, in numeric entries.

ZZ=?;?;?;…. 2.12345; 2.34567; 2.45678; …; 2.35678; 2.23567;?;?;?

Data points are provided as the Z coordinates in a regular XY grid. A right-handed coordinate system is used. The X-axis is positive to the right, the Y-axis is positive into the distance (away) and the Z-axis is positive upwards, all with reference to the cutting-plane, as shown in Figure 1.

The center point of the matrix shall have a Z-value equal to zero, and is deemed to lie in the plane of the surface matrix. Positive “ZZ” values indicate points above the plane of the surface matrix; negative values indicate points below the plane of the surface matrix. The specified center thickness (record “CTHICK”) is invariably obtained at the Prism Reference Point; GTHK is obtained at the center of the matrix (which may be distinct from the Prism Reference Point, if the PRP is decentered in the matrix as indicated in records SMOCIN/UP, which specifies the location of the PRP in the surface matrix).

The rotational orientation of the surface matrix shall be such that the X axis shall align with the horizon in the finished product, and with the blocking meridian determined by the location of fining centers or alignment grooves (which are traditionally aligned with the “base curve” in conventional surfacing).

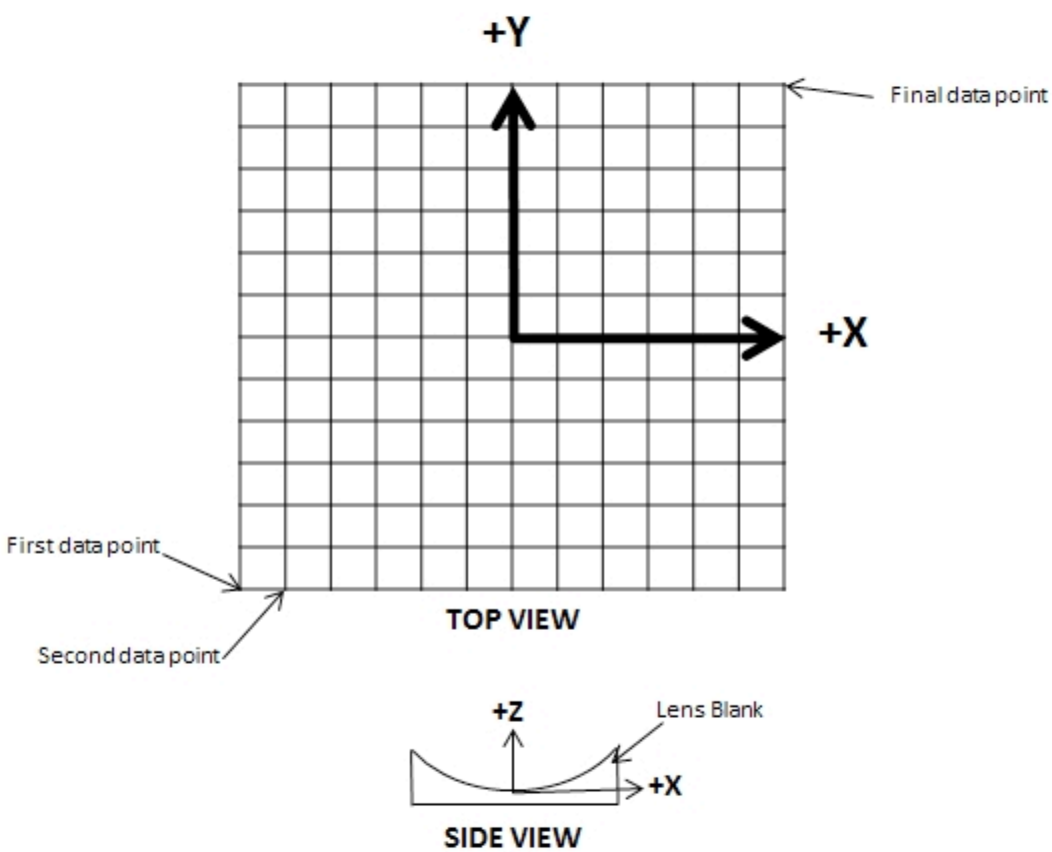

Figure A.8, showing surface matrix orientation

5.9.4.4 Slope data

When the slope format specifier is “1”, slope data appears immediate after the surface height (“ZZ”) records, in the following format.

Three additional records are included, representing (1) the x-components of the slopes of the normals along the sides of the matrix (“*dzdx*” values); (2) the y-components of the slopes of the normals along the top and bottom of the matrix (“*dzdy*” values); and (3) “mixed derivatives” as described below (“*dzdxdy*” values), representing the slopes of the normals at the corners of the matrix.

First, a number of *dzdy* fields equal to twice the number of columns in the matrix, separated by semicolons, appear in a record labeled “DZY”. The first value applies to the point (0,0) (the bottom left corner of the matrix), the next to the point (1,0), proceding rightwards to the point (ncolumns-1,0) (the bottom right corner of the matrix); the next field contains the *dzdy* value for the point at (0,nrows-1) (the top left corner of the matrix), the next to the point (1,nrows-1), proceding rightwards to the point (ncolumns-1,nrows-1) (the top right corner of the matrix).

Next, a number of *dzdx* fields equal to twice the number of rows in the matrix, separated by semicolons, appear in a record labeled “DZX”. The first value applies to the point (0,0) (the bottom left corner of the matrix), the next to the point (0,1), proceding upwards to the point (0,nrows-1) (the top left corner of the matrix); the next field contains the *dzdx* value for the point at (ncolumns-1,0) (the bottom right corner of the matrix), the next to the point (ncolumns-1, 1), proceding upwards to the point (ncolumns-1,nrows-1) (the top right corner of the matrix).

Last, four values, separated by semi-colons, appear in a record labeled “DZXY”. These values are computed as follows:

Given h as the smallest possible step length (e.g. 0.01mm) the mixed derivatives are calculated from the surface matrix for the positional values of (x,y) at (0,0), (0,nrows-1), (ncolumns-1,0), (ncolumns-1,nrows-1):

$f_x = ((dz/dx(x,y+h) - dz/dx(x,y-h)) + (dz/dy(x+h,y) - dz/dy(x-h,y)) / 4h$

The DZXY field values appear in the order (0,0); (0,nrows-1); (ncolumns-1,0); (ncolumns-1,nrows-1), i.e., bottom left, top left, bottom right, top right.

5.9.4.5 Minimum requirements for SURFMT data

This section specifies certain baseline characteristics of the surface matrix expressed in SDF files. Lens design systems may deviate from these baseline characteristics when appropriate for the context in which the data is to be used (specifically, when used in conjunction with certain devices). However, lens design systems must be capable of meeting the requirements specified in this section. A lens designer, therefore, *must* be able to produce a properly sized, fully populated, square matrix, with slope data; it *may* offer optional deviations from these requirements to improve efficiency when working with particular hosts and devices.

5.9.4.5.1 Size and shape

The minimum physical size expressed by the surface height matrix is determined by the crib diameter (record label “CRIB”) if present in the LDS data. If the crib diameter is not present in the LDS data, the minimum logical size of the surface definition matrix is determined by the lens blank diameter (record label “DIA”). The matrix must be no fewer than four points larger, and no less than two millimeters larger, than the crib diameter or blank diameter, whichever applies. The size and resolution (that is, the number of points) of the matrix shall be the same in both the horizontal and vertical dimensions. The maximum size of the matrix is not specified, but size in excess of the minimum is unnecessary and should be avoided.

5.9.4.5.2 Completeness

The entire surface matrix shall be populated with valid data, unless valid data cannot be calculated across the entire matrix, in which case, the unknown data indicator shall be used to indicate indeterminate surface heights.

Whenever slope data is included in a dataset, the matrix must be fully populated.

If indeterminate points are present at the periphery, and slope data is included, the indeterminate slope data will likewise be indicated by the unknown data indicator.

EXAMPLE A fragment of a SURFMT dataset:

```
SURFMT=1;R;B;141;141;70.0;70.0;1
ZZ=2.12345; 2.34567; 2.45678; ... 2.12345; 2.34567; 2.45678; 2.35678; 2.23567
```

```
ZZ=2.12345; 2.34567; 2.45678; ... 2.12345; 2.34567; 2.45678; 2.35678; 2.23567
ZZ=2.12345; 2.34567; 2.45678; ... 2.12345; 2.34567; 2.45678; 2.35678; 2.23567
...
ZZ=2.12345; 2.34567; 2.45678; ... 2.12345; 2.34567; 2.45678; 2.35678; 2.23567
DZY=0.317972; 0.317972; 0.317972; ... -0.303119; -0.303022; -0.303119; -0.303409
DZX=0.317972; 0.317972; 0.317972; ... -0.303119; -0.303022; -0.303119; -0.303409
DZXY=-0.000923818; -0.000923818; 0.000923818; 0.000923818
```

5.9.5 Low-Resolution Surface Definition Datasets

5.9.5.1 In addition to SDF data provided for manufacturing, LDSs may produce low-resolution datasets suitable for rendering surfaces on graphical displays, or for machines that do not require higher-resolution data. Such datasets shall follow the rules for SDF files, with the following exceptions:

5.9.5.2 The datasets shall use a Request Type of "SDL."

5.9.5.3 The path to the low-resolution dataset shall be specified in an SDLPATH record, which shall have only the simple, original form of LDPATH; that is, machine-specfic versions of SDLPATH are not allowed.

5.9.5.4 LMSs may specify to LDSs whether SDL files should be produced using the DOSDL record.

5.9.5.5 When SDFMODE specifies a manner of delivery other than "FILE" (see below), SDL data shall be delivered in the same manner as SDF data.

5.9.6 Surface Definition Dataset delivery

The method by which SDF data is delivered to a device may be specified in an SDFMODE record.

The default method is delivery via a file written to a network resource shared by the LDS software and the device, in which case the SDFMODE record may be omitted, or if present, will contain the literal value "FILE". The LDPATH record contains a fully qualified path name to the shared resource.

The second delivery method is indicated by an SDFMODE value of "LMS", in which case, the SDF data is included in the data packet delivered to a direct surfacing device. In this case, the LDPATH record contains a fully qualified path name identifying a file written to a network resource shared by the LDS and the host.

The third delivery method is indicated by a web address beginning with http://, in which case, the LDPATH contains a text value to be used in a method call to a web service from which the SDF data is retrieved. Such method calls are not standardized in this version of the Data Communications Standard.

The fourth delivery method is indicated by an ftp address beginning with ftp://, in which case, the LDPATH contains a name identifying a file that can be retrieved from the ftp site specified.

5.9.7 Encrypted Surface Definition Datasets

5.9.7.1 Surface definition datasets may be encrypted. The method for encrypting the dataset is not defined in this standard; however, the delivery of encrypted data is supported through the use of the LDPATH and LDCRYPT records. When the first field of an LDPATH record contains a single integer value, that value is used to link the LDPATH record to an LDCRYPT record. This form of LDPATH may be device-specific (it always has a device-specific form, but the device-identifying fields can be empty, see 5.9.3.3).

5.9.7.2 Encrypted datasets are delivered in one of two ways:

5.9.7.2.1 Method 1, which is by means of a network connection established between a device (typically an FSG) and an LDS, established according to the specification herein for establishing network connections (see 7.8.2); however, the autoconnect mode shall not be available for device-to-LDS connections; or

5.9.7.2.2 Method 2, by the inclusion of the encrypted data in a data packet transmitted from an LMS to a device.

5.9.7.2.3 The LDS controls which of the two methods is used by specifying an appropriate value in an LDCRYPT record (see below).

5.9.7.3 Communications sessions between devices and LDSs have exactly the same form and observe the same rules as communications sessions between download devices and LMSs. Additionally, devices establishing such connections must establish a TLS channel over the network connection, upon which:

1. The LDS will authenticate itself to the device;

2. Optionally, the device may then authenticate itself to the LDS using a previously-provided certificate;

3. The device may then conduct standard communications sessions with the LDS. In its simplest form, this would entail FSG, FSP, and ENG requests (or requests so initialized) from the DST-capable devices to the LDS, made after initial requests to the LMS. The delivery by the LMS of a properly-formed LDCRYPT record indicates to the device whether the request to the LDS is required.

5.9.7.4 The LDCRYPT record has two forms. The first is used only in packets transmitted from an LDS to an LMS. The second is used in packets transmitted to a device, whether transmitted to the device from an LMS or an LDS. The difference between the two forms is that the first form contains an additional first field that is used to link the LDCRYPT record with an LDPATH record appearing in the same packet, containing the same value in its first field. This makes it possible for multiple encrypted data records, for different machines, to be sent from the LDS to the LMS.

5.9.7.4.1 In the case of the first form, which is used only in packets transmitted from an LDS to an LMS, the LDCRYPT record contains two fields.

5.9.7.4.1.1 The first field contains an integer value that links the LDCRYPT record with an LDPATH record containing the same integer value (see, section 5.9.3.3).

5.9.7.4.1.2 The second field contains an integer value that identifies the encryption method:

0: No encryption

1: User-defined encryption

2: TLS (the encrypted data will be delivered by means of a request from a device to an LDS over a secure socket connection).

3: S/MIME (the encrypted data is contained in the data transmitted from the LDS to the LMS and will be delivered to the device in the data packet transmitted from the LMS to the device).

5.9.7.4.1.3 When the second field contains a value of three (3), an EDATA record shall follow the LDCRYPT record (see section 5.9.7.5).

5.9.7.4.2 The second form of LDCRYPT is used in response packets transmitted from an LMS or LDS to a device, and contains a single field, which corresponds to the second field sent from the LDS to the LMS.

5.9.7.4.2.1 A field value of zero in the LDCRYPT record indicates that there is no encrypted data record (EDATA, see section 5.9.7.5) in the packet.

5.9.7.4.2.2 A field value of 1 indicates a user-defined encryption method, which must be known to both sender and recipient in order to be usable, and which is not defined in this standard.

5.9.7.4.2.3 When the field value of the LDCRYPT record is two (2):

5.9.7.4.2.3.1 In a response packet transmitted to a device by an LMS, the device shall obtain encrypted data directly from an LDS as described in section 5.9.7.3.

5.9.7.4.2.3.2 In a response packet transmitted to a device by an LDS, the LDCRYPT record shall be followed by an EDATA record as described in section 5.9.7.5.

5.9.7.5 The EDATA record contains three fields:

5.9.7.5.1 The first field contains a literal that identifies the kind of encryption used. The literals are listed in Annex A.

5.9.7.5.2 The second field contains the number of bytes in the encrypted data.

5.9.7.5.3 The third field contains the encrypted data. The encrypted data shall be base64-encoded and shall therefore consist solely of printable ASCII characters.

5.10 Cribbing and crib datasets

5.10.1 Cribbing is the operation of reducing the diameter of the semi-finished blank, while affixed to the surface block. The simplest expression of cribbing data is the CRIB record, which may specify either the ending lens diameter (if greater than zero) or the amount by which the lens diameter is to be reduced (if less than zero). This results in a circular crib, centered at the surface blocking center. A CRIB record with a field value of exactly zero indicates that no crib operation should be performed.

5.10.2 The ELLH record may be used in conjunction with CRIB to specify an elliptically-cribbed lens. In this case, the field values in CRIB indicated the horizontal dimension of the ellipse and ELLH, the vertical. The major axis of the ellipse must be oriented rotationally on the surface block so as to align with the datum line of the frame.

5.10.3 The records FLATA and FLATB may be used to indicate dimensions to which the lens blank should be truncated, horizontally and vertically. In this case, the effective radius from the surface block center to the cribbed blank edge in any particular meridian will be the minimum of the radius to the rectangle of width FLATA and height FLATB, or to one half the value of CRIB.

5.10.4 A crib shape other than circular, elliptical, or truncated, may be indicated using a crib dataset, which consists of a CRIBFMT record followed R, and, in the case of non-equiangular data, A records. The form of the CRIBFMT record is identical to TRCFMT, discussed above (section 5.4). In this method, the crib shape is expressed exactly as a tracing might be, with the following exceptions.

5.10.4.1 Radii (and angles if present) have their origin at the surface block center.

5.10.4.2 Crib datasets do not include "Z" (sag) data.

5.10.4.3 The CRIBFMT record comprises the first four of the five fields specified for TRCFMT; the last field in TRCFMT specifies what kind of object was traced (frame, pattern, or demo lens), and is therefore inapposite in this context.

5.10.5 CRIBFMT negotiation occurs in the same fashion as for TRCFMT (see section 5.4). As for TRCFMT, in the absence of initialization, a device specifies one or more CRIBFMT records in its request, in order of preference.

EXAMPLE A crib dataset expressed in format #1, ASCII absolute.

```
CRIBFMT=1;40;E;R<CR/LF>
R=2479;2583;2605;2527;2394;2253;2137;2044;1975;1935<CR/LF>
R=1922;1939;1989;2072;2184;2322;2471;2599;2645;2579<CR/LF>
R=2517;2450;2379;2318;2247;2168;2086;2014;1958;1923<CR/LF>
R=1909;1914;1941;1983;2033;2089;2140;2200;2277;2371<CR/LF>
```

5.11 Surface Thickness datasets

Surface thickness datasets describe lens thickness after surfacing at a number of radii centered on the surface block. One important use of this dataset is to indicate areas around the perimeter of a lens where the thickness approaches zero (a "knife-edge").

5.11.1 A surface thickness dataset consists of a STHKFMT record followed by R, optionally A, and (mandatorily) T records.

5.11.2 "T" records indicate the thickness of the lens at the corresponding radius (and angle, which is either implicit in mode "E", or explicit in mode "A"), in units of 0.01mm.

5.11.3 STHKFMT negotiation occurs in the same fashion as for TRCFMT (see section 5.4). As for TRCFMT, in the absence of initialization, a device specifies one or more STHKFMT records in its request, in order of preference.

EXAMPLE A surface thickness dataset expressed in format #1, ASCII absolute.

```
STHKFMT=1;40;E;R<CR/LF>
```

```
R=3000;3000;3000;3000;3000;3000;3000;3000;3000;3000<CR/LF>
R=3000;3000;2990;2985;2980;2980;2985;2990;2998;3000<CR/LF>
R=3000;3000;3000;3000;3000;3000;3000;3000;3000;3000<CR/LF>
R=3000;3000;2990;2985;2980;2980;2985;2990;2998;3000<CR/LF>
T=60;80;120;140;150;180;140;120;110;96<CR/LF>
T=90;50;0;0;0;0;0;0;50<CR/LF>
T=60;80;120;140;150;180;140;120;110;96<CR/LF>
T=90;50;0;0;0;0;0;0;50<CR/LF>
```

5.12 Power Map Datasets

5.12.1 Power Map datasets are contained in files which have an extension “PMF” and which contain a request type record having the value “PMF”. The Power Map describes the surface power (front or back) or the through power for multiple points of the lens. Note that the surface power is an absolute geometrical quantity unlike the through power that depends on the chosen measurement method.

5.12.2 PMFMT record. Power Map datasets begin with a PMFFMT record having the following format:

PMFMT=*format*;R|L;B|F|T;D|C|A;M|T|E;ncolumns;nrows;width;height[;index; method]

5.12.2.1 The first field indicates the format of the following data. Current support is for one standard format, “1”, described below.

5.12.2.2 The second field contains R or L, indicating to which eye the following data applies.

5.12.2.3 The third field contains B, F, or T indicating the measured power type: front surface power (F), back surface power (B), or through power (T).

5.12.2.4 The fourth field contains D, C, or A indicating the power quantity: spherical equivalent power (D), cylinder power (C), or cylinder axis (A).

5.12.2.5 The fifth field contains M, T, or E indicating whether it is measured power (M), theoretical power (T) or the error power (E).

5.12.2.6 The sixth and seventh fields contain two integers specifying the number of columns (ncolumns) and rows (nrows) respectively in the matrix of PP records which follows (see below).

5.12.2.7 The eighth and ninth fields contain the vertical and horizontal size of the matrix.

5.12.2.8 The tenth, optional field, is the index of refraction used for the surface power (power = (index-1)/radius). The index is only needed for surface power (third field = F or B).

5.12.2.9 The eleventh, optional field, is the measurement method F (focus on axis) or I (infinity on axis), which is only meaningful in case of through power (fifth field = T). It indicates the chosen measurement method for the through power.

EXAMPLE

PMFMT=1;R;T;D;M;141;141;80.0;80.0

would therefore indicate that the ensuing matrix represents a rectangular array of points, 141 rows and 141 columns, representing data points that are approximately 0.571 mm apart horizontally and 0.571 mm apart vertically (80.0 / (141.0-1.0)). Each point would specify a through power measure of spherical equivalent power.

5.12.3 Power map matrix (PP records)

5.12.3.1 Following PMFMT, there shall be nrows lines of PP records, consisting of “PP=”, each containing ncolumns of numbers, separated by semi-colons (“;”), followed by a record terminator. Each numeric value represents the value specified by fields 3, 4 and 5 at that point in the array. Subject to the minimum requirements specified in section

5.9.4.5, unknown or undefined points may be represented by “?”. The 80-character line length limit shall not apply to PP records. Leading spaces are allowed, but not required, in numeric entries.

EXAMPLE

PP=?;?;?;…. 2.12345; 2.34567; 2.45678; …; 2.35678; 2.23567;?;?;?

5.12.3.2 Data points are provided as the power value in a regular Cartesian grid. A right-handed coordinate system with the PRP as origin is used. The X-axis is positive to the right, the Y-axis is positive to the top. The first point in the first row represents the point at the lower left-hand corner of the matrix. Points are ordered from left to right, bottom to top.

It should be noted that, if a point is desired to be located at the PRP, an odd number of rows and columns must be specified (see below). Therefore, an odd number of points for both X and Y is strongly recommended.

The name of a file containing a power map dataset is specified in an INSPATH record, which follows the rules of construction defined for SDF files (see 5.9.4).

5.13 Weighting Matrix Datasets

5.13.1 Weighting Matrix datasets are contained in files which have an extension “WDF” and which contain a request type record having the value “WDF”. The Weighting Matrix dataset is used for the inspection process. It defines the regions of interest as well as the relative importance of the different points of these regions in order to calculate global errors for quality control. WDF files can be the output of an LDS but can also be produced by an inspection device.

5.13.2 WMFMT record. Weighting Matrix datasets begin with a WMFFMT record having the following format:

WMFMT=*format*;R|L;index;ncolumns;nrows;width;height

5.13.2.1 The first field indicates the format of the following data. Current support is for one standard format, “1” described below.

5.13.2.2 The second field contains R or L, indicating to which eye the following data applies.

5.13.2.3 The third field is the index of the weighting matrix, starting with 0. The “index” identifies a particular matrix; several different matrices can be used, each representing a different error calculation.

5.13.2.4 The fourth and fifth fields each contain an integer specifying the number of columns (ncolumns) and rows (nrows) respectively in the matrix of WW records which follows (see below).

5.13.2.5 The sixth and seventh fields contain the vertical and horizontal size of the matrix respectively.

EXAMPLE

WMFMT=1;R;0;141;141;80.0;80.

would represent a rectangular array of points, 141 rows and 141 columns, representing data points that are approximately 0.571 mm apart horizontally and 0.571 mm apart vertically (80.0 / (141.0-1.0)).

5.13.3 Weighting matrix (WW records)

5.13.3.1 Following WMFMT, there shall be nrows lines of WW records, consisting of "WW=", each containing ncolumns of numbers, separated by semi-colons (";"), followed by a record terminator. Each numeric value represents the weighting factor value at that point in the array. The weighting factor value represents the relative importance of the subject point in the array in determining whether to pass or fail a lens. The value shall be in the range 0..1, with 0 specifying the least importance and 1, the greatest Subject to the minimum requirements specified in section 5.6.4.3, unknown or undefined points may be represented by "?". The 80-character line length limit shall not apply to WW records. Leading spaces are allowed, but not required, in numeric entries.

5.13.3.2 EXAMPLE

WW=?;?;?;…. 2.12345; 2.34567; 2.45678; …; 2.35678; 2.23567;?;?;?

5.13.3.3 Data points are provided as the weighting factor value in a regular XY grid. A right-handed coordinate system with the PRP as origin is used. The X-axis is positive to the right, the Y-axis is positive into the distance (away); all with reference to the tangent plane at the PRP, as shown in Figure A.8.The first point in the first row represents the point at the lower left-hand corner of the matrix. Points are ordered from left to right, bottom to top.

5.13.3.4 It should be noted that, if a point is desired to be located at the PRP, an odd number of rows and columns must be specified (see below). Therefore, an odd number of points for both X and Y is strongly recommended.

5.13.3.5 The name of a file containing a power map dataset is specified in a WDFPATH record, which follows the rules of construction defined for SDF files.

5.14 Design Deviation Datasets

5.14.1 Design deviation datasets are contained in files which have an extension "DDF" and which contain a request type record having the value "DDF". The design deviation dataset is used for the inspection process or for the process control. It defines the various global or local errors calculated on the surface for quality control. The DDF files can be the output of an LMS or LDS, but are generally produced by an inspection device.

5.14.1.1 The name of a file containing a power map dataset is specified in a DDFPATH record, which follows the rules of construction defined for SDF files.

5.14.2 DDFMT record

The design deviation datasets begin with a DDFMT record having the following format:

DDFMT=format;RIL;count

The first field indicates the format of the following data. Current support is for one standard format, "1" described below.

5.14.2.1 The second field contains R or L, indicating to which eye the following data applies.

5.14.2.2 The third field indicates the number of error map values given in the set.

5.14.3 DD record

Following DDFMT, there shall be count lines of DD records having the following format

DD=ID;BIFIT;errorname; tolrslt [;errorvalue;tol;tobedisplayed]

5.14.3.1 The first field is the ID of the Error Type, starting with 0. The ID identifies a particular design deviation.

5.14.3.2 The second field contains B, F, or T indicating the measured power type used to calculate the deviation: front surface power (F), back surface power (B), or through power (T).

5.14.3.3 The third field contains the name given to the deviation.

5.14.3.4 The fourth field contains the result of the tolerance control for the error type (0- failed; 1- passed; 9 – not tested).

5.14.3.5 The fifth field contains the amplitude of the deviation.

5.14.3.6 The sixth field contains the tolerance limit.

5.14.3.7 The seventh field indicates whether or not the Error Map Value needs to be displayed on the inspection instrument (0-not to be diplayed; 1-to be displayed). The default value is 1.

5.14.3.8 If an optional field is present, all the optional fields on its left should also be present.

5.14.3.9 If the second field is equal to B or F, the error value is given in TIND diopters.

Example :

```
DDFMT=1;R;4
DD=0;T;Distance;1;0.03;0.12
DD=1;T;Near;1;0.05;0.12
DD=2;T;Intermediate;0;0.18;0.15
DD=3;T;Lateral;1;0.12;0.20

DDFMT=1;L;4
DD=0;T;Distance;1;0.05;0.12
DD=1;T;Near;1;0.02;0.12
DD=2;T;Intermediate;1;0.13;0.15
DD=3;T;Lateral;0;0.27;0.20
```

5.15 Marking Records

In version 3.06 of this standard, records INKMASK, ENGMASK and ENGLOC were included to allow for the specification of a “picture” to be produced on a lens (probably a direct-surfaced one); the first leaving visible mark(s), and the latter two, semi-visible mark(s). The resource containing the picture was expected to reside on the marking device. In version 3.07, these records are deprecated in favor of the ENGMARK record, which allow multiple features to be drawn on either surface of a lens, including the picture that would be specified in ENGMASK or INKMASK in the prior version of the standard.

NOTE The terms “engraving” and “engraver” are used broadly and colloquially and are intended to refer to machines that put marks on lens surfaces by any means, not limited to those that do so by a process of engraving.

5.15.1 **ENGMARK.** The ENGMARK record contains up to thirteen fields. The first two fields are mandatory (in that the record would not be meaningful without them); the necessity of the balance of the fields depends on the “kind” of feature indicated by the first field. Multiple ENGMARK records may appear in a packet in order to specify multiple features which may or may not differ in “kind” (see 5.15.1.1.). ENGMARK can specify the location of semi-visible marks on one or both surfaces of a lens or pair of lenses.

5.15.1.1 The first field specifies the kind of feature, and is a literal value from the following enumeration:

5.15.1.2 **MASK**, in which case the second field contains an identifier, which specifies a picture stored on the marking device. Under version 3.06, this identifier would have appeared in the ENGMASK record.

5.15.1.3 **DCS**, in which case the second field contains a DCS record label, which must be included in the data packet that contains the ENGMARK record.

FILE, in which case the second field contains a fully-qualified file name of an appropriate type that contains a picture or image that is to be applied to the lens by the device.

TXT,in which case the second field contains text to be marked on the lens by the device, and the third field optionally contains an identifier which specifies the font to be used

OBJECT, in which case the second field contains an object name and the third field contains the content to be utilized by the object specified by the name.

5.15.1.4 The content of the second field depends on that of the first, as described above.

5.15.1.5 The content of the third field depends on that of the first, as described above.

5.15.1.6 The fourth field contains a single character indicating the lens, right (“R”), left (“L”), or both (“B”), on which the feature is to be produced.

5.15.1.7 The fifth field contains a single character indicating the side of the lens, front (“F”) or back (“B”), on which the feature is to be produced.

5.15.1.8 The sixth field contains a single character indicating the side of the lens, front (“F”) or back (“B”), indicating whether the feature is to be properly oriented from the perspective of a person looking at the indicated surface of the lens.

5.15.1.9 The seventh field contains the x-coordinate of the feature.

5.15.1.10 The eighth field contains the y-coordinate of the feature.

5.15.1.11 The ninth field contains the z-coordinate of the feature.

5.15.1.12 The tenth field contains a numeric value from zero to ten indicating the visibililty of the feature, with zero being the least, and ten the most visible.

5.15.1.13 The eleventh field contains a numeric value indicating the height of the feature in millimeters.

5.15.1.14 The twelfth field contains a numeric value indicating the rotation of the feature in degrees.

5.15.1.15 The thirteenth field contains a TEXT value indicating the font to be used in producing the feature. This is applicable to kinds DCS and TXT.

5.15.1.16 The fourteenth field contains a TEXT value specifying the color of the visible medium used to produce the mark(s). This field is only used when visible (usually, “inked” or “painted”).

5.15.1.17 The fifteenth field contains a numeric value specifying the width of the feature in millimeters.

5.15.1.18 The sixteenth field contains a LITERAL value specifying whether the feature is to be engraved (‘ENG’) or painted (‘INK’) on the surface.

5.15.2 The coordinates specified for a feature refer to the center of that feature. For an irregularly-shaped feature, the coordinates refer to the center of an implied rectangle circumscribing the feature.

5.16 Device-Specific Records

Certain records defined herein may be identified as “device-specific”, in which case fields will be defined in the record to identify the device type (DEV), device vendor (VEN), device model (MODEL), and in some cases, device serial number (SN). Examples of these include LDPATH and PROC. When device-specific records are communicated to devices requesting job data, the records do not include the fields that identify the device. For a further example of how this works, see section 5.9.3.2.

5.17 PROC Records

Records having the label “PROC” are used for two purposes. The first is to allow hosts and devices to convey process-related data, in which case, the PROC record contains a label (which may be standard or proprietary) that may be specified by a device during initialization. The second is to allow any DCS record to be treated as Device Specific.

5.17.1.1 EXAMPLE

A record uploaded from an LDS in an LMS data packet like the following:

PROC=_P1PROC;POL;IGX;TURBO;010702250607000000;010702250607000000

… would be delivered to a device identifying itself as DEV=POL, VEN= IGX, MODEL=TURBO requesting a _P1PROC record as:

_P1PROC=010702250607000000;010702250607000000

Similarly, an LDS can upload an LMATID for a specific generator (only) as follows:

PROC=LMATID;FSG; IGX; MARKX;62;62

So that upon a request from a device identifying itself as DEV=FSG, VEN= IGX, MODEL=MARKX, assuming that it has requested LMATID in its initialization, it would receive:

LMATID=62;62

5.18 TOKEN Records

The TOKEN record may be used by hosts and Lens Design Systems to facilitate the management of transactions between the two, which principally consist of charges by the LDS for lens designs. TOKEN can help identify orders for which the LDS may not charge, or may charge a different rate. Using TOKEN to accomplish this is optional ; in the following sections, the use of the term « shall » shall mean that the usage described is mandatory only when the scheme is being used.

5.18.1 When a given order is first submitted by an LMS to an LDS, the LMS shall include a TOKEN record having an unknown value (e.g. TOKEN= ?). This indicates that the LMS expects to be charged for the order (assuming that DOSDF has a value other than « N »).

5.18.2 Upon returning a successful calculation to the LMS, the LDS shall populate the TOKEN record with a unique identifer that shall then be used to identify the order in question to the LDS for any and all subsequent communications regarding the order in question. A TOKEN field value may consist of a GUID or other unique identifier ; responsibility for its uniqueness lies with the producer of it, which is the LDS.

5.18.3 The LMS shall include the TOKEN record in any and all subsequent submissions to the LDS for the order in question. An XSTATUS code should be included to indicate the reason for the resubmission. The inclusion of the TOKEN, together with status information as described below, may cause the LDS to process the order without further charges.

5.18.4 On a resubmission of an order, the LMS may include a BILLOK record to indicate whether or not the LDS should proceed with the order even if the LDS will bill for the requested designs. BILLOK=0, or the absence of a BILLOK record, shall mean that the LDS must not proceed with the order if doing so will result in an additional charge. In that case, if the LDS would charge for the designs, the LDS shall not process the order and shall return STATUS=20 to the LMS. The LMS may then resubmit the order with the same TOKEN, together with a BILLOK record having a value of « 1 », which shall indicate to the LDS that it should proceed the order and charge for it

5.18.5 The LMS may include a BILLOK=1 record in any resubmission in order to prevent its receiving a rejection of the order due to the submission of a TOKEN that does not afford the LMS a « free » design.

5.18.6 When returning its results to an LMS, an LDS may include a CLICKS having a value of R, L, B, or N, indicating whether it has recorded a « click » charge for the right, left, both, or neither lens(es), on that and only that sunbmission of the order.

5.19 Job Status Notification

An LMS may notify an LDS of the statuses of orders. The primary purpose of this is to allow the LDS to know when an order has been completed so that it may defer billing for the designs until at least such time as the order has been shipped by the lab.

5.19.1 Status notification is accomplished through the use of files (or such other mechanisms as may be developed to implement communications between hosts and Lens Design Systems) containing the request type INF, together with the following records : JOB, TOKEN, TIME, and XSTATUS.

5.19.2 When job status notifications are included in files sent to Lens Design Systems, files shall have an « LDS » extension.

5.19.3 Standard XSTATUS codes indicate job statuses. See Table A.26.

EXAMPLE

Contents of an LDS file containing a job status notification:

REQ=INF

JOB=1234

TOKEN=UNIQUETOKENFORTHISORDERSUPPLIEDBYLDS

TIME=2013030301220000

XSTATUS=B;100;Job completed.

5.20 Records with Multiple-Value Fields

Beginning with version 3.10 of this standard, certain records may be defined as consisting of chiral data, comprising fields for right- and left-eye data, which fields are further divided into two or more sub-fields. Sub-fields are delimited by the pipe character (« | »). An example of multiple-value fields can be found in the section on tolerance values, below.

5.21 Tolerance Records

5.21.1 Tolerance records, all of which begin with the letters « TOL », are listed in Table A.1a. « TOL » records are used to indicate whether a given value (specified by the remainder of the record label) is within tolerance according to a measuring device.

5.22.2 Tolerance records may have corresponding « value » records, which begin with the letters « TOLV », and which in most cases comprise multiple-value fields. Each field contains two values ; the first value expresses the lower bound of the range of allowable values, and the second value is the upper bound of the range of allowable values

6 Packets

6.1.1 Order of records in packets

For records other than those in datasets, order is not important, except as specified in this section. In all packets, the REQ or ANS record shall appear first. The JOB record shall immediately follow the REQ or ANS record in non-initialization packets. When present, the CRC record must be last, following the <RS> character. Sequencing of records within datasets is specified in the section in which the dataset is defined.

6.1.2 CRC record

6.1.2.1 Hosts and devices may include, as the last record in any packet, a record used to validate the contents of the packet, as described in this section. Hosts and devices should include a CRC record in responses to any packet that includes a CRC record. However, a device or host that cannot calculate a CRC may simply ignore the CRC record and is not required to send a CRC. A device or host that receives a packet that does not contain a CRC record value must accept the packet without returning an error message because of such absence.

6.1.2.2 The CRC record will consist of a record label (CRC), a label separator, an unsigned integer data field, and a record separator.

6.1.2.3 The CRC field value is an unsigned integer calculated according to the description in annex C.

6.1.2.4 When the CRC record is included in a packet, it appears immediately after the <RS> character. The <RS> character shall occur immediately after the record separator for the prior record, which would otherwise be the last record in the packet, and immediately before the record label for the CRC record. When no CRC record is included, the <RS> character shall appear immediately prior to the <GS> character.

6.1.2.5 The CRC value will be calculated on all of the characters in the message after (but not including) the start character, up to and including the <RS> character immediately prior to the record label of the CRC record.

EXAMPLE The part of a packet used in the CRC calculation. The underlined data is used to calculate the CRC.

<FS>REQ=INI<CR><LF>

```
DEV=GEN<CR><LF>
VEN=IGC<CR><LF>
MODEL=BLASTER1<CR><LF>
MID=IGC12345<CR><LF>
<RS>
CRC=51242<CR/LF>
<GS>
```

6.1.3 Confirmation and timeouts

6.1.3.1 The confirmation message consists of a single character, either an ASCII ACK (decimal 06) or an ASCII NAK (decimal 21). Whenever a packet is received, the receiver must transmit an ACK to the sender to indicate its packet was received correctly (a positive acknowledgement). If the receiver believes that the packet was not received correctly, it should transmit a NAK to the sender (a negative acknowledgement). The sender should retry its transmission up to three times (a total of four transmissions answered by NAK) before giving up and posting an appropriate error to the equipment operator.

EXAMPLE A failed session. When a receiver cannot recognize a message or finds an invalid CRC in a packet, it sends a negative acknowledgement to the sender, which retries up to three times.

```
DEVICE                                  HOST
<FS>REQ=12<CR/LF>
JOB=1234<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                                        <NAK>
<FS>REQ=12<CR/LF>
JOB=1234<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                                        <NAK>
<FS>REQ=12<CR/LF>
JOB=1234<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                                        <NAK>
<FS>REQ=12<CR/LF>
JOB=1234<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                                        <NAK>
```

NOTE After the fourth negative acknowledgement, the device gives up and displays an error message. If the host’s response is other than ACK or NAK, the device might help lead a troubleshooter in the direction of port parameters by displaying a message such as "Unrecognized confirmation“.

6.1.3.2 Confirmation timeout: A sender of a packet shall wait up to six seconds for a confirmation message. If six seconds elapse after transmission of a packet without receipt of a confirmation, the session is aborted. The sender posts an appropriate error to the equipment operator and does not retry its transmission.

6.1.3.3 Packet timeout : When a sender transmits a packet, receives a positive acknowledgement, and further expects a packet from the receiver in return, the sender shall wait up to twelve seconds after its receipt of the positive acknowledgement for the receiver's packet to begin to arrive. Should the receiver's packet fail to begin to arrive within that period, the session is aborted. The sender posts an appropriate error to the equipment operator and does not retry its transmission.

6.1.3.4 Intercharacter timeout: When receipt of a packet pauses for more than five seconds prior to the receipt of the end character, the session is aborted and the receiver posts an appropriate error to the equipment operator.

6.1.3.5 The defined timeouts may be altered by a host during initialization. Timeouts are expressed in seconds. The range of timeouts shall be 2-255 seconds.

EXAMPLE : a device uploading a frame tracer (TRC) packet showing timeouts.

NOTE Up to 5 seconds can pass between characters (not shown).

DEVICE	HOST
<FS>REQ=TRC<CR/LF> **JOB=1234<CR/LF>** **(Optional Interface Records)** **<RS>** **CRC=12345<CR/LF>** **(The CRC record is optional; the number shown is not properly calculated).** **<GS>**	
up to six seconds pass…	
	<ACK>
up to twelve seconds pass…	
	<FS>ANS=TRC<CR/LF> **JOB=1234<CR/LF>** **STATUS=0<CR/LF>** **(any Optional Interface Records that must be echoed)** **<RS>** **CRC=12345<CR/LF>** **(The CRC record is optional; the number shown is not properly calculated).** **Optional<CR/LF>** **<GS>**
up to six seconds pass…	
<ACK>	
up to twelve seconds pass…	
<FS>ANS=TRC<CR/LF> **JOB=1234<CR/LF>** **(All Records for TRC Packet)** **<RS>** **CRC=12345<CR/LF>** **(The CRC record is optional; the number shown is not properly calculated).** **<GS>**	
up to six seconds pass…	

```
                                  <ACK>
up to twelve seconds pass…
                                  <FS>ANS=TRC<CR/LF>
                                  JOB=1234<CR/LF>
                                  STATUS=0<CR/LF>
                                  <RS>
                                  CRC=12345<CR/LF>
                                  (The CRC record is optional; the number shown is
                                  not properly calculated).
                                  <GS>
up to six seconds pass…
    <ACK>
```

6.2 Deprecated Requirements

6.2.1 Mandatory records

The need for mandatory records has been superseded by the addition of the unknown data indicator. Mandatory records may still be supported for backward compatibility, but should not be used in new implementations. Any records that are contained in a record label list or a pre-set packet should be sent using the unknown data indicator if necessary. Where the term "mandatory" now appears in this document, it has its normal meaning. The following paragraph contains the old explanation of mandatory records:

Records that are mandatory are so designated in their definitions. Records not so designated may be presumed to be optional.

6.2.2 Quotation marks

6.2.2.1 In versions prior to OMAV 3.03, quotation marks should be placed around limited and text data. Parsers should strip the quotes before using the enclosed data. Starting with OMAV 3.03, the quotation marks are no longer necessary, but implementations may include them for backward compatibility.

6.2.2.2 The presence or absence of quotation marks should not cause an implementation to break.

6.2.2.3 In versions prior to OMAV 3.03, when the unknown data indicator (?) is used to replace a limited or text data value, it should be quoted following the rules for the data value it replaces.

6.2.2.4 Quotation marks do not count toward text length limits since they simply enclose data values. They do count against the 80-character line limit.

6.2.3 DRILL record

The DRILL record introduced in OMAV 3.02 is deprecated in favor of the DRILLE record introduce in OMAV 3.04. See section 5.5.

6.2.4 Preset Initialization

As of version 3.04, Preset Initialization is deprecated for devices that use download sessions. It was originally believed that Preset Initialization would be of value as a mode of operation that would be less onerous to implement than Auto Initialization. However, in practice, the additional difficulty of implementing Auto Initialization over that of implementing Preset Initialization has been found to be insignificant.

Preset Initialization persists as an active specification for upload devices. The Record Label List is the feature that distinguishes Preset Initialization from Auto Initialization, and upload devices do not specify the set of records that they want to upload (instead, they simply upload them). Therefore, when upload devices perform initialization, they perform Preset Initialization.

In all other cases, hosts and devices that implement initialization must implement auto-format initialization.

While the implementation of initialization is not mandated for either hosts or devices, it is very strongly recommended. The “no-initialization” mode is intended to provide a more easily implemented mode of operation for use in development, testing, and light-duty applications.

7 Sessions

7.1 General

7.1.1 Sessions consist of an exchange of messages between a device and a host.

7.1.2 Devices initiate all sessions by transmitting a request packet to a host.

7.1.3 Once begun, a session shall be concluded before another session can be initiated between a host and a given device. Sessions are never interleaved.

7.1.4 The Standard establishes three categories of sessions: initialization, upload, and download. The detailed structure of each type of session is described in 6.2 to 6.4. A file-based information transfer variant using the FIL device type is described in 6.5.

7.1.5 When a host receives a packet while not engaged in a session with a device, it expects a request packet. If it receives a packet which has the overall form of a Standard packet, but which lacks a request type (REQ) record, the host shall use the literal data "ERR" in its answer type record, in the form "ANS=ERR", along with a status record with a status code of 18.

7.1.6 The job ID record must appear in all packets outside of initialization (and, pursuant to § 6.1.1, must be the second record in any packet). A device shall verify that the job ID received matches that specified in the request packet. Hosts shall return the job ID specified in the same format it was received. The job ID record does not appear in the TIM request or response packets.

7.2 Initialization sessions

7.2.1 General

There are two types of initialization: auto-format (also known as “auto-initialization”) and preset. Preset initialization is deprecated in this version of the Standard for download devices, and hosts and download devices that implement initialization should always implement auto-format initialization.

The purpose of initialization is to:

allow devices to provide information to hosts that remains constant between sessions;

provide a means by which hosts can identify devices and tailor their responses to them;

allow devices to negotiate with the host on various options within the interface, i.e. tracing formats and timeout values.

These features serve to minimize the amount of data that must be exchanged during subsequent sessions.

Auto-format initialization allows devices to define a set of records, from the device records defined in the Standard, which they wish to receive in data packets from the host in subsequent sessions. Upload devices should not use auto-format initialization unless they wish to receive data from the host in addition to uploading information to it. Preset initialization allows devices to identify themselves as a one of a set of preset device types defined in the Standard.

Both types of initialization provide for other identifying information to be conveyed which further allows hosts to tailor the information sent.

Devices that do not support initialization will use the preset device types in the REQ record (i.e. GEN, EDG, TRC, etc.); however, this is considered an “entry-level” or “limited” mode of operation. Initialization is preferred.

Devices capable of supporting initialization should attempt initialization when powered on. They should use the Request ID received from the host in subsequent REQ records. If a Host receives a numeric request which it does not understand, it should send a response packet containing a status record with a status code of 5 (Need initialization). Any device receiving a STATUS=5 from a host should attempt initialization. A device not capable of initialization should never receive a STATUS=5 from a host because it should never make a numeric request.

7.2.2 Composition of initialization sessions

Initialization sessions consist of the following exchange of messages.

a) A device transmits a request packet including a request type record with data INI to a host.

b) The host transmits a confirmation message to the device.

c) The host transmits a response packet to the device.

d) The device transmits a confirmation message to the host. If the host has sent a STATUS=15 (initialization not supported) record in its response packet (in step c), the session ends at this point.

e) The device transmits a data packet to the host.

f) The host transmits a confirmation message to the device.

g) The host transmits a data packet to the device.

h) The device transmits a confirmation message to the host.

Table 5 – Summary of initialization session

DEVICE transmission	HOST transmission
Request packet, REQ=<INI>	
	Confirmation <ACK>
	Response packet, ANS=<INI>
Confirmation <ACK>	
Data packet, ANS=<INI>	
	Confirmation <ACK>
	Data packet, ANS=<INI>
Confirmation<ACK>	

7.2.3 Elements common to auto-format and preset initialization

7.2.3.1 A device may effect auto-format initialization or preset initialization if the host's response packet contains a status code of 0. A device may only effect preset initialization if the host's response packet contains a status code of 14. The set of interface records that the device includes in its data packet may be the same in either case. These records contain data that serve to identify the device to the host. In the case of preset initialization, this comprises the entirety of the device's data packet.

7.2.3.2 A device may over-ride any of the parameters specified during initialization by including them in any subsequent request or data packet. The host should re-set the parameter to that specified during initialization for all sessions subsequent to the one in which the parameter was over-ridden. Should it be necessary for a device to change a parameter set in initialization permanently, it should issue a new initialization request.

7.2.3.3 Hosts that support auto-format or preset initialization must manage the assignment of request type identifiers so that the identifiers do not repeat except over long periods. This is required to avoid a situation that could arise when a host is restarted: it could assign a request ID which, prior to its being re-started, had been used by a different machine than the one to which it is assigned after the re-start.

7.2.3.4 A host may include the interface records HMODEL, HNAME, HSN, HVEN and/or HVER in the data packet it returns to the device.

7.2.3.5 Both types of initialization are initiated with the same request packet

EXAMPLE 1 An initialization request packet:

```
<FS>
REQ = INI
<RS>
<GS>
```

EXAMPLE 2 A response packet in which the host indicates its initialization capability:

```
<FS>
ANS = INI
STATUS = *initialization status
<RS>
<GS>
```

*initialization status can be 0, 14, or 15

EXAMPLE 3 - A portion of a device's initialization data packet containing interface records.

```
<FS>
ANS = INI
DEV = GEN
VEN = IGC
MODEL = B16
MNAME = Intergalactic Generator Co. Model 16
MID = QB485M
OPERID = JACK
<RS>
<GS>
```

7.2.4 Auto-format initialization

7.2.4.1 Upon receipt of a host's response packet with a status code of 0, a device may effect auto-format initialization by including a request definition in its data packet. The request definition consists of a record labeled DEF which contains a text field (called a device tag) that identifies the definition, followed by a variable number of record label list records, labeled "D", each of which contains a series of record labels. The maximum number of characters in a single record label list, between the label separator and the record separator, is eighty (80) characters. The minimum number of fields in a record label list is one; however, the preferred method of expression is to place the maximum number of fields in each record label list record without exceeding the 80-character maximum. This minimizes the number of record label list records required. The request definition ends with an ENDDEF record that is labeled with the device tag found in the DEF record. Process-control records are treated the same as data records in this context.

7.2.4.2 R, A, Z, and ZA labels should not appear in request definitions. The use of these labels is determined by TRCFMT and ZFMT that should be present as interface records in the initialization request.

7.2.4.3 If any case where BEVP is included in the record list, BEVM and BEVC shall also be included in the response if they are necessary.

7.2.4.4 A host will assign an integer request ID to each request definition set sent from the device. The request ID is returned to the device in the host's data packet. The device can then use this request ID when initiating sessions.

7.2.4.5 Interface record labels (e.g., DO, JOB, TRCFMT) should never appear in a record label list.

EXAMPLE 1 - A device's initialization data packet in which the device (in the case, a generator) effects auto-format initialization. The device tag is MYREQUEST.

```
<FS>
ANS = INI
DEV = GEN
VEN = IGC
MODEL = B16
MNAME = Intergalactic Generator Co. Model 16
MID = QB485M
OPERID = JACK
DEF = MYREQUEST
D = FRNT;BACK;GBASE;GCROS;GAX;GBASEX;GCROSX;SAGRD
D = SAGBD;SAGCD;TIND;RNGH;RNGD;GTHK;LMATID;LAPM
D = DIA;BCTHK;BETHK;CRIB;GPRVA;GPRVM;KPRVA;KPRVM;PIND
ENDDEF = MYREQUEST
<RS>
<GS>
```

EXAMPLE 2 - A host's initialization data packet in which the host assigns a request ID to the dataset defined in the device's data packet. The host assigns request ID 1643 to device tag MYREQUEST.

```
<FS>
ANS = INI
STATUS = 0
DEF = MYREQUEST;1643
REM = I have assigned '1643' to 'MYREQUEST'
<RS>
<GS>
```

EXAMPLE 3 - A device's request packet for the dataset identified above.

```
<FS>
REQ = 1643
JOB = 1234
<RS>
<GS>
```

EXAMPLE 4 - A host's data packet transmitted in response to the above request.

```
<FS>
```

```
ANS = 1643
JOB = 1234
STATUS = 0
DO = B
FRNT = 6.21 ; 6.21
BACK = -5.00 ; -5.00
GBASE = 6.50 ; 6.75
GCROS = 6.75 ; 7.00
GAX = 135 ; 45
GBASEX = 6.48 ; 6.76
GCROSX = 6.76 ; 7.01
SAGRD = 3.81 ; 3.81
SAGBD = 4.50 ; 4.50
SAGCD = 4.10 ; 4.10
TIND = 1.53 ; 1.53
DIA = 75.0 ; 75.0
BCTHK = 15.0 ; 15.0
BETHK = 17.0 ; 17.0
CRIB = 70.0 ; 70.0
GPRVA = 140.0 ; 46.5
GPRVM = 2.4 ; 2.17
KPRVA = ? ; ?
KPRVM = ? ; ?
PIND = 1.498 ; 1.498
LMATID = 1 ; 1
LAPM = -1 ; -1
RNGH = 11.80 ; 11.80
RNGD = 60.0 ; 60.0
GTHK = 2.54 ; 2.52
<RS>
<GS>
```

EXAMPLE 5 – Complete auto-initialization (for an edger)

```
DEVICE                              HOST

<FS>REQ=INI<CR/LF>
<RS>
<GS>
                                    <ACK>
                                    <FS>ANS=INI<CR/LF>
                                    STATUS=0;
                                    <RS>
                                    <GS>
<ACK>

<FS>ANS=INI<CR/LF>
DEV = EDG<CR/LF>
VEN = GC<CR/LF>
MODEL = ELITE<CR/LF>
TRCFMT = 4;400;E;R<CR/LF>
TRCFMT = 1;400;E;R<CR/LF>
```

```
INFO=1;
DEF = FIRSTREQ <CR/LF>
D = HBOX;VBOX;*CIRC;FCRV<CR/LF>
ENDDEF = FIRSTREQ <CR/LF>
<RS>
<GS>
                            <ACK>

                            <FS>ANS=INI<CR/LF>
                            STATUS=0<CR/LF>
                            DEF= FIRSTREQ ;*number<CR/LF>
                            TRCFMT=4;400;E;R<CR/LF>
                            INFO=0<CR/LF>
                            HVEN=ABC<CR/LF>
                            HMODEL=SOFTWARENAME<CR/LF>
                            HVER=1.0<CR/LF>
                            <RS>
                            <GS>
<ACK>
```

**number = number to be used in subsequent requests by device*

7.2.5 Preset Initialization

As of version 3.04, Preset Initialization is deprecated for download devices, having been found to offer no significant benefit. Hosts and devices that implement initialization must implement auto-format initialization. While the implementation of initialization is not mandated for either hosts or devices, it is very strongly recommended. The "no-initialization" mode is provided primarily to provide a more easily implemented mode of operation for use in development, testing, and light-duty applications.

7.2.5.1 A device may effect preset initialization when a host responds to a device's request for initialization with a status code of 14 or, if the host responds with a status code of 0, but the device does not support auto-format initialization.

7.2.5.2 The device's data packet should include the following interface records: device type, vendor, and model ID. The host may be able to use this information to tailor its responses to requests based on the descriptive information provided. It is anticipated that manufacturers of devices may publish the set of records required by their various devices, especially in those cases in which the devices do not support auto-format initialization. In the event that the host does not recognize the vendor and model ID specified, it shall send the entire set of records specified herein for the indicated device type as shown in Annex A. Other interface records, such as operator ID, machine ID, and serial number, should be included if available.

7.2.5.3 NOTE When a device type of "DNL" is specified, and the vendor and model ID are not recognized, the host will send the entire set of device records for the indicated job ID.

7.2.5.4 The device's data packet must not contain a request definition record ("DEF"). It is the absence of this record that provides an indication to the host that preset, rather than auto-format, initialization is being requested.

7.2.5.5 The host's subsequent data packet will contain a request ID that the device will use for subsequent requests. Because there is no request definition, there is no device tag, however there is a field separator as shown in example below.

EXAMPLE - A session in which a host supports only preset initialization.

```
DEVICE                                   HOST

<FS>REQ=INI<CR/LF>
<RS>
<GS>
                                         <ACK>
                                         <FS>ANS=INI<CR/LF>
                                         STATUS=14;PRESET only<CR/LF>
                                         <RS>
                                         <GS>
<ACK>
<FS>ANS=INI<CR/LF>
DEV = TRC<CR/LF>
VEN = GC<CR/LF>
MODEL = FTX<CR/LF>
MNAME = Company XYZ's Tracer<CR/LF>
MID = 24076<CR/LF>
OPERID = JILL<CR/LF>
TRCFMT = 1;400;E;R<CR/LF>
TRCFMT = 4;400;E;R<CR/LF>
(any other Optional Interface RECORDS)<RS>
<GS>
                                         <ACK>
                                         <FS>ANS=INI<CR/LF>
                                         STATUS = 0<CR/LF>
                                         DEF = ;1234<CR/LF>
                                         TRCFMT = 4;400;E;R<CR/LF>
                                         <RS>
                                         <GS>
<ACK>
```

7.2.6 No initialization

While the implementation of initialization is not mandated for either hosts or devices, it is very strongly recommended. The “no-initialization” mode is provided primarily to provide a more easily implemented mode of operation for use in development, testing, and light-duty applications.

7.2.6.1 When neither type of initialization is supported by a host or device, the parameters that would be transmitted from host to device during initialization would have to be included in the data packets transmitted from host to device during each session. Because this is quite inefficient, it is not recommended.

7.2.6.2 The device’s request packets should contain device type, vendor, and model records to allow the host to better tailor the set of records it returns to the device, and such informational records (machine ID, operator ID) as the device supports. When a host receives a request of this type, and does not recognize the vendor and/or model records, it will send all of the records defined in the “preset packet” for the device type. Likewise, when a device type of DNL is specified, all records will be downloaded.

EXAMPLE A device’s request packet, after no initialization has taken place.

```
<FS>
REQ =  GEN
JOB = 1234
VEN = IGC
MODEL = B16
```

```
MNAME = Intergalactic Generator Co. Model 16
MID = QB485M
OPERID = JACK
<RS>
<GS>
```

7.2.7 Minimum host support for initialization

A host shall, as a minimum, be able to respond to an initialization request with a response packet containing a **STATUS=15** (initialization not supported) record.

EXAMPLE A session with a host that does not support initialization.

```
DEVICE                      HOST
<FS>REQ=INI<CR/LF>
<RS>
<GS>
                            <ACK>

                            <FS>ANS=INI<CR/LF>
                            STATUS=15; No can do <CR/LF>
                            <RS>
                            <GS>
<ACK>
```

7.2.8 Special initialization requirements for devices using trace data

Hosts and devices need to negotiate a tracing format that will be acceptable to both. Devices shall identify one or more desired formats by sending one or more TRCFMT records in the initialization data packet or, in the absence of initialization, in the request packet. Each TRCFMT record shall specify one of the tracing types the device would like to use in the order it would prefer to use them (typically from most compact to least compact). The host shall send back in its response one of the TRCFMT records to indicate which one it supports and will actually use. (For more information, see also the description of status code 17 in A.4).

The same method is used to negotiate sag data formats. It is possible for hosts to not support sag data; in such cases, the host shall return "ZFMT=0" to provide the device a clear indication that sag data cannot be accepted or provided.

All hosts and devices conforming to this Standard must support format 1, the ASCII format.

EXAMPLE 1 - An initialization session showing trace format negotiation. Because the device is a tracer, this is a Preset Initialization session.

```
DEVICE                      HOST
<FS>REQ=INI<CR/LF>
<RS>
<GS>
                            <ACK>
                            <FS>ANS=INI<CR/LF>
                            STATUS=14;PRESET only<CR/LF>
                            <RS>
                            <GS>
<ACK>
<FS>ANS = INI<CR/LF>
```

```
DEV = TRC<CR/LF>
VEN = GC<CR/LF>
MODEL = FTX<CR/LF>
TRCFMT = 4;512;E;R<CR/LF>
TRCFMT = 4;400;E;R<CR/LF>
TRCFMT = 1;512;E;R<CR/LF>
TRCFMT = 1;400;E;R<CR/LF>
<RS>
<GS>
                           <ACK>
                           <FS>ANS=INI<CR/LF>
                           STATUS=0<CR/LF>
                           DEF = ;1234<CR/LF>
                           TRCFMT=4;400;E;R<CR/LF>
                           <RS>
                           <GS>
<ACK>
```

EXAMPLE 2 - An initialization session in which the host does not support any of the device’s proposed trace formats.

```
DEVICE                     HOST
<FS>REQ=INI<CR/LF>
<RS>
CRC=12345<CR/LF>
<RS>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                           <ACK>
                           <FS>ANS=INI<CR/LF>
                           STATUS=14;PRESET only<CR/LF>
                           <RS>
                           <GS>
<ACK>
<FS>ANS=INI<CR/LF>
DEV=TRC<CR/LF>
VEN= GC<CR/LF>
MODEL= FTX<CR/LF>
TRCFMT=1;400;E;R<CR/LF>
TRCFMT=4;400;E;R<CR/LF>
TRCFMT=1;512;E;R<CR/LF>
TRCFMT=4;512;E;R<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                           <ACK>
                           <FS>ANS=INI<CR/LF>
                           STATUS=529; 400 ? 512 ? <CR/LF>
                           <RS>
                           <GS>
<ACK>
```

7.3 Upload sessions

7.3.1 Upload sessions consist of the following exchange of messages:

a) A device transmits a request packet to a host.

b) The host responds with a confirmation message.

c) The host transmits a response packet to the device.

d) The device transmits a confirmation message to the host.

e) The device transmits its data packet to the host.

f) The host transmits a confirmation message to the device

g) The host transmits a response packet to the device.

h) The device transmits a confirmation message to the host.

Table 6 – Summary of upload session

DEVICE Transmission	HOST Transmission
Request packet, REQ=<Request>	
	Confirmation <ACK>
	Response packet, ANS=<Request>
Confirmation <ACK>	
Data packet, ANS=<Request>	
	Confirmation <ACK>
	Response packet, ANS=<Request>
Confirmation <ACK>	

7.3.2 If any of the confirmation messages is a negative acknowledgement, the sender shall retry its transmission up to three times. If a negative acknowledgement is received after the third retry (i.e., the fourth attempt), the session terminates.

7.3.3 An upload session can also fail if either of the host's response packets contains a status record whose field value is non-zero. In this case, the session ends after the device's subsequent confirmation message.

7.3.4 The content of each upload session is determined by the contents of the request type (REQ) record. At this time, the preset request types that indicate regular upload sessions are TRC, INS, and the generic UPL.

EXAMPLE : An abstract upload session.

DEVICE HOST

```
<FS>REQ=(*Device type)<CR/LF>
JOB=1234<CR/LF>
[Optional interface RECORDS]
<RS>
```

```
<GS>
                                         <ACK>
                                         <FS>ANS=[*Device type]<CR/LF>
                                         JOB=1234<CR/LF>
                                         STATUS=0<CR/LF>
                                         (any RECORDS that must be echoed)
                                         <RS>
                                         <GS>
<ACK>
<FS>ANS=(*Device type)<CR/LF>
JOB=1234<CR/LF>
(All Mandatory Records for *Device type Packet)
<RS>
<GS>
                                         <ACK>
                                         <FS>ANS=(*Device type)<CR/LF>
                                         JOB=1234<CR/LF>
                                         STATUS=0<CR/LF>
                                         <RS>
                                         <GS>
<ACK>
```

NOTE * Device type = TRC, UPL, INS, or request ID sent by host during initialization.

7.3.5 UPL, the generic upload session request type, allows a device to upload any subset of the defined records to a host. This Standard does not specify what actions hosts shall take upon receipt of such data.

NOTE What a host might do with records uploaded via a UPL request but which are normally downloaded to a device is undefined. Replacing such values as it may currently have would be a reasonable action, but is not explicitly mandated.

7.3.6 INF is sent from devices to hosts to indicate that the process associated with the most recent request for the job indicated has been completed. It is a special, abbreviated variant of upload request, as it contains all of the data to be conveyed in the request packet itself. The INF request type will contain only request type, job, status, and (optionally) CRC and model ID records. The host shall respond to the INF packet with a confirmation message.

EXAMPLE: an INF upload session. First, a device makes a request (in this example, a GEN request).

DEVICE HOST

```
<FS>
REQ=GEN<CR/LF>
JOB=1234<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                                         <ACK>
                                         <FS>
                                         ANS=GEN<CR/LF>
                                         JOB=1234<CR/LF>
                                         (Generator records…)
                                         <RS>
                                         CRC=12345<CR/LF>
                                         (The CRC record is optional; the number shown is
                                   not properly calculated).
```

```
                                                  <GS>
      <ACK>
```

Then, when the generator has finished its processing, it sends the following indicating a normal completion of its process. The host responds only with a confirmation message.

```
<FS>
REQ=INF<CR/LF>
JOB=1234<CR/LF>
STATUS=0<CR/LF>
<RS>
CRC=12345<CR/LF>
(The CRC record is optional; the number shown is not properly calculated).
<GS>
                                                  <ACK>
```

7.3.7 MNT is a request that allows a device to transmit machine status and configuration information to its host. This is a special, abbreviated variant of upload request, as it contains all of the data to be conveyed in the request packet itself. The host shall respond to the MNT packet with a confirmation message.

The contents of the MNT packet are, in this version, entirely up to the manufacturers of devices that implement this feature. As such, it is reasonable to expect that the data packet may consist largely of experimental records having record labels with a leading underscore. Hosts can support this feature minimally by saving the records received to a file.

7.4 Download sessions

7.4.1 Download sessions consist of the following exchange of messages :

a) A device transmits a request packet to a host.

b) The host responds with a confirmation message.

c) The host transmits a data packet to the device.

d) The device responds with a confirmation message.

Table 7 – Summary of download session

DEVICE Transmission	HOST Transmission
Request packet, REQ=<Request>	
	Confirmation <ACK>
	Data packet, ANS=<Request>
Confirmation <ACK>	

7.4.2 If any of the confirmation messages is a negative acknowledgement, the sender shall retry its transmission up to three times. If a negative acknowledgement is received after the third retry (i.e., the fourth attempt), the session terminates.

7.4.3 The content of each download session is determined by the contents of the request type (REQ) record. At this time, the preset request types that indicate download sessions are of PTG, EDG, FBK, SBK, GEN, AGN, COA, FSG, LMD, and the generic DNL.

EXAMPLE 1: an abstract preset packet download. *Device Type* is one of PTG, EDG, FBK, SBK, GEN, AGN, COA, FSG, LMD, DNL, or request ID sent by host during initialization.

DEVICE	HOST
<FS>REQ=(*Device Type*)<CR/LF> **JOB=1234<CR/LF>** **(Optional Interface Records)** **<RS>** **CRC=12345<CR/LF>** **(The CRC record is optional; the number shown is not properly calculated).** **<GS>**	
	<ACK>
	<FS>ANS=(*Device Type*)<CR/LF> **JOB=1234<CR/LF>** **STATUS=0<CR/LF>** **DO=B<CR/LF>** **(All Records for *Device type* Packet)** **<RS>** **<GS>**
<ACK>	

EXAMPLE 2 – A download session showing trace format negotiation with no previous initialization. *Device Type* is one of PTG, EDG, FBK, SBK, GEN, AGN, COA, FSG, LMD or DNL.

DEVICE	HOST
<FS>REQ=(*Device Type*)<CR/LF> **JOB=1234<CR/LF>** **VEN = GC<CR/LF>** **MODEL = FTX<CR/LF>** **TRCFMT = 4;512;E;R<CR/LF>** **TRCFMT = 4;400;E;R<CR/LF>** **TRCFMT = 1;512;E;R<CR/LF>** **TRCFMT = 1;400;E;R<CR/LF>** **<RS>** **CRC=12345<CR/LF>** **(The CRC record is optional; the number shown is not properly calculated).** **<GS>**	
	<ACK>
	<FS>ANS=(*Device Type*)<CR/LF> **JOB=1234<CR/LF>** **STATUS=0<CR/LF>** **DO=B<CR/LF>** **(All Records for *Device type* Packet)**

```
                                         TRCFMT=1;400;E;R;F<CR/LF>
                                         R=2479;2583;2605;2527;2394;2253;2137;
                                         2044;1975;1935<CR/LF>
                                         R=1922;1939;1989;2072;2184;2322;2471;
                                         2599;2645;2579<CR/LF>
                                         (etc…)
                                         <RS>
                                         CRC=12345<CR/LF>
                                         (The CRC record is optional; the number shown is
                                         not properly calculated).
                                         <GS>
<ACK>
```

7.4.4 A device may send a TIM request, which is distinguished from other requests by its lack of a following JOB record. A host's response to a TIM request shall consist of an ANS record (as usual) followed by a TIME record containing the current date and time. A CRC record may be included (as usual).

EXAMPLE 1: a time synchronization download.

```
DEVICE                                   HOST

<FS>REQ=TIM<CR/LF>
<RS>
CRC=12345<CR/LF>
<GS>
                                         <ACK>

                                         <FS>ANS=TIM<CR/LF>
                                         TIME=20070501163330<CR/LF>
                                         STATUS=0<CR/LF>
                                         <RS>
                                         <GS>
<ACK>
```

7.5 File-based information transfer

7.5.1 The Request Types FIL, FRM, and SDF shall be used to identify the contents of a standard data file.

7.5.2 The file will contain printable ASCII characters, line terminators, and depending on the source platform, may contain an end-of-file marker. The line terminators and end-of-file characters may differ depending on the hardware and/or software platform on which the file is produced.

7.5.3 Lines in the file correspond to Records as defined in the Standard.

7.5.4 The contents of the file will be formatted according to the rules for Data Packets specified in the Standard, e.g., the first line of any file shall contain an REQ record, and in the case of files containing job data will have a JOB record on the second line.

7.5.5 The contents of the file will differ from the contents of a Data Packet in the following ways:

a) The Start, Stop, and CRC Position characters will not be included in the file;

b) The CRC Record will not be included in the file;

c) The STATUS record will not be included in the file;

d) The Request Type record shall have the value "FIL", e.g. REQ = FIL, or in the case of a frame data file (as described in the VCA “Rimless Frame Drill Mount Standard”), “FRM”, e.g., REQ=FRM.

7.5.6 The file may contain trace data, in which case, the only format allowed is format 1, ASCII.

7.6 Command sessions

7.6.1 Command sessions provide a means for a device to instruct a host to take an action related to job data (or, if in place of a JOB record, and LIB or RMT record appears, library or remote tracing data). A command session has a Request Type of “CMD”. The command session consists of a series of packets similar to an upload session: an initial request packet sent from a device to a host, followed by a response packet from the host to the device, followed by a data packet from the device to the host, followed by a response packet sent from host to device. The format of the record is:

ACT=ACTION;OPTION

where ACTION is one of a set of literal identifiers defined below; OPTION is one of a set of literals, which may be used depending on the specified ACTION.

The identifiers are listed in sections 7.6.1.1 and following.

EXAMPLE : A command session.

```
DEVICE                                HOST
<FS>REQ=CMD<CR/LF>
JOB=1234<CR/LF>
<RS>
<GS>
                                      <ACK>
                                      <FS>ANS=CMD<CR/LF>
                                      JOB=1234<CR/LF>
                                      STATUS=0<CR/LF>
                                      <RS>
                                      <GS>
<ACK>
<FS>ANS=CMD<CR/LF>
JOB=1234<CR/LF>
ACT=NEW<CR/LF>
DO=B
(Data records to be saved for job 1234)
<RS>
<GS>
                                      <ACK>
                                      <FS>ANS=CMD<CR/LF>
                                      JOB=1234<CR/LF>
                                      STATUS=0<CR/LF>
                                      <RS>
                                      <GS>
<ACK>
```

7.6.1.1 **NEW**. Any existing data for the specified job shall be deleted, and data in the packet shall be saved under the specified job identifier. The data must be complete and apply for both sides of a two-eye job. The DO record must be included to indicate the sides comprised in the order.

7.6.1.2 **MERGE**. Any existing data for the specified job shall be preserved, but records in the packet shall replace any existing matching records.

7.6.1.3 **DELETE**. Existing data for the specified job shall be deleted. The OPTION "SDF" may appear to indicate that any SDF files for the specified job are to be deleted, in which case, any SDF files related to the job shall be deleted and other job data shall remain undisturbed.

7.6.2 Hosts that do not support command sessions should respond to the CMD request by including STATUS=8 in its response packet. Upon receipt of a response packet including a non-zero field value for STATUS, the session shall end with the device's acknowledgement message.

EXAMPLE : An command session in which the host does not support command sessions.

```
DEVICE                                          HOST
<FS>REQ=CMD<CR/LF>
JOB=1234<CR/LF>
<RS>
<GS>
                                                <ACK>
                                                <FS>ANS=CMD<CR/LF>
                                                JOB=1234<CR/LF>
                                                STATUS=8;unrecognized request type<CR/LF>
                                                <RS>
                                                <GS>
<ACK>
```

7.6.3 Initialization. A device may include a special form of the ACT record in its initialization data packet in order to indicate to the host the actions the performance of which it can request. Upon receipt of such a record, the host shall reply with a similar record, listing the actions it can perform. A host that does not support command sessions should return no ACT record in its response packet (because it would not recognize the record, and would therefore ignore it), indicating to the device that it does not support them.

EXAMPLE : An initialization session including an ACT record:

```
DEVICE                                  HOST
<FS>REQ=INI<CR/LF>
<RS>
<GS>
                                        <ACK>
                                        <FS>ANS=INI<CR/LF>
                                        STATUS=0;
                                        <RS>
                                        <GS>
<ACK>

<FS>ANS=INI<CR/LF>
```

```
DEV=EDG<CR/LF>
VEN=GC<CR/LF>
MODEL=ELITE<CR/LF>
TRCFMT=4;400;E;R<CR/LF>
TRCFMT=1;400;E;R<CR/LF>
INFO=1
DEF=FIRSTREQ <CR/LF>
D=HBOX;VBOX;CIRC;FCRV<CR/LF>
ENDDEF=FIRSTREQ <CR/LF>
ACT=NEW;MERGE;DELETE
<RS>
<GS>
                           <ACK>

                           <FS>ANS=INI<CR/LF>
                           STATUS=0<CR/LF>
                           DEF=FIRSTREQ;1234<CR/LF>
                           TRCFMT=4;400;E;R<CR/LF>
                           INFO=0<CR/LF>
                           HVEN=ABC<CR/LF>
                           HMODEL=SOFTWARENAME<CR/LF>
                           HVER=1.0<CR/LF>
                           ACT=NEW;MERGE;DELETE
                           <RS>
                           <GS>
<ACK>
```

7.7 Lens Data Session

7.7.1 A Lens Data Session is a download session that conveys lens blank data to a device. It commences with **REQ=LNF**, which is followed immediately by an OPC record specifying one or two OPC codes.

7.7.2 Lens data requests may be intialized, however, hosts that are not equipped to handle LNF requests may have a problem with the missing JOB (or LIB or RMT) record on the second line in an initialized request. (In a preset LNF request, such a host would probably return a STATUS=8, Invalid Request Type). Devices must therefore afford the possibility of not making LNF requests.

7.8 Communications

7.8.1 Serial connections

7.8.1.1 Serial connections shall be effected pursuant to EIA-232C (equivalent to ITU V.24).

7.8.1.2 **EIA-232 Communications parameters.**

Default parameters are 9600 baud, 8 data bits, 1 stop bit, and no parity. Hosts and devices should default to these parameters. Installation personnel could implement other settings on-site. Faster baud rates are highly recommended to avoid timeouts. Flow control, if implemented, shall be accomplished using EIA-232 control lines. Software flow control (XON/XOFF) is *not* implemented under this Standard. The XON and XOFF characters are, however, reserved.

7.8.2 Network connections

7.8.2.1 Network connections shall consist of TCP/IP socket connections. A “socket” consists of an IP address and a TCP port number.

7.8.2.2 In the context of TCP socket connections, the designation “server” indicates that on such a computer, a software process listens for connection requests; “clients” make such connection requests.

7.8.2.3 Sessions are identical over network or serial connections.

7.8.2.4 Standard timeouts apply to network connections.

7.8.2.5 Standard Ports

Standard port numbers are designated, which ports shall be registered with the IANA (see http://www.iana.org/cgi-bin/usr-port-number.pl) as described below. Standard ports shall be designated for the “Rendezvous” and “Remote” ports as described below:

7.8.2.5.1 A “Rendezvous Port” is specified, port number 33511, to be used to allow Devices to broadcast information describing a socket to which a Host will effect a connection through the Autoconnect process. A Host server process listens on this port for UDP broadcast datagrams as defined herein. The Rendezvous Port is invariant across networks, that is, on any network on which a Host that supports Autoconnect resides, a Device can be assured that the Host has a listener process on this port. The Rendezvous Port is used only in conjunction with the Autoconnect method described in 7.2.7.

7.8.2.5.2 A “Device Port” is specified, on which a Device’s server process listens for connections from Hosts. The connection established on this socket is the connection over which standard DCS sessions are transacted. The default Device Port number shall be 50000.

NOTE It is unnecessary to register a port number for the Device Port because devices are unlikely to encounter port contention issues – unlike the computers to which they are connected, it is unlikely that devices will run any software unknown to the manufacturer of the Device, which might use ports arbitrarily. It is sufficient to specify a port in the IANA “private” range. The Device Port number is specified in order to provide a reasonable default.

7.8.2.5.3 A “Remote Port” is **specified,** port number 33512, on which a Host server process listens for connection requests from Devices. This is intended for use by Devices connecting over the Internet, or by Devices which do not use the Autoconnect method described below. (The Autoconnect method is not intended to function over the Internet.) The connection established on this socket is the connection over which standard DCS sessions are transacted.

7.8.2.5.4 Hosts and Devices shall use the standard port numbers specified herein for their corresponding purposes, but shall also provide a means whereby alternative port numbers can be used to accommodate such special circumstances as may arise.

7.8.2.6 IP Address Assignment.

There are three Standard means of assigning IP addresses to devices.

7.8.2.6.1 Manual. Compliant Devices shall provide a means whereby a device's IP Address and Device Port can be set manually, to be used in the event that neither of the following two automatic address-assignment methods are available.

7.8.2.6.2 DHCP. Compliant devices should support automatic IP address assignment via Dynamic Host Configuration Protocol.

7.8.2.6.3 APIPA. Compliant devices may support Automatic Private IP Addressing.

NOTE: APIPA is used by computers running the Windows operating system when DHCP is not available on a network as is frequently the case on very small networks. In this scheme, computers assign themselves addresses in the APIPA range 169.254.0.0 – 169.254.255.255, using a subnet mask of 255.255.0.0.

7.8.2.7 Connections

Once a socket connection has been established, normal DCS communications ensue. There are no differences between Sessions conducted over a network connection and those conducted over a serial connection. However, it should be noted that CRC records are superfluous on a network, as TCP provides its own error-checking. Three different methods of establishing network connections are defined in sections 7.2.7.1, 7.2.7.2, and 7.2.7.3.

7.8.2.7.1 Autoconnect Connection

The purpose of the Autoconnect process is to enable connections between devices and hosts to be completely automatic, requiring no manual configuration on either side. Upon power-up, or at some other time prior to sending any Request Packets, a device broadcasts information about the socket on which it runs a server process, awaiting a connection. Upon receiving such a broadcast, a host will effect a connection to the socket specified by the device (which should nominally be the "Device Port" at the Device's IP address, but which is in all cases specified in the Autoconnect Packet described below).

Complete automation also requires the implementation of one of the automatic IP address assignment techniques specified in 7.2.6.

The Autoconnect process consists of the following steps:

A Host on the network runs a server process on the Rendezvous Port.

On power-up, after obtaining its IP address by one of the standard means, a Device broadcasts a UDP datagram containing the following information (see packet description below) and using the UDP broadcast address and the Rendezvous Port:

The Device's IP Address; and

The Device Port, which is the port number on the Device on which it has a server process awaiting connection.

Upon receipt of the Device's UDP datagram, the Host effects a connection to the Device Port specified therein.

Upon connection to the socket specified by the Device's IP Address and Device Port, Standard DCS Sessions (such as initialization) can commence.

The Autoconnect Packet has the same general form as all DCS packets.

A new Request Type, "**CON**", is provided to uniquely identify the packet, e.g., **REQ=CON**. In addition, the packet must include the following records:

IP Address, using Record Label "IP"; example **IP=169.254.1.1**

Device Port, using Record Label "PORT"; example **PORT=50000**

In addition, the Autoconnect packet shall contain **DEV, VEN,** and **MODEL** records, and may contain **MNAME** and **SN** records.

EXAMPLE: An "Autoconnect" packet:

```
<FS>
REQ=CON
IP=169.254.1.1
PORT=50000
DEV=GEN
VEN=GC
MODEL=FERRET
MNAME= Ferret Lens Generator
SN=12345
<RS>
<GS>
```

There is no response to the Autoconnect packet other than the establishment of a connection by the host.

In the event of disconnection, the device re-initiates the Autoconnect process.

7.8.2.7.2 Local Connection

In the "Local Connection" mode, devices are servers, and hosts are clients. A device operating in this mode must therefore await a connection to its "Device Port" from a host. Although there is no standard means for a host to know the device sockets to which it is to effect such connections, the default port defined in section 7.2.5.2 (50000) should be assumed.

7.8.2.7.3 Remote Connection

In the "Remote Connection" mode, hosts are servers, and devices are clients. A device operating in this mode must therefore initiate a connection to a host's "Remote Port". There is no standard means for a device to know the host socket to which it is to effect such a connection (while the standard port 33512 is used, no means is provided for determining the server's IP address, which will therefore need to be set manually).

7.8.2.7.4 Connection Delay

Devices shall wait three seconds before initiating communications sessions on a newly established TCP/IP connection. This provision applies to all of the Connections defined in this section 7.2.7.

8 Other requirements

8.1 Operator messages

Devices should provide operators with messages that report on the progress of sessions; e.g., a device could display “Sending Request…” when transmitting a request packet and change this to “Awaiting Response…” after receipt of the confirmation message.

8.2 Host Requirement

Hosts shall allow multiple initializations or request types on each communications port in use.

Annex A
(normative)

Record labels

A.1 Device records

A.1.1 Table A.1 contains all of the device records defined for use by systems conforming to this Standard, arranged alphabetically. Under the column labeled "Data type", the proper characteristics of the data associated with each label is indicated.

A.1.1.1 The field separator character identifies the chirality of the record.

A.1.1.2 The lack of a field separator (semi-colon) indicates that only one value is expected, i.e., the record is not chiral.

EXAMPLE: A non-chiral record descriptor

```
LABEL     type      description
```

A.1.1.3 A trailing semi-colon indicates that chiral data is expected. Data for either eye may be empty, but the field separator shall be present.

EXAMPLE: A chiral record descriptor

```
LABEL     type;     description
```

A.1.1.4 Square brackets around the semi-colon indicate the start of optional left eye data which, if present, follows the specification for chiral data. A single value, whether or not followed by a semi-colon, applies to both eyes. A leading semi-colon followed by a value applies to the left eye only. For example, if only one circumference is supplied, it is assumed to apply to both right and left eyes.

EXAMPLE: An optional chiral record descriptor

```
LABEL     type[;]   description
```

A.1.1.5 The data type indicates constraints on the kind of data that may appear in the record's fields.

A.1.1.5.1 Numeric fields may or may not contain a decimal point. If a decimal point is present, hosts and devices should be able to correctly parse any degree of precision, including zero. Numeric format should be flexible, but reasonable.

A.1.1.5.2 Integer fields shall not contain decimal points.

A.1.1.5.3 The ± symbol indicates that a number may be positive or negative. The absence of the symbol indicates that a value is expected to be positive.

A.1.1.5.4 Square brackets [] around any element other than the semi-colon indicate that the enclosed item is optional, however, it is strongly recommended that whenever a single field value is present in a chiral record, whether it is followed by a field separator or not, the field value should be applied to both sides, or to the left side of the left-side-only order.

A.1.1.6 The following units of measure are used in Table A.1:

— millimeters.

— diopters: the index of refraction to be used will be specified

— degrees.

Table A.1 – Device records

Record Label	Data type	Description
A	integer; integer;…	Angle data for radius data (angular locations of trace radii expressed in R record(s)).
ABBE	numeric;	Abbe value of lens material.
ACCN	text	Account number
ACOAT	text[;]	Applied coating
ADD	numeric;	Addition power if multifocal, progressive or executive type lens. (diopters)
ADD2	numeric;	2^{nd} (upper) Addition power (diopters)
ADJADD	± numeric;	Delta (correction to) Add Power
ADJAX	± numeric;	Delta (correction to) Cylinder Axis
ADJCYL	± numeric;	Delta (correction to) Cylinder Power
ADJSPH	± numeric;	Delta (correction to) Sphere Power
ADJPRVA	± numeric;	Delta (correction to) Prism Axis
ADJPRVM	± numeric;	Delta (correction to) Prism (LIND or "natural" diopters)
AR	numeric	Flag indicating whether lens is to be anti-reflection coated. 0 – no A/R coating specified 1 – A/R coating specified
AVAL	numeric;	Value indicating the height of a semi-finished blank prior to generating measured relative to the machine's reception chuck on the centerline of the chuck
AX	numeric;	Prescribed cylinder axis. May be 0 - 180. (degrees) Notice that this may be different from GAX which is generator AXIS.
BACK	± numeric;	Blank back curve. (TIND diopters)
BCERIN	± numeric;	Horizontal distance from blank center to engraving reference point (midpoint on 180 line between semi-visible marks)
BCERUP	± numeric;	Vertical distance from blank center to engraving reference point (midpoint on 180 line between semi-visible marks)

BCOCIN BCOCUP	± numeric; ± numeric;	Blank geometrical center to Prism Reference Point (PRP) In & Down (mm). Useful for uncuts where frame information is not available. +IN means PRP towards nasal with respect to the geometrical blank center. -IN means PRP towards temporal with respect to the geometrical blank center. +UP means PRP is above the geometrical blank center. -UP means PRP is below the geometrical blank center.
BCSGIN BCSGUP	± numeric; ± numeric;	Blank geometrical center to Layout Reference Point (LRP) In & Down (mm), i.e. manufacturer's stated segment position relative to geometric center of the blank. +IN means LRP towards nasal from the blank center. -IN means LRP towards temporal from the blank center. +UP means LRP is above the blank center. -UP means LRP is below the blank center.
BCTHK	numeric;	Blank center thickness (mm).
BETHK	numeric;	Blank edge thickness (mm).
BEVC	numeric;	Bevel curve. A diopter value (1.530 index) for the bevel curve to follow.
BEVH	numeric;	Bevel height. Used when HICRV=1; indicates height of bevel in mm.
BEVM	numeric;	Bevel modifier. Percentage or distance in mm based on value of BEVP. Percentage should be expressed as a number between 0 and 100.
BEVP	integer[;]	Bevel position 0 – Manual 1 – Follow front (BEVM = mm from front) 2 – Percent back (BEVM = % from front) 3 – Frame curve (BEVC = curve to follow, BEVM = mm from front) 4 - 50/50 5 - Follow back (BEVM = mm from back) 6, 8, 9 - Unused 7 – Automatic (Use edger settings) (If HICRV=1, use BEVH, BTILT, TILTBASE records) 10 - Free float
BLKB	± numeric;	Block base curve (TIND diopters).
BILLOK	integer	Billing acceptance indicator, sent by LMS to LDS: 0 – do not process order if billable 1 – process order, billable or not.
BLKCMP	± numeric;	Generator thickness compensation for block / lens curve mismatch.
BLKD	numeric;	Block diameter (mm).

BLKM	integer;	Surface Blocking mode. 0 – Blank Geometric Center; 1 – Prism Reference Point; 2 – Frame Center; 3 – Other.
BLKROT	numeric;	Block rotation. A configuration value provided by an LDS to be sent by the LMS to the LDS.
BLKTYP	integer;	Integer block type. A number agreed to by device and host to indicate which of several block tables to use. Can be used to distinguish between different types or materials of blocks.
BPRVA	numeric;	Base setting at which BPRVM is located, 0-360 degrees.
BPRVM	numeric;	Amount of prism in semi-finished blank. (LIND or “natural” diopters.)
BRGSIZ	integer	Bridge size of frame.
BSIZ	± numeric[;]	Boxed lens size adjustment: sizing to be applied to shape by horizontal lens size. A device that accepts this record is expected to be able to size any shape sent to it by the amount indicated.
BTILT	± numeric[;]	Bevel tilt. Desired tilt of the bevel, expressed in degrees; a positive value tilts the bevel towards the posterior side of the lens.
BVAL	numeric[;]	Block Center Height distance (mm). Height of the block cavity at the center of the block above the reference surface of the block.
BVD	numeric;	Back vertex distance, as worn (mm).
CAPE	numeric	Headscape angle. Yaw (lateral) angle (in degrees) of the wearer’s natural head position, used to determine a true interpupillary distance.
CARDID	text[;]	Identifier for warranty card to use or print.
CARDTYP	text[;]	Identifier for warranty card layout.
CBUMP	numeric[;]	Required minimum excess material for edging used in computing crib diameter in mm; may be conveyed to LDS.
CCFILMTYP	integer;	Integer indicator of film type to be applied to concave surface, agreed upon by host and device.
CCPOSDRP	± numericl ± numericl numericl numericl numeric[;]	Position of the intersection with the back surface of a measurement ray passing through the Distance Reference Point (defined on the front surface) and normal to the back surface, and the orientation of this ray, expressed in spherical coordinate form. There are five fields for each side: the x, y, and z coordinates of the location of the intersection of the ray with the back surface, the magnitude of the tilt (in degrees), and the rotational orientation of the tilt (in degrees).

CCPOSNRP	numericl numericl numericl numericl numeric[;]	Position of the intersection with the back surface of a measurement ray passing through the Near Reference Point (defined on the front surface) and normal to the back surface, and the orientation of this ray, expressed in spherical coordinate form. There are five fields for each side: the x, y, and z coordinates of the location of the intersection of the ray with the back surface, the magnitude of the tilt (in degrees), and the rotational orientation of the tilt (in degrees).
CCPOSPRP	numeric, numeric, numeric, numeric, numeric[;]	Position of the intersection with the back surface of a measurement ray passing through the Prism Reference Point (defined on the front surface) and normal to the back surface, and the orientation of this ray, expressed in spherical coordinate form. There are five fields for each side: the x, y, and z coordinates of the location of the intersection of the ray with the back surface, the magnitude of the tilt (in degrees), and the rotational orientation of the tilt (in degrees).
CIRC	numeric[;]	Circumference of tracing (two-dimensional).
CIRC3D	numeric[;]	Circumference of tracing (three-dimensional).
CLAGE	numeric	Wearer's age (years).
CLAMP	integer;	Edger clamping pressure ranging from 1 to 10 where 1 is low pressure and 10 is high pressure. Machines with smaller ranges should convert the value given to a comparable value for their use.
CLDEYE	RILIN	Wearer's dominant eye, sent to LDS.
CLFRNT	numeric;	Wearer's prior lens front curve (TIND diopters).
CLHLAT	RIL]N	Wearer's dominant hand, sent to LDS.
CLHT	numeric	Wearer's height (meters).
CLICKS	RILIBIN	In an LMS packet, sent by a Lens Design System, indicates the lens(es) for which charges have been incurred in a given session.
CLIENT	text	Wearer's full or last name (surname).
CLIENTF	text	Wearer's first (given) name,
CLINIT	text	Wearer's initials.
CLLIFE	text	Wearer's lifestyle or hobby.
CLLNAM	text	Wearer's prior lens brand or type.
CLZTILT	numeric;	Panoramic (faceform) tilt of lens as measured on patient (degrees).
COLR	text[;]	Lens color abbreviation
CORRLEN	integer[;]	Nominal Corridor Length (mm).
CPID	integer	Coating process ID 0 – None Others agreed on by device and host
CPRVA	numeric;	Meridian at which CPRVM is located, 0-360 degrees.
CPRVM	numeric;	Magnitude of compensated prism to grind. Prism is compensated for the effect of tilting the lens off the axis of the cutter. See also record GPRVM. (degrees or diopters, see record PIND)

CRIB	± numeric;	Crib diameter (mm). A value of zero (0) will mean no cribbing. A positive value will mean the crib diameter, a negative value will mean the take off amount (from blank diameter) intended for touching up uncut lenses.
CRIBAX	numeric;	Rotational orientation (long dimension) of elliptical or truncated crib.
CRIBELLOK	Integer	0 – LDS may not return elliptical crib parameters (CRIB, ELLH). 1 – LDS may return elliptical crib parameters (CRIB, ELLH).
CRIBFLATOK	Integer	0 – LDS may not return truncation crib parameters (FLATA, FLATB). 1 – LDS may return truncation crib parameters (FLATA, FLATB).
CRIBFMTOK	Integer	0 – LDS may not return CRIBFMT dataset. 1 – LDS may return CRIBFMT dataset.
CRIBRNDOK	Integer	0 – LDS may not return CRIB parameter. 1 – LDS may return CRIB parameter.
CSIZ	± numeric[;]	Circumference sizing: sizing to be applied to shape by circumference. A device that accepts this record is expected to be able to size any shape sent to it by the amount indicated.
CTHICK	numeric;	Finished center thickness (mm at distance O.C.).
CTHKA	numeric;	Angle (in degrees) at which CTHKP occurs.
CTHKP	numeric;	Thickness (in mm) at thickest point along crib perimeter.
CTHNA	numeric;	Angle (in degrees) at which CTHNP occurs.
CTHNP	numeric;	Thickness (in mm) at thinnest point along crib perimeter.
CXFILMTYP	integer;	Integer indicator of film type to be applied to convex surface, agreed upon by host and device.
CYL	± numeric;	Rx cylinder power (diopters).
DBL	numeric	Distance between lenses (mm).
DDFPATH	text	Design Deviation File location, see section 5.14.
DESTID	text	Shipping bin location or shipping method identifier.
DIA	numeric;	Blank diameter (mm).

DRILL *Deprecated, see DRILLE*	literal;±numeric; ±numeric; [±numeric]; [±numeric]; [±numeric]; [±numeric]; [literal]; [±numeric]; [±numeric]	This record has the form DRILL = RILIB; x-start; y-start;[diameter];[x-end];[y-end];[depth];[BIFIA]; [lateral angle]; [vertical angle] The first field is the eye drill data is to be applied to: R = right eye L = left eye B = both eyes. The xy coordinates oriented as right eye data. The second field is the Cartesian x coordinate (mm) of hole start position. The third field is the Cartesian y coordinate (mm) of hole start position. The fourth field is the hole diameter (mm). If this field is empty, the hole will be drilled to diameter of tool. The fifth field is the Cartesian x coordinate (mm) of hole end position. If this field is empty, a normal (round) hole will be d. The sixth field is the Cartesian y coordinate (mm) of hole end position. If this field is empty, a normal (round) hole will be drilled. The seventh field is the depth (mm) for drilling a blind hole. If this field is empty, hole will be drilled completely through lens. The eighth field is the lens surface hole should be drilled normal to: B = normal to back lens surface F = normal to front lens surface A = specified drill angle from next field The ninth field is the lateral hole-drilling angle. (degrees) The tenth field is the vertical hole-drilling angle. (degrees) The Cartesian coordinates for the hole start and end positions are referenced viewing the front surface of the lens, with its origin at the Frame Box center and: +x = Right Eye towards nasal, Left eye towards temple -x = Right Eye towards temple, Left eye towards nasal +y = above frame center -y = below frame center See Figure A.1.

DRILL (cont.)		Fields 8, 9 and 10 are used as follows: The hole is drilled either parallel to a 'drill-reference axis' or at a specified angle relative to this axis. The drill-reference axis is the normal to the front or back surface at box center. The letter F or B in field 8 indicates use of the front-surface drill-reference axis or back-surface drill-reference axis. The letter A in field 8 indicates that the hole is drilled at an angle to the front-surface drill-reference axis. Field 8 may contain only a single letter: F, B or A. The lateral and vertical components of the angle appear in fields 9 and 10, respectively. If the letter A appears in field 8 and both fields 9 and 10 are empty, the hole will be drilled parallel to the front-surface drill-reference axis. If field 8 does not contain a valid letter (F, B or A) the hole will be drilled parallel to the front-surface drill-reference axis. The lateral and vertical angles of the drilled hole are expressed in degrees with zero being parallel to the drill-reference axis. The lateral angle for the right lens has a positive value if the centerline of the hole, when projected forward from the lens front surface, moves toward the nasal side. In the case of the left lens, a positive lateral angle results in the centerline moving toward the temple side. The vertical angle for both right and left lens has a positive value if the centerline of the hole, when projected forward from the lens front surface, moves toward the top of the lens. See Figures A.2, and A.3. Each hole will have a separate DRILL record, so there will typically be multiple occurrences of the DRILL record within an OMA message. If FBFCIN and FBFCUP fields are present, indicating shape is to be decentered, DRILL x and y coordinates must be altered appropriately.

DRILLS	literal; integer; numeric; [numeric]; [numeric]; [numeric]; [numeric]; numeric; [±numeric]; [±numeric]	This record has the form DRILLS = RILIB; type; start angle; start distance; end angle; end distance; diameter; depth; [lateral angle]; [rotational angle] a) The first field is the eye drill data is to be applied to. This field is required. R = right eye L = left eye B = both eyes b) The second field is the feature type. This field is required. 1 = A hole or a slot 2 = A rectangle c) The third field is the start angle. (degrees) This is the start location of the hole, slot, or rectangle as viewed from the lens front. This field is required. d) The fourth field is the start distance. (mm) This is the start location of the hole, slot, or rectangle in relation to the lens front. If this field is absent, then the start location will be centered on the lens edge. e) The fifth field is the end angle. (degrees) This is the end location of the hole, slot, or rectangle as viewed from the lens front. This field is required if the drill feature is not a hole. f) The sixth field is the end distance. (mm) This is the end location of the hole, slot, or rectangle as viewed from the lens front. If this field is absent, then the end distance will be the same as the start distance. g) The seventh field is the diameter of the hole, or the width if the feature is a slot or rectangle. (mm) h) The eighth field is the depth of the drill feature. (mm) This field is required. i) The ninth field is the lateral angle of the drill feature. (degrees) If this field is absent, the lateral drill angle will be normal to the lens edge. j) The tenth field is the rotational angle of the drill feature in relation to the lens edge. (degrees) If this field is absent, the rotational drill angle will be normal to the lens edge. A negative angle indicates the rotation is in a clockwise direction as viewed from the lens front, and a positive angle indicates the rotation is in a counterclockwise direction as viewed from the lens front.

DRILLE	literal; literal; ±numeric; ±numeric; [numeric]; [±numeric]; [±numeric]; [±numeric]; [integer]; [literal]; [±numeric]; [±numeric]	See section 5.5.2 for detailed descriptions of the fields in this record.
DZX	numeric;numeric…	Slopes at sides of surface matrix, see 0
DZXY	numeric;numeric…	Slopes at corners of surface matrix, see 0
DZY	numeric;numeric…	Slopes at top and bottom of surface matrix, see 0
EDATA	numeric;numeric;text	Encrypted data. The first field specifies the encryption method: 1 – RSA 2 – PKCS#7 (S/MIME) The second field specifies the number of characters in the third field (excluding the record terminator). The third field contains the encrypted data, see 5.9.7.5.
EDGED	0I1;	Indicates current state of lens (0 = not edged; 1 = edged)
EECMP	± numeric;	Elliptical error curve compensation (diopters based on TIND). This compensation is applied to GCROS.
ELLH	numeric;	Cribbing ellipse height (mm). Used with the record CRIB to form an ellipse.

ENGMARK		See section 5.15.1 for a description of this record.
ENGMASK	text[;]	Identifier of engraving mask to apply
ENGLOC	literal	Surface on which engraving is to be done, indicated by one of F (front), B (back), or N (no engraving).
ENVID	text	Shipping packaging identifier.
EPRESS	integer	Pressure applied to lens against edger wheels 0 – Undefined 1 – Very Fragile 2 - Soft Pressure 3 – Medium Pressure 4 - Hard Pressure
ERCD	numeric[;]	Eye center-of-rotation distance. Distance (mm) from the back surface of the lens to the point at which all gaze directions pass within the eye.
ERDRIN ERDRUP	± numeric;	Engraving reference point (ERP) to distance reference point (DRP) vectors. +IN means DRP is towards nasal relative to the ERP. -IN means DRP is towards temporal relative to the ERP. +UP means DRP is above the ERP. -UP means DRP is below the ERP.
ERNRIN ERNRUP	± numeric;	Engraving reference point (ERP) to near reference point (NRP) vectors. +IN means NRP is towards nasal relative to the ERP. -IN means NRP is towards temporal relative to the ERP. +UP means NRP is above the ERP. -UP means NRP is below the ERP.
EROCIN EROCUP	± numeric;	Engraving reference point (ERP) to prism reference point (PRP) vectors. +IN means PRP is towards nasal relative to the ERP. -IN means PRP is towards temporal relative to the ERP. +UP means PRP is above the ERP. -UP means PRP is below the ERP.

ERSGIN ERSGUP	± numeric;	Engraving reference point (ERP) to layout reference point (LRP) vectors +IN means LRP is towards nasal relative to the ERP. -IN means LRP reference point is towards temporal relative to the ERP. +UP means LRP reference point is above the ERP. -UP means LRP reference point is below the ERP.
ETYP	± integer	Type of edge to cut on lens: -1 - Uncut 1 – Bevel ("V" shown in Figure A.7) 2 – Rimless ("Flat" shown in Figure A.7) 3 – Groove 4 – Mini-bevel (a smaller-than-traditional "V") 5 – "T" bevel (see Figure A.7) 6 – "U" bevel (see Figure A.7) 7…32 reserved 33…127 defined between host and device
EYESIZ	integer	Nominal lens size of frame.
FBERIN FBERUP	± numeric;	Finish block to Engraving Reference Point vectors. +IN means ERP is towards nasal relative to the finish block. -IN means ERP is towards temporal relative to the finish block. +UP means ERP is above the finish block. -UP means ERP is below the finish block.
FBFCIN FBFCUP	± numeric;	Finish block to frame center vectors. Causes decentering of shape on edger by moving frame center with respect to the finish block. +IN means frame center is towards nasal relative to the finish block. -IN means frame center is towards temporal relative to the finish block. +UP means frame center is above the finish block. -UP means frame center is below the finish block.
FBLKH	numeric;	Finish block horizontal dimension (mm).
FBLKTYP	numeric;	Integer finish block identifier. A number agreed to by the device and host. The value zero is reserved to mean undefined.
FBLKV	numeric;	Finish block vertical dimension (mm).

FBOCIN FBOCUP	± numeric; ± numeric;	Finish block to prism reference point vectors (mm). Can be used for single vision and for multifocals. +IN means PRP is towards nasal relative to the finish block. -IN means PRP is towards temporal relative to the finish block. +UP means PRP is above the finish block. -UP means PRP is below the finish block.
FBSGIN FBSGUP	± numeric; ± numeric;	Finish block to layout reference point vectors (mm). If sent for single vision, assumed the same as FBOCUP/FBOCIN defined below. +IN means LRP is towards nasal relative to the finish block. -IN means LRP is towards temporal relative to the finish block. +UP means LRP is above the finish block. -UP means LRP is below the finish block
FCHT	± numeric;	Height of eyewire centerline in frame measured from bottom of frame (mm). Half of VBOX plus elevation due to additional thickness of frame at bottom as it rests on a flat surface.
FCOAT	text[;]	Factory coating
FCOCIN FCOCUP	± numeric; ± numeric;	Frame center to prism reference point vectors, i.e., O.C. Inset and Drop (mm). +IN means PRP is towards nasal with respect to the frame center. -IN means PRP is towards temporal with respect to the frame center. +UP means PRP is above the frame center. -UP means PRP s below the frame center.
FCOL	text	Color name of frame.
FCRV	± numeric [;]	Frame curve (TIND diopters).
FCSGIN FCSGUP	± numeric;	Frame center to layout reference point vectors, i.e. seg inset & drop (mm). +IN means LRP is towards nasal with respect to the frame center. -IN means LRP is towards temporal with respect to the frame center. +UP means LRP is above the frame center. -UP means LRP is below the frame center.
FDSRC	integer	Frame data source; 0 – uncut blank dimensions 1 – tracing 2 – frame measurements (HBOX, VBOX, FED, FEDAX, DBL)
FED	numeric;	Frame Effective Diameter (twice the longest radius from box center to edge of shape) (mm).
FEDAX	numeric;	Meridian at which FED occurs (degrees).

FINCMP	± numeric;	Fine off allowance compensation (mm). This compensation is applied to GTHK.
FLATA	numeric;	Flattening dimension vertical (for square crib) (mm). Use with record CRIB.
FLATB	numeric;	Flattening dimension horizontal (for square crib) (mm). Use with record CRIB.
FMAT	text	Material of frame. (e.g., METL)
FMFR	text	Manufacturer of frame.
FOD	numeric[;]	Refracted object distance at far viewing point (meters). Use 15m to indicate infinity.
FPINB	numeric;	Edger front pin bevel width in mm measured along cut part of lens.
FRAM	text	Name of frame.
FRNT	± numeric;	Blank true front curve for power calculations. (TIND diopters)
FTTHK	numeric;	First touch thickness, i.e. thickness at which generator wheel first touches the lens (at any location).
FTYP	integer	Integer frame type. 0– Undefined 1 – Plastic 2 – Metal 3 – Rimless 4 – Optyl 5..127 – reserved
FUPC	text	Frame SKU code.
FWA	numeric;	Far working angle - elevation of far viewing point (degrees; above/below horizon, positive indicating above).
FWD	numeric;	Working object distance at far viewing point (meters). Use 15m or greater for infinity.
GAX	numeric;	Cylinder axis for surfacing machines, 0 - 180 degrees.
GBASE	± numeric;	Generator base curve. (rounded, TIND diopters)
GBASEX	± numeric;	Generator base curve. (unrounded, TIND diopters)
GCROS	± numeric;	Generator cross curve. (rounded, TIND diopters)
GCROSX	± numeric;	Generator cross curve. (unrounded, TIND diopters)
GDEPTH	numeric;	Groove depth in mm when ETYP = 3.
GPRVA	numeric;	Base setting at which to generate GPRVM, 0-360 degrees. NOTE: Prism base setting means rotational orientation of the line from apex to base in a principal section of a prism (see 10.7 of ISO 13666:1998).

GPRVM	numeric;	Amount of prism to generate. See also records KPRVM and CPRVM. (degrees or diopters, see record PIND)
GRADIENT	integer	Lens is gradient tinted: 0 – solid or no tint 1 – gradient tint
GTHK	numeric;	Generator thickness at center of surface block (mm).
GWIDTH	numeric;	Groove width in mm when ETYP = 3.
HASENG	Integer[;]	Lens has semi-visible markings 0 – no semi-visible markings 1 – semi-visible markings present
HBOX	numeric [;]	Horizontal boxed lens size of frame (mm).
HEADK	numeric	Head-eye coefficient.
HEADS	numeric	Head-eye stability.
HICRV	integer;	High-curve edging mode 0 – regular mode 1 – high-curve mode
IFRNT	± numeric;	Blank implied front curve; equivalent to the SAGRD value converted to a dioptric curve based on TIND.
INKMASK	text[;]	Identifier of ink marking mask to apply
INSADD	± numeric;	Measured add power (diopters).
INSAX	numeric;	Measured axis (degrees).
INSCTHK	numeric;	Measured center thickness at the Prism Reference Point (mm).
INSCYL	± numeric;	Measured cylinder power (diopters).
INSDIA	numeric;	Measured lens diameter (mm).
INSDRAX	± numeric;	Measured cylinder axis at Distance Reference Point (degrees)
INSDRCYL	± numeric;	Measured cylinder power at Distance Reference Point (diopters).
INSDRPRVA	numeric;	Measured prism base setting at Distance Reference Point (degrees).
INSDRPRVM	numeric;	Measured prism magnitude at Distance Reference Point (LIND or “natural” diopters).
INSDRSPH	± numeric;	Measured sphere power at Distance Reference Point (diopters).
INSFCSGIN	numeric;	Measured lateral Frame Center to Layout Reference Point distance (mm).
INSFCSGUP	numeric;	Measured vertical Frame Center to Layout Reference Point distance (mm).
INSFCYL	numeric;	Measured front curve cylinder (TIND diopters).
INSFRNT	numeric;	Measured front curve (TIND diopters).

INSMOD	literal	Measurement aspect and method: CCF – concave (posterior) surface, focus on axis CCI – concave (posterior) surface, infinity on axis CXF – convex (anterior) surface, focus on axis CCF – convex (anterior) surface, infinity on axis
INSNRAX	numeric;	Measured cylinder axis at Near Reference Point (degrees)
INSNRCYL	± numeric;	Measured cylinder power at Near Reference Point (diopters).
INSNRPRVA	numeric;	Measured prism base setting at Near Reference Point (degrees).
INSNRPRVM	numeric;	Measured prism magnitude at Near Reference Point (LIND or “natural” diopters).
INSNRSPH	± numeric;	Measured sphere power at Near Reference Point (diopters).
INSOCHT	± numeric;	Measured height of Prism Reference Point (mm).
INSPATH	text	Power map file location, see section 5.12.
INSPRAX	numeric;	Measured cylinder axis at Prism Reference Point (degrees)
INSPRCYL	± numeric;	Measured cylinder power at Prism Reference Point (diopters).
INSPRPRVA	numeric;	Measured prism base setting at Prism Reference Point (degrees).
INSPRPRVM	numeric;	Measured prism magnitude at Prism Reference Point (LIND or “natural” diopters).
INSPRSPH	± numeric;	Measured sphere power at Prism Reference Point (diopters).
INSSGIN	± numeric;	Measured lateral distance from LRP to PRP (mm).
INSSGUP	± numeric;	Measured vertical distance from LRP to PRP (mm).
INSSPH	± numeric;	Measured sphere power (diopters).
INSTYPE	integer	1 – focimeter uses 4-point measuring system 2 – focimeter uses ellipse measuring system
IPD	numeric;	Monocular centration distance (mm)
KPRVA	numeric;	Base setting at which to block KPRVM, 0-360 degrees.
KPRVM	numeric;	Magnitude of blocked prism. (degrees or diopters, see record PIND).
LAB	text	Lab identifier.
LAPAX	numeric;	Best-fit toric axis for direct-surfaced jobs. For regular lenses, this should be zero.
LAPBAS	± numeric;	Lap base curve (rounded, TIND diopters).
LAPBASX	± numeric;	Lap base curve (unrounded, TIND diopters). In an LMS packet, this is the “best fit toric” base curve.
LAPBIN	text;	Lap location.
LAPCRS	± numeric;	Lap cross curve (rounded, TIND diopters).

LAPCRSX	± numeric;	Lap cross curve (unrounded, TIND diopters). In an LMS packet, this is the “best fit toric” cross curve.
LAPM	integer;	Integer lap material number. A number agreed to by device and host to access lap material setup tables. The value zero is reserved to mean undefined.
LAPPRB	integer;	Lap probing method: 0 - No HOST control 1 - Normal probing 2 - Re-true probing 3 - Probing OFF
LBLTYP	text[;]	Identifier for label on lens envelope
LDADD	± numeric;	Designed add power (diopters).
LDAX	numeric;	Designed axis at Prism Reference Point (degrees).
LDBCSGIN	± numeric;	Designed lateral distance from Blank Center to LRP (mm).
LDBCSGUP	± numeric;	Designed vertical distance from Blank Center to LRP (mm).
LDCRYPT	integer;integer; or integer	See section 5.9.7.4.
LDCTHK	numeric;	Designed center thickness (mm).
LDCYL	± numeric;	Designed cylinder power at Prism Reference Point (diopters).
LDDRAX	numeric;	Designed Distance Reference Point Cyl Axis (degrees).
LDDRCYL	± numeric;	Designed Distance Reference Point Cyl Power (diopters).
LDDRSPH	± numeric;	Designed Distance Reference Point Sphere Power (diopters).
LDENV	text;	LDS system environment identifier (informational; sent by LDS to LMS)
LDIPD	numeric;	Designed far interpupillary distance (mm).
LDNAM	text;	Designed lens product identifier.
LDNPD	numeric;	Designed near interpupillary distance (mm).
LDNRSPH	± numeric;	Designed Near Reference Point Sphere Power (diopters)
LDNRCYL	± numeric;	Designed Near Reference Point Cyl Power (diopters)
LDNRAX	numeric;	Designed Near Reference Point Cyl Axis (degrees)
LDPATH	text	SDF file location; see section 5.9.3
LDPRVA	numeric;	Designed prism base setting (degrees).
LDPRVM	numeric;	Designed magnitude of prism (LIND or “natural” diopters).
LDPTPRVA	numeric;	Designed prism thinning base setting (degrees).
LDPTPRVM	numeric;	Designed prism thinning magnitude (LIND or “natural” diopters).

LDROT	± numeric;	Rotation of surface matrix from the standard (degrees), see section 5.9.4.1. Normally produced by an LDS, this instructs an FSG to rotate the surface matrix by the specified amount, and notifies other devices of the rotation. This is distinct from SMROT, by means of which an LDS indicates that it has rotated the surface matrix by the amount specified therein.
LDSEGHT	numeric;	Designed layout reference point height (mm).
LDSGAX	numeric;	Designed Layout Reference Point Cyl Axis (degrees)
LDSGCYL	± numeric;	Designed Layout Reference Point Cyl Power (diopters)
LDSGSPH	± numeric;	Designed Layout Reference Point Sphere Power (diopters)
LDSPH	± numeric;	Designed sphere power at Prism Reference Point (diopters).
LDTYPE	literal	Type of LDPATH contents, see section 5.9.3
LDVEN	text[;]	Lens design vendor ID.
LDVER	text[;]	Lens designer software version
LENPRB	integer;	Lens probing method: 0 - No HOST control 1 - Probing OFF 2 - OFF center probing 3 - ON center probing 4 - Pre-cut probing
LENVID	text[;]	Identifier for envelope used to package lens.
LHO	numeric;	Least allowable amount of material removed from uncut blank at edging (lens hangover).
LIND	numeric;	Index of lens material
LMATID	integer;	Integer material number. A number agreed to by the device and host to access material setup tables on both. The value zero is reserved to mean undefined.
LMATNAME	text[;]	Name of lens material (e.g. "GLASS", "PLASTIC")
LMATTYPE	integer;	Integer basic material type. 0 - Undefined/invalid 1 – Plastic 2 – Polycarbonate 3 – Glass 4 – Pattern 5 – Hi-Index 6 – Trivex 7 – Tribrid 8..127 – reserved

LMFR	text[;]	Blank manufacturer
LNAM	text[;]	Name of lens style "SV", "VX INFINITY".
LSIZ	numeric;	Manufacturer's nominal blank diameter (mm).
LTYP *Deprecated, see LTYPE*	literal[,];	Lens type, constructed using the following 2 letter codes. All codes that apply (up to 4) should be concatenated together and separated by commas. For example, an aspheric bifocal should be indicated as AS,BI AS - Aspheric AT - Atoric BI - Bifocal CT - Curve top DS - Double segment EX – E-line multifocal FT - Flat top LT - Lenticular PR – Progressive addition QD - Quadrafocal RD - Round SV - Single vision TR – Trifocal

LTYPE	literal[literal][literal][literal];	Lens type, constructed using the following 2 letter codes separated by spaces. In the case of lenses with differing front and back surface types, as described by LTYPEF and LTYPEB, LTYPE describes the effective combination of both. Exactly one identifier must be selected from for the "Lens Type" set. Depending on the selected "Lens Type", zero or one identifiers are selected from the "Trifocal Power" set, and zero, one or two identifiers are selected from the "Segment Type" set. In the cases of "SV" and "NV", no elements from the second or third sets may appear. In the case of Lens Type "BI", one and only one Segment Type appears. In the case of Lens Type "TR", one or two Segment Types may occur; the only allowable use of two Segment Types is "EX FT" which describes the "E/D' style trifocal. An intermediate power value shall appear when it is other than 50% ("50" never appears). In the case of Lens Type "PR", no Segment Type may appear. In the case of Lens Types "QD" and "DS", one or two Segment Types may appear. When one Segment Type appears, the two segments are the same; when two Segment Types appear, the first describes the upper segment and the second, the lower. Any number of identifiers may appear from the "Special Characteristics" set in conjunction with any combination of Lens Type, Trifocal Power, and Segment Type. The Special Characteristics identifiers AS and AT are mutually exclusive; if one appears, the other may not. Special Characteristics identifiers, when present, must appear in the order in which they are defined herein. Set one: lens type SV – Single vision NV - Near-variable-focus BI – Bifocal TR – Trifocal PR – Progressive addition QD – Quadrafocal DS – Double segment Set two: Trifocal power 40 - 40% intermediate power 60 - 60% intermediate power 70 - 70% intermediate power

LTYPE (cont.)		Set three: segment type CT – Curve top EX – E-line multifocal FT – Flat top RD – Round Set four: Special characteristics AS – Aspheric AT - Atoric LT – Lenticular FF – Digitally Surfaced SO – Slab-off OR - Oriented
LTYPEB		Back surface type, constructed like LTYPE; describes back surface (usually on lenses with one digitally surfaced surface).
LTYPEF		Front surface type, constructed like LTYPE; describes front surface (usually on lenses with one digitally surfaced surface).
MADD	numeric;	Marked add on semi-finished blank.
MAC	text	MAC address of sender.
MAXBACK	numeric	Maximum (steepest) lens back curve allowed (in frame or in lens design; TIND diopters).
MAXDRL	numeric;	Maximum lens thickness at drill feature locations (mm),
MAXFRT	Numeric;	Maximum (steepest) lens front curve allowed (in frame or in lens design; TIND diopters).
MAXGPRVM	numeric[;]	Maximum prism to grind at the generator (sent to LDS). Used in conjunction with MAXKPRVM and PRCPRMODE.
MAXKPRVM	numeric[;]	Maximum prism to block (sent to LDS). Used in conjunction with MAXGPRVM and PRCPRMODE.
MBASE	± numeric;	Nominal lens box top base curve (e.g. 2, 4, 6, etc.)
MBD	numeric;	Minimum blank diameter (mm)
MCIRC	numeric	Measure circumference and cut to tolerance. 0 = no measurement. A value larger than zero should be viewed as a tolerance.
MINAVG	numeric;	Minimum average thickness (mm). (CTHICK + THNP) / 2 must be greater than or equal to MINAVG.
MINBACK	numeric	Minimum (flattest) lens back curve allowed (in frame or lens design; TIND diopters).
MINBCTHK	numeric	Minimum blank center thickness (mm; used in LDS communication).

MINBLOCK	numeric	Diameter of smallest available block in mm; may be conveyed to LDS system.
MINCTR	numeric;	Minimum center thickness (mm). CTHICK must be greater than or equal to MINCTR.
MINDIA	numeric	Minimum blank diameter (mm; used in LDS communication).
MINDRL	numeric;	Minimum thickness at drill feature locations (mm),
MINEDG	numeric;	Minimum edge thickness of finished lens cut to shape (mm). THNP must be greater than or equal to MINEDG.
MINFRT	Numeric;	Minimum (flattest) lens front curve allowed (in frame or lens design; TIND diopters).
MINTHKCD	numeric;	Minimum edge thickness at crib shape.
MIXBLKOK	Integer	0 – LDS must select same surface blocks on both eyes. 1 – LDS may mix surface blocks.
MPD	numeric	"Frame mechanical D" or "distance between centers"; equals left and right HBOX divided by two, plus DBL, plus any compensations applied by host.
NOD	numeric;	Refracted object distance at near viewing point (meters).
NPD	numeric;	Monocular near centration distance (mm).
NOTETYP	text[;]	Delivery note layout identifier
NREF	eld	Reference wavelength. Indicates whether the mercury (*e*-line) or helium (*d*-line) wavelength is used in determining lens material index. Literals lower-case "e", "d", and the unknown data indicator, are the only permissible field values.
NWA	numeric;	Near working angle - elevation of near viewing point (degrees; above/below horizon, positive indicating above).
NWD	numeric;	Working object distance at near viewing point (meters).
OCHT	numeric;	Vertical Prism Reference Point height measured from the (frame) lower boxed tangent
OCR	integer	Oblique Central Refraction compensation calculation indicator. 0 – Don't modify Rx for effects of OCR 1 – Modify Rx for effects of OCR
OPC	text;	10 digit optical product code (OPC) for lenses to be used.
OPCF	text;	Front wafer 10 digit product code.
OPCB	text;	Back wafer 10 digit product code.
OPTFRNT	numeric;	Optimal front curve (TIND diopters), returned by LDS in BRS exchange (see section 5.9.1).
OTHK	numeric;	Generator thickness at the prism reference point (mm). Used in conjunction with SBOCIN/SBOCUP.
PADTHK	numeric;	Pad thickness (mm).

PANTO	numeric[;]	Pantoscopic angle (degrees).
PBOK	integer;	Prism blocking OK flag. 0 = Prism blocking not allowed 1 = Prism blocking allowed (default – when no value is specified, prism blocking is allowed).
PHOTO	integer	Photochromic flag 0 = Lens is not photochromic. 1 = Lens i photochromic.
PINB	numeric;	Edger back pin bevel width in mm measured along cut surface of lens (0 = no pin bevel).
PINBS	integer;	Special pin bevel indicator: 0 = no special pin bevel. 1 = heavy chamfer.
PIND	numeric;	The index of prism value, if prism expressed in diopters, otherwise the value should be zero if prism is expressed in degrees.
PLRCRV	numeric;	Curvature of polarizing or other film (TIND diopters).
PLRLOC	numeric;	Location of polarizing or other film behind front surface of blank at blank center (mm).
POLAR	integer	Lens is polarized: 0 = not polarized. 1 = polarized.
POLISH	integer	Polishing mode. 0 = no polish, 1 = polish edge and pin bevels. 2 = polish pin bevels only. 3 = polish edge only. +128 indicates high luster.
PREEDGE	integer;	Enable=1/Disable=0. Pre-edging of lens by generator.
PRCPRMODE		Used in conjunction with MAXGPRVM and MAXKPRVM, indicates order in which required prism is to be achieved. Sent by LMS to LDS. 0 – prism may be generated or blocked in its entirety. 1 – generate prism up to MAXGPRVM, block remainder up to MAXKPRVM. 2 – block prism up to MAXKPRVM, generate remainder up to MAXGPRVM.
PRI	integer	Job priority, sent to LDS; 0 – normal priority, 1 – elevated priority.

PROC	literal;literal;text;text; text;text	Process data record. The first field specifies either 1) a user-defined record label that identifies the process information contained in subsequent fields, or 2) a standard label; the second field specifies the device type to which the process applies; the third field optionally narrows the application to a specific vendor; the fourth field optionally narrows the application to a specific model for the vendor specified in the third field; the fifth field contains the process-specific data for the right lens (or for both lenses in the case of non-chiral data); the sixth field contains the process-specific data for the left lens (in the case of chiral data).
PRVA	numeric;	Rx Prism base setting, 0 – 360 (degrees).
PRVM	numeric;	Rx Prism in diopters, including equithinning prism to inspect at Prism Reference Point. (LIND or “natural” diopters).
PTOK	integer	Prism Thinning Allowed specifier: 0 = Prism Thinning not allowed 1 = Prism Thinning allowed (default – when no value is specified, prism thinning is allowed).
PTPRVM	numeric;	Thinning Prism (If PTPRVM=0, then PRVM will be Rx Prism only; if PTPRVM>0, then PRVM will be the resultant of Rx Prism & PTPRVM) (LIND or “natural” diopters).
PTPRVA	numeric;	Thinning Prism base setting, 0 – 360 (degrees).
PUBK	text	Public key used for encrypted communications.
R	integer;integer;....	Radius data, used in datasets (e.g., TRCFMT).
RCRIB	numeric;	Smallest circular crib (may differ from CRIB when ELLH is less than CRIB).
RNGCMP	± numeric;	Thickness compensation for prism or blocking ring thickness (mm). This compensation is applied to GTHK.
RNGD	numeric;	Ring diameter (mm).
RNGH	numeric;	Ring height (mm).
RPRVA	numeric;	Base setting at which to generate RPRVM, 0-360 (degrees).
RPRVM	numeric;	Amount of prism to generate when using the SBOCIN/SBOCUP records for decentration (diopters or degrees, see record PIND)
RVD	numeric;	Back vertex distance, as refracted (mm).
RXNM	text	RX number
S	integer;integer;...	Radius data for the interior shape of the shelf. If a given radius is zero, then do not shelve this point and ignore all associated measurements for this point (SA, SD, SW, etc.)
SA	integer;integer;...	Angle data for shelf radius data (angular locations of shelf radii expressed in S record(s)).
SAGBD	± numeric;	Sag at blank diameter (mm). Indicates the sag of the lens front surface at its edge prior to cribbing.

SAGCD	± numeric;	Sag at crib diameter (mm). Indicates the sag of the lens front surface at its edge after cribbing.
SAGRD	± numeric;	Sag at ring diameter (mm). Indicates the sag of the lens front surface in the specified blocking ring diameter (RNGD).
SBBCIN SBBCUP	± numeric; ± numeric;	Surface block to blank center vectors. +IN means blank center is towards nasal relative to the surface block. -IN means blank center is towards temporal relative to the surface block. +UP means blank center is above the surface block. -UP means blank center is below the surface block.
SBCCIN SBCCUP	± numeric; ± numeric;	Surface block to cribbing center vectors. +IN means cribbing center is towards nasal relative to the surface block. -IN means cribbing center is towards temporal relative to the surface block. +UP means cribbing center is above the surface block. -UP means cribbing center is below the surface block.
SBEV	numeric;	Generator safety bevel depth (in Z plane) to cut (mm).
SBFCIN SBFCUP	± numeric; ± numeric;	Surface block to frame center vectors, i.e. indicates the position of the center of the frame SHAPE relative to the surface blocking position, so that the SHAPE may be shown in its final relative position for determining cutout. +IN means frame center is towards nasal relative to the surface block. -IN means frame center is towards temporal relative to the surface block. +UP means frame center is above the surface block. -UP means frame center is below the surface block.
SBLKCC	text[;]	Back surface block name.
SBLKCX	text[;]	Front surface block name.
SBOCIN SBOCUP	± numeric; ± numeric;	Surface block to prism reference point vectors. Instructs the generator to grind the prism reference point at a position relative to the surface block. +IN means PRP is towards nasal relative to the surface block. -IN means PRP is towards temporal relative to the surface block. +UP means PRP is above the surface block. -UP means PRP is below the surface block.
SBSGIN SBSGUP	± numeric; ± numeric;	Surface block to layout reference point vectors. Position of the layout reference point relative to the center of the surface block. +IN means LRP is towards nasal relative to the surface block. -IN means LRP is towards temporal relative to the surface block. +UP means LRP is above the surface block. -UP means LRP is below the surface block.

SDEPTH	numeric;	Segment depth (see Figure 2 ISO 13666:1998)
SD	integer;integer;...	Depth data for a shelf as viewed from the lens edge. Distance in mm from the lens front or back as specified in the 5th field of the SHLFFMT. If SD=0, then do not shelve this point. If this data is supplied, the number of SD points must be equal to the number of S points.
SDLPATH	text	SDL file location; see section 5.9.5
SEGDHT	numeric;	Layout Reference Point height measured along a vertical line segment from the LRP center down to the point at which it intersects with the edge of the lens or shape.
SEGHT	numeric;	Layout Reference Point height measured from the LRP center to the lower horizontal element of a box circumscribing the shape.
SF	numeric;	Shelf angle (degrees). Tilt of the edge of the shelf, which is otherwise parallel to the lens spindle. A positive number indicates an inclination towards the machine spindle on the side of the lens on which the shelf is produced.
SGOCIN SGOCUP	± numeric; ± numeric;	Layout reference point to prism reference point vectors. Can be used to locate the PRP for multifocals. +IN means PRP is towards nasal relative to the LRP. -IN means PRP is towards temporal relative to the LRP. +UP means PRP is above the LRP. -UP means PRP is below the LRP.
SHLFC	numeric;	Curvature of shelf edged on lens (1.53 diopters); see section 5.7.
SHLFM	numeric;	Shelf modifier. Percentage or distance in mm based on value of SHLFP. Percentage should be expressed as a number between 0 and 100.
SHLFP	integer[;]	Shelf position 0 – Manual 1 – Follow front (SHLFM = mm from front) 2 – Percent back (SHLFM = % from front) 3 – Frame curve (SHLFC=curve to follow, SHLFM=mm from front) 4 – 50/50 5 – Follow back (SHLFM = mm from back) 6 – Follow bevel 7 – Automatic (Use edger settings) 8 – Follow shelf depth data (SD=depth data record to follow and correlates with the number of S records) 9 - Unused

SLBP	numeric;	Slaboff prism (diopters)
SLHT	numeric;	Slaboff line height (measured from frame lower edge)
SLDRP	numeric;	Slab off drop. Similar to SLHT, but can be used for uncuts (mm). Measured from blank center; a down vector is positive.
SMDMAX	numeric; numeric; numeric; numeric	A set of values expressing constraints on surface matrix decentration, sent by an LMS to an LDS. The fields specify the maximum decentration in, out, up and down respectively. Does not apply when SMOCIN/UP are specified in the LDS file.
SMOCIN, SMOCUP	numeric;	Center of Surface Matrix to PRP in and up (mm). Instruction from LMS to LDS to decenter the location of the PRP in the surface matrix. The PRP is by default located at the center of the surface matrix.
SMROT	numeric[;]	Instruction from LMS to LDS to rotate surface matrix by the specified number of degrees; notification from LDS to LMS and devices that the surface matrix has been rotated. This is distinct from LDROT, by means of which an LMS or LDS can command a generator to rotate a surface matrix by the number of degrees specified therein.
SPEED	integer;	Integer number agreed to by the device and host to access speed control tables on both. The value zero is reserved to mean undefined.
SPH	± numeric;	Rx Sphere power (diopters)
SVAL	numeric;	Value indicating the thickness of a lens after generating measured relative to the machine's reception chuck on the centerline of the chuck
SW	integer;integer;intege r…	Width data for a shelf as viewed from the lens front. Distance in mm outward from the corresponding S record and along the same axis. If SW=0 or if this data is not present, then the device should shelf all the way out from the corresponding S record to the lens edge if the corresponding S record is greater than zero. If this data is supplied, the number of SW points must be equal to the number of S points. SW records are only needed if a shelf does not go all the way to the edge of the lens at any shelf point. SW records, for example could be used to define a groove that should be cut on the front or back lens surface.
SWIDTH	numeric;	Segment width (mm)
T	integer;integer;intege r…	Lens edge thicknesses (in units of 0.01mm), used in datasets (e.g., STHKFMT).
TFADD	numeric;	True add power on front of semi-finished blank (TIND diopters).
THKA	numeric;	Thick point angle—the meridian at which THKP occurs. (degrees)
THKCMP	± numeric;	General purpose thickness compensation (mm). This compensation is applied to GTHK.
THKP	numeric;	Thickest edge thickness on finished edge. (mm)
THKR	numeric;	Length of radius, measured from Frame Center, or geometric center of uncut lens, at which THKP occurs. (mm)
THNA	numeric;	Thin point angle—the meridian at which THNP occurs on finished edge. (degrees)
THNP	numeric;	Thinnest edge thickness on finished edge. (mm)

THNR	numeric;	Length of radius, measured from Frame Center, or geometric center of uncut lens, at which THNP occurs. (mm)
TILTBASE	numeric[;]	Panoramic angle of the bevel relative to the front of the lens expressed in degrees. The angle is formed at the apex of a lateral projection of the bevel at the nasal.
TIND	numeric;	Index of refraction used for all diopter curves in a packet. In the absence of a TIND record or field value, 1.53 may be presumed.
TIME	text (formatted)	UTC time and date in the format YYYYMMDDHHNNSS, where YYYY – year MM – month DD – day HH – hour NN – minute SS – second
TINT	text[;]	Tint abbreviation
TNORM	integer	? – state of radius data with regard to z axis is unknown. 0 – tracing is normalized for tilt (vertical positions of points are indicated by Z-data or FCRV). 1 – tracing is tilted laterally as specified in ZTILT 2 – tracing is "raw"; vertical positions are indicated by Z-data. 3 – tracing is flattened to 2-dimensional, planar form, devoid of tilt or curve. FCRV or Z data, and ZTILT, specify original measurements.
TOKEN	text	Authentication token (for lens design request)
TPERR	numeric, numeric, numeric[;]	Through-power error. Deviation of the image of back-side reference marks when viewed through the lens. The first field contains the lateral deviation (x in mm), the second, the vertical deviation (y in mm), and the third, the rotational shift (Θ in degrees).
TPSIZ	integer	(Side) Temple length of frame (mm)
TPTYP	text	(Side) Temple type of frame such as cable or straight
TZ	integer	Time zone indicator; minutes before (-) or after (+) GMT.
UNI	integer	Patient vision. A value of 1 means the patient is monocular or only has sight in a single eye. A value of 0 indicates binocular vision.
VBOX	numeric[;]	Vertical boxed lens size (mm)
VIEWP	literal; numeric; numeric; numeric	Viewing point location. The first field identified the side (R, L or B); the second field contains the working object distance in millimeters; the third field contains the elevation of the viewing point above (+) or below (-) the horizon (degrees); the forth field contains the lateral location of the viewing point (expressed as inset from IPD in millimeters).
VMAP	text	Information provided by fitting instrument, used by LDS.
VTOLFCYL	numeric;	Magnitude of front cylinder tolerance (LIND diopters).

VTOLFRNT	numeric;	Magnitude of front curve tolerance (LIND diopters).
WDFPATH	text	Weighting matrix file location, see section 5.13.
WEIGHT	numeric;	Weight of lens (estimated) (grams).
XMIN	numeric;	Minimum transmission of gradient lens (for solid color, XMIN=XMAX)
XMAX	numeric;	Maximum transmission of gradient lens (for solid color XMIN=XMAX)
Z	integer;integer;...	Sag data (trace Z dimension, used in ZFMT dataset).
ZA	integer;integer;…	Angle data for sag data (used in ZFMT dataset)
ZZ	numeric;numeric…	Surface height data, see 5.9.4.3 (used in SURFMT dataset)
ZTILT	numeric[;]	Side-to-side or horizontal tilt of frame as traced.

Table A.1a – Tolerance records

With the exception of those whose descriptions are marked with an asterisk in this Table, all tolerance records share the same descriptions:

0 – Failed inspection

1 – Passed inspection

9 – Not tested

Those records for which a corresponding TOLV record are allowed are indicated in the fourth column of the table.

TOLV records are multi-value chiral records, and have the form (using TOLVADD for example, with an ADD value of 2.00, i.e., ADD=2.00;2.00):

TOLVADD=-0.4l0.6;-0.4l0.6

The two values in each field are the deviations establishing the lower and upper bound for the allowable range for the corresponding record (ADD in the example; in each case, the ADD power must be between 1.96 and 2.06).

NOTE: Measured values are contained in INS records

Record Label	Data type	Description	TOLV
TOLADD	integer;	Tolerance of add power.	x
TOLASPEC	integer;	Tolerance of aspect (cosmetics/appearance).	
TOLAX	integer;	Tolerance of axis.	x
TOLCTHK	integer;	Tolerance of center thickness at the Prism Reference Point.	x
TOLCYL	integer;	Tolerance of cylinder power.	x
TOLDIA	integer;	Tolerance of lens diameter.	x
TOLFCYL	integer;	Tolerance value on front cylinder.	x
TOLFRNT	integer;	Tolerance of front curve inspection.	x

TOLHC	integer;	Tolerance of hard coat.	
TOLIPD	integer;	Tolerance of distance PD.	x
TOLNPD	integer;	Tolerance of near PD.	x
TOLOCHT	integer;	Tolerance of prism reference point height.	x
TOLPOLAR	integer;	Tolerance of polarization.	
TOLPRVA	integer;	Tolerance of prism angle.	x
TOLPRVH	integer;	Tolerance of horizontal prism.	x
TOLPRVHI	integer;	Tolerance of prism horizontal imbalance.	x
TOLPRVM	integer;	Tolerance of prism magnitude.	x
TOLPRVV	integer;	Tolerance of vertical prism.	x
TOLPRVVI	integer;	Tolerance of prism vertical imbalance.	x
TOLREFL	integer;	Tolerance of reflectivity.	x
TOLSET	integer	*Numerical Tolerance Table Identifier	
TOLSGIN	integer;	Tolerance of segment in horizontal direction from O.C.	x
TOLSGUP	integer;	Tolerance of segment in vertical direction from O.C.	x
TOLSHAPE	integer;	Tolerance of shape cutout.	x
TOLSPH	integer;	Tolerance of sphere power.	x
TOLSTD	literal	*Tolerance procedure identifier (ISO, ANSI, BSI, JIS, CUSTOM)	
TOLXMIT	integer;	Tolerance of transmission.	x
TOLXMITB	integer	Tolerance of transmission balance.	x

NOTE: For the records so marked, corresponding TOLV records exist, viz., TOLVADD, TOLVAX, TOLVCTHK, TOLVCYL, TOLVDIA, TOLVFCYL, TOLVFRNT, TOLVIPD, TOLVNPD, TOLVOCHT, TOLVPRVA, TOLVPRVH, TOLVPRVHI, TOLVPRVM, TOLVPRVV, TOLVPRVVI, TOLVREFL, TOLVSGIN, TOLVSGUP, TOLVSHAPE, TOLVSPH, TOLVXMIT, TOLVXMITB.

A.2 Interface records

A.2.1 Table A.2 contains all of the interface records defined for use by systems conforming to this Standard, arranged alphabetically. The data format is the same as in Table A.1.

Table A.2 – Interface records

Record label	Data type	Description
ANS	literal or integer	Answer type, see record REQ. This appears in all except initial request packets to echo the request type specified in the request packet.
ACT	literal;literal	Action specified in CMD request; see section 7.6
CANDO	literal [;literal…]	List of capabilities of LDS software, used in LDI file. Initial allowed value is "BRS", indicating that LDS can process BRS (blank selection) requests.
CRC	integer	Cyclical-redundancy check. See Annex C for a description of how this is calculated.
CRIBFMT	integer ; integer ;UIE ;RI LIB	Crib dataset header, see section 5.10. CRIBFMT=#;###; EIU; RILIB a) The first field is the format identifier: 0 - No crib dataset available 1 - Basic ASCII radii format 2 - BINARY absolute radii format 3 - BINARY differential format 4 - PACKED BINARY format 5..100 - Reserved for future standard formats. b) The second field is the number of radii in which the data shall be expressed ; c) The third field is the radius mode identifier: "E" indicates that radii are evenly spaced ("equiangular"); "U" indicates that radii are unevenly spaced, so that an angle data "A" record must follow the "R" record. d) In initialization or request packets, the fourth field indicates which eye(s) are included: R)ight, L)eft, or B)oth. In data packets, it specifies the orientation of the data.
D	literal;literal ;…	Record label list for auto-initialization. This is used in conjunction with the DEF and ENDDEF records to define the set of records to be associated with an auto-format request. The record has the form D=label;label;label . . .<CR/LF> and can contain one or more record labels. Multiple D records may occur between the DEF and ENDDEF records. This is used to identify to the host such records as are necessary for the device to operate properly.

DEF	limited;integer	Request definition. This is used in the initialization data packet transmitted from the device to the host to indicate the beginning of a list of records that should be assigned a request ID by the host. An ENDDEF record terminates the list. In a device's transmission, the record has the form: DEF = *Device tag* <CR/LF> where <*Device tag*> is a limited string used by the device to identify the request type. In the host's initialization data packet, it has the form DEF = *Device tag* ; request ID <CR/LF> where request ID is an integer that the DEVICE must use when it wants to make the kind of request it defined for device tag. The device tag may be blank if preset initialization is being performed. The HOST sends a single DEF record per request definition transmitted by the DEVICE. In a device's initialization data packet, the "D" (record label list) record(s) shall immediately follow the request definition record.
DEV	literal	Device type. This record contains one of the literal data choices enumerated in Table A.4, which identifies the kind of device and the corresponding set of device records which should be conveyed. It can also be used in auto-format initialization to indicate the direction of data flow. This appears in initialization packets.
DO	RILIBIN	This record identifies which lens in a pair is to be processed regardless of presence of other kinds of right/left data. R)ight, L)eft, B)oth, or N)one indicates which eye the device should process.
DRLFMT	BICIE	Drill format negotiation record, used in request packets or initialization sessions. See section 5.5.3.
DSDEV	literal;limited;limited;text;text	Direct Surface Device SDF file identifier. See section 5.9.2.2
ENDDEF	limited	End of request definition. This is used to mark the end of a request definition during auto format initialization begun by a request definition (DEF) record. The data field must contain the matching device tag to the DEF record.
HID	limited	Host identification. The host ID is limited data, which may optionally appear in a request packet. It is echoed in any packets sent by a host to a device during a session begun with a request packet including such a record.
HMODEL	text	Host software model identifier, which may appear in a host's data packet sent to a device during initialization.
HNAME	text	Host software name identifier, which may appear in a host's data packet sent to a device during initialization.
HSN	text	Host software serial number identifier, which may appear in a host's data packet sent to a device during initialization.
HVEN	text	Host software vendor identifier, which may appear in a host's data packet sent to a device during initialization.
HVER	text	Host software version identifier, which may appear in a host's data packet sent to a device during initialization.
INFO	integer	Information packet control. This is used by hosts to control the transmission of INF requests. When a device indicates INFO=1 in a request or initialization packet, a host can suppress INF requests by specifying INFO = 0 in its data packet.

IP	text	IP address. Used in the Autoconnect packet described in section 7.8.2.7.1.
JOB	limited	Job number. Also called job ID. The JOB ID is limited data that appears in all packets outside of initialization. A device should verify that job ID's in packets received match the job ID originally specified in its request packet. Likewise, hosts should take care to return the job ID's specified in request packets in the same format as was received. Job ID's are case sensitive when they include letters. Upload devices that cannot provide job identification shall transmit a field value consisting of the unknown data indicator ("?").
LIB	text	Library name under which to store trace.
MESG	text	This is text data conveyed in either direction for any purpose. A device can inform a host of the maximum message length that it can handle by means of the message length (MSL) record.
MID	limited	Machine identification (ID). This is limited data that should uniquely identify a device among all those connected to a particular host, at least within a given set of device types and machine models. It is recommended that devices allow the MACHINE ID to be set in the field. It is optional in initialization or request packets.
MNAME	text	Machine name. This is optional TEXT data that can be used by HOSTS to identify DEVICES using names that will be meaningful to users, such as "ACME Lens Generator Model 101“.
MODEL	limited	Machine model. This is useful in initialization to allow hosts to determine what parameters would be appropriate to send to a machine. It is optional in initialization or request packets to allow a host to refine its handling of preset requests.
MSL	integer	Initialization record to set the maximum message length. This is to inform the host that the device can only display messages of a certain length.
MVER	text	Device software version
OMAV	MM.mm	Interface version. This is used by hosts to match the structure of its packets to the interface versions of devices. It is optional in request or initialization packets. The structure of the interface version will be “MM.mm“ where MM is the major version identifier and mm is the minor version identifier. A change to the major identifier shall occur when changes to the interface occur which could affect the ability of hosts and devices to communicate. Changes to the minor version identifier shall occur when additional records are defined and when other changes occur that should not affect the ability of hosts and devices to communicate.
OPERID	limited	Operator Identification. This can be used by a host to provide statistical data related to machine operators. It is limited data and may optionally appear in request or initialization packets.
PORT	integer	Port number. Used in the Autoconnect packet described in section 7.8.2.7.1.
REM	text	Remarks. This is text data that can appear in any packet. Neither hosts nor devices take any action based on remarks. Remarks are *not* echoed back to the sender in any ensuing packets during a session.
REQ	literal or integer	Request type. This appears in the request packet used to identify to the host the kind of request being made. Request type contains a value consisting of one of the literal data choices enumerated in Table A.3 or a request ID number returned from a host during Initialization. REQ must be the first record in a request packet.

RMT	Text	Information which could be used to link tracings with Rx orders delivered by some means other than that by which the tracing is delivered.
SDFMODE	text|literal	Specifies the method for delivery of SDF data to direct-surfacing devices, and the content of the LDPATH record. When absent, the literal FILE is assumed. FILE (literal) – LDPATH contains a fully-qualified path name to a file to which the device has access, and from which the device retrieves SDF data LMS (literal) – LDPATH contains a fully-quailifed path name to a file to which the host has access, from which the host retrieves SDF data, which it delivers to the device in its data packet. Web address (text) – SDFMODE contains a Uniform Resource Identifier beginning with "http://", indicating that SDF data is to be retrieved from a web service; LDPATH contains an identifier produced by the LDS, and used in the (non-standardized) method call to the web service. FTP address (text) – SDFMODE contains a Uniform Resource Identifier beginning with "ftp://", indicating that SDF data is to be retrieved from an ftp service; LDPATH identifies the file containing SDF data

SHLFFMT	integer;integer; UIE ;RILIB	Shelf data format. SHLFFMT =#;###; EIU; RILIB;FIRIB a) The first field is the format identifier: 0 – No shelf radii available 1 – Basic ASCII radii format 2 – BINARY absolute radii format 3 – BINARY differential format 4 – PACKED BINARY format 5..100 – Reserved for future standard formats. b) During initialization, the second field is the preferred number of radii in which the shelf should be expressed. During actual operation, the actual number of shelf points may vary widely depending on the shape and size of the shelf. c) The third field is the radius mode identifier: “E“ indicates that the shelf radii are evenly spaced (“equiangular“). “U“ indicates that radii are unevenly spaced, so that an angle data “SA” record must follow the “S” record. d) In initialization or request packets, the fourth field indicates which eye(s) are included: R)ight, L)eft, or B)oth. In data packets, it specifies the orientation of the shelf data. e) In initialization, the fifth field allows a device to specify that it can shelf the (F)ront, (B)ear, or (E)ither side(s) of a lens. In data packets, this field specifies the surface for the shelf data. “F” indicates the shelf is on the front of the lens. “R” indicates the shelf is on the rear of the lens. f) This field indicates the shelf should be created by removing material from F)ront, or B)ack surface of the lens. If this field is absent, the B)ack surface is assumed.
SN	text	Device serial number. This is text data that shall contain a device’s serial number, i.e., an identifier given the device by its manufacturer. It is expected that this identifier would be useful to the manufacturer for service and/or diagnostic purposes. Its uniqueness is not guaranteed. It is optional in initialization or request packets.
STATUS	integer[;text]	Status code (described in A.4).

STHKFMT	integer ;integer ;UIE ;RILIB	Surface thickness dataset header, see section 5.11. STHKFMT=#;###; EIU; RILIB a) The first field is the format identifier: 0 - No trace available 1 - Basic ASCII radii format 2 - BINARY absolute radii format 3 - BINARY differential format 4 - PACKED BINARY format 5..100 - Reserved for future standard formats. b) The second field is the number of radii in which the tracing shall be expressed ; c) The third field is the radius mode identifier: "E“ indicates that radii are evenly spaced ("equiangular“); "U“ indicates that radii are unevenly spaced, so that an angle data "A" record must follow the "R" record. d) In initialization or request packets, the fourth field indicates which eye(s) are included: R)ight, L)eft, or B)oth. In data packets, it specifies the orientation of the tracing.
SURFMT	literal; literal; literal; integer; integer; integer; integer[;integer] [;integer]	Surface data format. See section 5.9.4.
TXTENC	literal[;literal]	Encoding specifier for TEXT and LIMITED fields. See Section 5.1.7.4. Two encoding schemes are presently normative, and their literals are, shown in order of complexity: ASCII UTF-8
TIMEOUT	integer;integer; integer	This record has the form TIMEOUT = *confirmation* ; p*acket* ; *intercharacter*. It may be sent from a host to a device during initialization to override the pre-defined timeout values. If a device does not allow such modification, the fact that it does not should be clearly stated in the device’s documentation. A device which does not support altering timeout values should not return an error status if it receives a TIMEOUT record. Timeout values are in seconds and must be between two (2) and two hundred fifty five (255) seconds.F

TRCFMT	integer ;integer ;UIE ;RILIB ;FI PID	Trace dataset header. TRCFMT=#;###; EIUIC RILIB;FIPID a) The first field is the format identifier: 0 - No trace available 1 - Basic ASCII radii format 2 - BINARY absolute radii format 3 - BINARY differential format 4 - PACKED BINARY format 5..100 - Reserved for future standard formats. b) The second field is the number of radii in which the tracing shall be expressed ; c) The third field is the radius mode identifier: "E“ indicates that radii are evenly spaced ("equiangular“); "U“ indicates that radii are unevenly spaced, so that an angle data "A" record must follow the "R" record. “C” indicates the “creative” mode, the variant of unevenly-spaced radii wherein the angles are not required to increase sequentially (see section 5.4.3.1). d) In initialization or request packets, the fourth field indicates which eye(s) are included: R)ight, L)eft, or B)oth. In data packets, it specifies the orientation of the tracing. e) The fifth field indicates what has been traced: F)rame, P)attern, or D)emo lens. The fifth field is not present when the TRCFMT record is sent during initialization. All five fields must be sent on subsequent upload and download sessions unless the first field is 0. In that case, only the first field is sent.
VEN	limited	Vendor identification (ID). This contains data used to identify a device’s manufacturer. If the vendor ID is not present in a device’s initialization data packet, the host may not be able to transmit setup parameters to the device.
XSTATUS	RILIB;integer; text; EIWIN	One or more XSTATUS records may appear, containing data specific to one or both lenses (see section A.5).
ZFMT	Integer ;integer ;UIE ;RILIB ;FI PID	Z dimension format for trace data. Identical to TRCFMT.

Table A.3 – Request types (literal data)

Request type	Desired action
INI	Initialization
TRC	Frame trace upload
PTG	Pattern generator download
EDG	Edger download

SBK	Surface blocker download
FBK	Finish blocker download
AGN	Laminator download
BAS	Blank Selection Response (to LMS from LDS)
BRS	Blank Selection Request (to LDS from LMS)
CMD	Command Request
COA	Surface coater download
CON	Network connection request
DNL	Generic download
DRL	Drill download
ENG	Lens engraver download
ERR	Error answer response
FSG	Direct surface generator download
FSP	Direct surface polisher download
GEN	Surface generator download
INF	Information upload
INK	Inking device download
INS	Inspection device upload
LAP	Lap feeder download
LDI	LDS initialization file identifier
LDS	Lens Design System download
LMD	Lens measuring device download
LMI	LMS initialization file identifier
LMS	Host upload from LDS
LNF	Lens information request
MNT	Maintenance upload
POL	Surfacer download
Request ID (integer)	Action as determined by preset or auto-format initialization
SDF	Surface Definition File Identifier
TIM	Time Synchronization Request
UPL	Generic upload

Table A.4 – Device types

Device	Device type (literal data)
Frame tracer	TRC
Pattern generator	PTG
Lens edger	EDG
Surface blocker	SBK
Finish blocker	FBK
Surface generator	GEN
Laminator	AGN
Generic upload device	UPL
Generic download device	DNL
Surface coater	COA
Direct surface generator	FSG
Lens measuring device	LMD
File	FIL
Inspection Device	INS
Lap feeder	LAP
Drill	DRL
Direct surface polisher	FSP
Surfacer	POL
Lens engraver	ENG

A.3 Preset packets

Preset packets describe sets of records to be downloaded when auto-initialization is not performed. In the case of upload devices, the packets describe records to be uploaded. For exchanges requiring tracing data, the phrase "(tracing data)" is included in the lists of records below, and the trace format negotiation shall be performed according to 5.4.1.

The following formats are used for all the preset definitions.

*LABEL Mandatory, should always be sent. The asterisk IS NOT SENT in the actual transmission. All labels are functionally mandatory starting with OMAV 3.02 (in that all records listed in an initialization session's record label list, or in a preset packet, must be sent to a device, using the unknown data indicator if necessary), but the list of previously mandatory labels may be necessary for backwards compatibility.

NOTE: For datasets, the set of records sent (R, A, Z, ZA) depends on the format used.

For informational purposes, for labels added in version 3.02 or higher, the version in which each label was added is indicated by a trailing subscript. Pursuant to Newer

$\text{LABEL}_{3.02}$ Should be sent if the device OMAV is 3.02 or higher.

LABEL$_{3.03}$ Should be sent if the device OMAV is 3.03 or higher.

Table A.5 – Preset Frame Tracer (“TRC”) packet

Record label	See Table A.1 for record descriptions
(tracing data)	
BSIZ	
*CIRC	
CSIZ	
*DBL	
FCRV	
FTYP	
HBOX	
TNORM$_{3.02}$	
VBOX	
ZTILT	

Table A.6 – Preset Pattern Generator (“PTG”) packet

Record label	See Table A.1 for record descriptions
(tracing data)	
TNORM$_{3.02}$	

Table A.7 – Preset Lens Edger (“EDG”) packet

Record label	See Table A.1 for record descriptions
(tracing data)	
BEVP	BEVM and BEVC shall also be included in the response if they are necessary, depending upon BEVP.
BSIZ	
*CIRC	
CLAMP$_{3.02}$	
CSIZ	
*DBL	
*DIA	
DRILL$_{3.02}$	
DRILLE$_{3.04}$	Format negotiation via DRLFMT required.
EPRESS$_{3.02}$	
ERDRIN$_{3.03}$, ERDRUP$_{3.03}$	

$ERNRIN_{3.03}$, $ERNRUP_{3.03}$ ERSGIN, ERSGUP *ETYP *FBFCIN, *FBFCUP FBSGIN, FBSGUP FCRV $FPINB_{3.02}$ *FTYP $GDEPTH_{3.02}$ $GWIDTH_{3.02}$ *IPD *LMATTYPE LMATID *LTYP $LTYPE_{3.03}$ $MCIRC_{3.02}$ *NPD *OCHT *PINB *POLISH *SEGHT $TNORM_{3.02}$ ZTILT	

Table A.8 – Preset Finish Blocker (“FBK”) packet

Record label	See Table A.1 for record descriptions
(tracing data)	
*AX CYL *DBL *DIA $ERDRIN_{3.03}$, $ERDRUP_{3.03}$ $ERNRIN_{3.03}$, $ERNRUP_{3.03}$ ERSGIN, ERSGUP *FBFCIN, *FBFCUP	

*FBOCIN, *FBOCUP	
FBSGIN, FBSGUP	
FCOCIN$_{3.02}$, FCOCUP$_{3.02}$	
FCSGIN$_{3.02}$, FCSGUP$_{3.02}$	
*HBOX	
*IPD	
*LTYP	
LTYPE$_{3.03}$	
*NPD	
*OCHT	
PRVA$_{3.02}$	
PRVM$_{3.02}$	
SDEPTH	
SEGHT	
SGOCIN, SGOCUP	
SPH	
SWIDTH	
*VBOX	

Table A.9 – Preset Surface Blocker (“SBK”) packet

Record Label	See Table A.1 for record descriptions
*BACK	
BCSGIN, BCSGUP	
*BCTHK	
*BETHK	
*BPRVA	
*BPRVM	
*DIA	
*FRNT	
*GAX	
*GPRVA	
*GPRVM	
*IFRNT	
*KPRVA	
*KPRVM	

*LTYP $LTYPE_{3.03}$ $OPC_{3.02}$ *SAGRD *SBBCIN, *SBBCUP *SBFCIN, *SBFCUP SBOCIN, SBOCUP SBSGIN, SBSGUP SDEPTH SWIDTH	

Table A.10 – Preset Surface Generator (“GEN”) packet

Record label	See Table A.1 for record descriptions
*AVAL *BACK *BCTHK *BETHK *BLKB *BLKCMP *BLKD *BLKTYP $CPRVA_{3.02}$ $CPRVM_{3.02}$ *CRIB *DIA EECMP *ELLH FINCMP *FLATA *FLATB *FRNT *FTTHK *GAX *GBASE *GBASEX	

*GCROS	
*GCROSX	
*GPRVA	
*GPRVM	
*GTHK	
*IFRNT	
*KPRVA	
*KPRVM	
*LAPBAS	
*LAPBASX	
*LAPCRS	
*LAPCRSX	
LAPM	
*LAPPRB	
*LENPRB	
*LIND	
*LMATTYPE	
*LMATID	
*LSIZ	
*LTYP	
$LTYPE_{3.03}$	
*OTHK	
PADTHK	
*PIND	
$PREEDGE_{3.02}$	
*RNGD	
*RNGH	
RNGCMP	
RPRVA	
RPRVM	
*SAGBD	
*SAGCD	
*SAGRD	
*SBBCIN, *SBBCUP	
*SBEV	
SBOCIN, SBOCUP	

*SPEED *SVAL THKCMP *TIND (CRIBFMT)	 . Crib dataset (see section 5.10.4)

Table A.11 – Preset Laminating Device ("AGN") packet

Record label	See Table A.1 for record descriptions
*AX *ADD *CYL *FCOAT *LIND *LNAM *LSIZ *LTYP LTYPE3.03 *OPCB *OPCF *RXNM *SPH	 .

Table A.12 – Maintenance information ("MNT") packet

Record label	See Table A.1 for record descriptions
MID MODEL DEV OPERID SN VEN other records are up to device vendor	

Table A.13 – Process status ("INF") packet

Record label	See Table A.1 for record descriptions

*JOB	
*STATUS	Indicates mode of completion (0 = OK; 19 = Failed)

Table A.14 – Preset Lens Coater ("COA") packet

Record label	See Table A.1 for record descriptions
CPID	
CRIB	
DIA	
FRNT	
GBASEX	
GPRVM	
LMATTYPE	
LTYP	
$LTYPE_{3.03}$	
TIND	
TINT	

Table A.15 – Lens Measuring Device ("LMD") packet

Record label	See Table A.1 for record descriptions
(tracing data)	
ADD	
ADD2	
AX	
BACK	
BCOCIN, BCOCUP	
BCSGIN,BCSGUP	
CTHICK	
CYL	
DBL	
DIA	
FBFCIN, FBFCUP	
FBOCIN, FBOCUP	
FBSGIN, FBSGUP	
FRNT	

HBOX IPD LIND LTYP $LTYPE_{3.03}$ NPD OCHT PRVA PRVM SEGHT SGOCIN, SGOCUP SPH SWIDTH VBOX	

Table A.16 – Inspection Device ("INS") packet

Record label	See Table A.1 for record descriptions
INSADD INSAX INSCTHK INSCYL INSPRVA INSPRVM INSSGIN INSSGUP INSSPH TOLADD TOLASPEC TOLAX TOLCTHK TOLCYL TOLPRVA TOLPRVM TOLSGIN TOLSGUP	

TOLSHAPE TOLSPH	

Table A.17 – Lap Feeder Device (“LAP”) packet

Record label	See Table A.1 for record descriptions
LAPBAS LAPBASX LAPBIN LAPCRS LAPCRSX LAPM TIND	

Table A.18 – Drill Device (“DRL”) packet

Record label	See Table A.1 for record descriptions
(tracing data)	
BSIZ	
CIRC	
CSIZ	
*DBL	
*DIA	
DRILLE$_{3.04}$	Format negotiation via DRLFMT required.
EPRESS$_{3.02}$	
*ETYP	
*FBFCIN, *FBFCUP	
FBSGIN, FBSGUP	
FCRV	
*FTYP	
*IPD	
*LMATTYPE	
LMATID	
LTYPE$_{3.03}$	
*NPD	

OCHT	
SEGHT	
TNORM$_{3.02}$	
ZTILT	

Table A.19 – Lens Design System (“LDS”) file record set

Record label	See Table A.1 for record descriptions
(tracing data)	TRCFMT
(cribbing data)	CRIBFMT
(thickness data)	STHKFMT
LNAM	
SPH	
CYL	
AX	
ADD	
PRVM	
PRVA	
PIND	
MINCTR	
MINEDG	
ADJSPH	
ADJCYL	
ADJAX	
ADJPRVM	
ADJPRVA	
OPC	
LMATTYPE	
LMATID	
MBASE	
FRNT	
BACK	
DIA	
BCTHK	
BETHK	
LIND	

TIND CRIB ELLH FLATA FLATB PTOK PTPRVM PTPRVA BLKTYP BLKD RNGD RNGH HBOX VBOX FED FEDAX DBL MPD IPD FCSGIN, FCSGUP ENGMASK SEGHT FTYP ETYP PANTO BVD ZTILT FCRV	

Table A.20 – Host System ("LMS") file record set

Record label	See Table A.1 for record descriptions
(tracing data) (cribbing data) (surface thickness data)	TRCFMT CRIBFMT STHKFMT
CTHICK	

THNP	
THNA	
THKP	
THKA	
BLKTYP	
BLKD	
RNGD	
RNGH	
CRIB	
ELLH	
FLATA	
FLATB	
LDPATH	
OPC	
FRNT	
SAGBD	
SAGCD	
SAGRD	
LDDRSPH	
LDDRCYL	
LDDRAX	
LDNRSPH	
LDNRCYL	
LDNRAX	
LDSGSPH	
LDSGCYL	
LDSGAX	
LDPRVM	
LDPRVA	
BCERIN,BCERUP	
ERDRIN,ERDRUP	
ERNRIN,ERNRUP	
LAPBASX	
LAPCRSX	
GAX	
TIND	

Table A.21 – Preset Direct Surface Generator ("FSG") packet

Record label	See Table A.1 for record descriptions
(cribbing data)	CRIBFMT
*AVAL *BACK *BCTHK *BETHK *BLKB *BLKCMP *BLKD *BLKTYP CPRVA$_{3.02}$ CPRVM$_{3.02}$ *CRIB *DIA EECMP *ELLH FINCMP *FLATA *FLATB *FTTHK *GAX *GBASE *GBASEX *GCROS *GCROSX *GPRVA *GPRVM *GTHK *IFRNT *KPRVA *KPRVM *LAPBAS	

*LAPBASX *LAPCRS *LAPCRSX LAPM *LAPPRB LAPAX LDPATH SAGRD SAGBD SAGCD	

Table A.22 – Preset Direct Surface Polisher (“FSP”) packet

Record label	See Table A.1 for record descriptions
(cribbing data) (surface thickness data)	CRIBFMT STHKFMT
*AVAL *BACK *BLKB *BLKD *BLKTYP CPRVA$_{3.02}$ CPRVM$_{3.02}$ *CRIB *DIA *ELLH FINCMP *FLATA *FLATB *GAX *GBASE *GBASEX *GCROS *GCROSX *GPRVA	

*GPRVM *GTHK *IFRNT *KPRVA *KPRVM *LAPBAS *LAPBASX *LAPCRS *LAPCRSX LAPM *LAPPRB LAPAX LDPATH SAGRD	

Table A.23 – Preset Lens Information (“LNF”) packet

Record label	See Table A.1 for record descriptions
ABBE ADD ADD2 AR BACK BCERIN, BCERUP BCOCIN, BCOCUP BCSGIN, BCSGUP BCTHK BETHK COLR CTHICK DIA ERDRIN, ERDRUP ERNRIN, ERNRUP ERSGIN, ERSGUP LIND LMATNAME	

LMFR LNAM LSIZ LTYP LTYPE, LTYPEB, LTYPEF MBASE NREF SDEPTH SWIDTH	

A.4 Status codes

Status codes are numbers defined in Table A.24 which are transmitted in response packets to indicate the presence or absence of error conditions. This record has the form STATUS = <code>[;<description>]. The second field in the record is text, in which the error can be expressed literally. The record always appears in response packets. The presence of a non-zero status code in a response to a request will typically cause the current session to abort. The exception is during initialization, where the status code is used to indicate what form of initialization is supported. Status code modifiers, defined in Table A.25, may be added to the status codes to provide more detailed information about the error condition. The current set of modifiers is related to a specific Code, 17. When a host returns a description in the status code record, the device should display it if possible.

A.5 Enhanced Status codes

A new record type for extended Status data is provided in order to optionally allow more precise reporting of errors, including errors only on one lens, or different errors on each of the two lenses in an order, and to provide for warning messages. The previous specification, using only STATUS, continues to be normative, and is not deprecated.

When using XSTATUS to convey only warnings on an order that is manufacturable, a sender shall include STATUS=0. When using XSTATUS to indicate error(s) on an order that is not manufacturable, a sender shall include STATUS=3 together with one or more XSTATUS records, each containing error data specific to one or both lenses. Any number of warning XSTATUS records may appear in a packet, regardless of the manufacturability of an order.

When used in Job Status Notifications (see section 5.19), XSTATUS records may appear without a STATUS record.

The format is:

XSTATUS=RILIB; integer; text[;] EIWIN

The first field indicates the eye(s) to which the XSTATUS code applies. The second field contains the code. The third field contains a textual description of the code. The fourth field indicates whether the code indicates an error (“E”), a warning (“W”), or a notification. In the absence of a value in the fourth field, “W” is presumed.

New XSTATUS codes may be defined, which shall have values between 100 and 999. Codes greater than 1000 are non-standard, but may be used for vendor-specific purposes. The existing STATUS codes (especially zero) may appear in XSTATUS records where appropriate, though to preserve backward compatibility, where possible, errors should be expressed in STATUS records.

Example 1: Right lens has a problem, left is ok:

STATUS=3
XSTATUS=R;1001;Sphere out of range for xxxxx design;E<CR/LF>
XSTATUS=L;0<CR/LF>

(Note: in the above example, it would be equivalent to simply omit the XSTATUS record for the left eye.)

Example 2: Both lenses have problems:

STATUS=3
XSTATUS=R;1001;Sphere out of range for xxxx design;E<CR/LF>
XSTATUS=L;1002;Cylinder out of range for xxxx design;E<CR/LF>

NOTE: In the case of a successful process, a single “STATUS=0” record would be sufficient.

Table A.24 – Status codes

Status code	Description
0	No error
1	Job not found
2	Can’t store data; job is protected.
3	General, defined by host or device; the description field describes the error.
4	Cannot process job (data is available in host system, but is not appropriate for the device making the request).
5	Need initialization. This would normally be sent by a host when it has been re-started after an auto-format or preset initialization session has taken place
6	Invalid job ID.
7	Missing record. The record label for the missing record should appear in the description field.
8	Host error. Some internal error prevents the host from meeting the request. The error should be described in the description field.
9	Incompatible device/host
10	A value in a record is out of range. The offending record label should appear in the description field.
11	Receiver is busy, temporarily unable to fulfill the request.
12	Out of sync. When a host or device receives an out-of-sync packet, it should issue this status code and both sides should abort the session. The receiver of such a packet sends a confirmation to the sender.
13	Invalid initialization. Something is wrong with the initialization packet.

14	Auto format not supported. This is sent by a host which cannot perform auto format initialization but which can perform preset initialization. This lets the device know that it should proceed with preset format initialization.
15	Initialization not supported. This is sent when a host that cannot perform any type of initialization.
16	Invalid request. This is sent when a host cannot identify the type of request the device is making.
17	Unsupported tracing format. This is sent when a device attempts to use a tracing form that is not supported by the host. The status code modifiers (see Table A.22) may be added to this error code to help indicate the exact nature of the error.
18	Format error. This is sent when a device or host receives a packet that contains data that cannot be parsed in accordance with this Standard.
19	Process failed. This is used only in INF requests sent from devices to hosts to indicate that the process associated with the most recent request for the job indicated has been completed. This status indicates that the process did not finish successfully.
20	Invalid TOKEN. This is sent by LDS in an LMS file or packet when a TOKEN value supplied is not accepted for processing without additional billing, for a reason which may be further detailed in an XSTATUS record.
21	Subscription invalid, or click count exhausted. This is sent by an LDS in an LMS file or packet when either the subscription agreement for the lab has expired or, in a context in which “clicks” for lens designs are prepaid, the number of prepaid clicks has been exhausted.

Table A.25 – Modifiers for status code 17

Modifier	Description
256	None of the proposed tracing formats is acceptable
512	None of the proposed number of points is acceptable.
1024	None of the proposed radius modes is acceptable

NOTE The status code modifiers are evaluated as bits in the high byte of a 16-bit word, so they can be combined without information loss. The 255 possible values in the low byte can represent general error conditions, and the 8 bits in the high byte can be used to refine this data. The current set of modifiers relate only to status code 17. In the future, the values may have other meanings in conjunction with other codes.

Table A.26 – XSTATUS codes

XSTATUS code	Description
100	Job completed.
101	Job submitted under warranty.
102	Lab breakage.
103	Lab customer redo (e.g., Rx change).
104	Lab customer breakage.

105	Lab redo or recalculation.
106	Job cancelled.
107	Job shipped.
108	Job called to generator.
109	Job outsourced (therefore cancelled).
110	Job delayed.

NOTE XSTATUS codes in the range 100 – 110 are intended for notifications from a Host to a Lens Design System.

A.6 Process Control Records

Process control records are similar to data records. While data records contain static job data, process control records contain commands, which instruct devices to perform certain operations. The performance of a given operation may depend on not only job attributes, but upon the location of an operation in a series of operations, and the location of a job in such a series. They are therefore dynamic in nature; the field values of these records will depend on context.

Table A.27 – Process Control Records

Record label	Data type	Description
CKADD	0I1[;]	Check reading addition.
CKAX	0I1[;]	Check lens axis.
CKCOLR	0I1[;]	Check lens tint
CKCUT	0I1[;]	Check lens cutout.
CKDIA	0I1[;]	Check lens diameter.
CKHC	0I1[;]	Check hard coat
CKINK	0I1[;]	Check ink markings
CKPOLAR	0I1[;]	Check polarization
CKPRSM	0I1[;]	Check lens prism.
CKPRHI	0I1	Check horizontal prism imbalance.
CKPRVI	0I1	Check vertical prism imbalance.
CKPWR	0I1[;]	Check lens power.
CKREFL	0I1[;]	Check lens reflectivity
CKTHK	0I1[;]	Check lens thickness
CKXMIT	0I1[;]	Check lens transmission
CKXMITB	0I1	Check lens transmission balance
DOINK	0I1[;]	Perform ink marking
DOENG	0I1[;]	Perform engraving

<table>
<tr><td>DOPRINT</td><td>integer</td><td>Print ticket (0 = don’t print; n = print ticket using format n)</td></tr>
<tr><td>DOFBLK</td><td>0|1[;]</td><td>Perform finish blocking</td></tr>
<tr><td>DOSDF</td><td>N|R|L|B</td><td>Create SDF file (N)one, (R)ight, (L)eft, or (B)oth.</td></tr>
<tr><td>DOSDL</td><td>N|R|L|B</td><td>Create SDL (low-resolution surface matrix) file (N)one, (R)ight, (L)eft, or (B)oth.</td></tr>
</table>

Annex B
(Informative)

Packed binary format example

The following code is a complete implementation of packing and unpacking data to and from the format described in section 5.5.5. To pack data, call ***packit(src, dst, nradii)*** where "src" is a pointer to the array of integers to pack, "dst" is a pointer to a buffer into which the packed data will be written, and "nradii" is the number of elements in the dataset. To unpack data, call ***unpackit(dst, src, nradii)*** where "src" is a pointer to packed data, "dst" is a pointer to an array of integers, into which the unpacked data will be written, and "nradii" is the number of elements in the dataset.

The *pack()* and *unpack()* functions are used internally.

NOTE In the following code segments, integers are defined to be 16 bit quantities.

The following special requirements should be observed:

When packing one byte values and the nibble counter is not on an even byte boundary, the nibble order output is:

— HIGH NIBBLE of byte goes into the LOW NIBBLE of output byte N;

— LOW NIBBLE of byte goes into the HIGH NIBBLE of output byte N+1;

This makes the byte read correctly when looking at a byte dump.

When packing two byte values and the nibble counter is not on an even byte boundary, the nibble order output is:

— Original data: 0x1234

— 80x86 storage order is: 34 12. Nibble 3 is output first, then nibbles 4, 1, and 2.

If the nibble counter is on an even boundary, the same nibble order is used. This corresponds to the normal storage order of the 80x86 processor.

If at the end of the process, an odd number of nibbles results, a 0 nibble is added to the end of the data stream.

```
/* Global variables */
static unsigned char *outftp;
static int ftpn=0;

/*****************************************************************************
pack() function

INPUTS:
 i - The value to be packed into the output stream.
 n - The number of nibbles it should take up.
OUTPUTS:
 Deposits the packed data into the output buffer pointed to by outftp and updates
 outftp as needed.
GLOBALS:
 None
STATICS:
 outftp, ftpn

Packs nibbles into *outftp and increments the nibble counter ftpn by the number of nibbles
packed.

pack()is private to this module, called by packit()
```

```
*****************************************************************************/

void pack(i, n)
 unsigned int i;
 int n;

{
#ifdef NOT_80x86
 if (n == 4)
   swab(outftp, outftp, 2);              /* Put into 80x86 order */
#endif
if ( !(ftpn & 1) )                       /* on an even nibble boundary? */
{
 if ( n > 1 )
 {
   *outftp++ = (i & 0xff);
    if ( n > 2 )
      *outftp++ = (i & 0xff00)>>8;
 }
 else
   *outftp = i<<4;                       /* pack high nibble first */
}
else
{
 if ( n > 1 )
 {
   if ( n == 2 )
   {
      *outftp++ |= (i & 0xf0)>>4;        /* fill in low nibble with high nibble of next byte*/
      *outftp = (i&0x0f)<<4;             /* fill in high nibble with low nibble */
   }
   else
   {
      *outftp++ |= (i & 0x00f0)>>4;                       /* fill in nibble 3 */
      *outftp++ = (i & 0x000f)<<4 | (i & 0xf000)>>12;     /* nibbles 4 and 1  */
      *outftp = (i & 0x0f00)>>4;                          /* and nibble 2     */
    }
 }
 else
   *outftp++ |= (i & 0x0f);                               /* fill in low nibble */
}
ftpn+=n;
}

/****************************************************************************
packit() function

INPUTS:
 src      Pointer to integers to pack.
 dst      Buffer to contain packed data.
 num      Number of integers to pack.
OUTPUTS:
 Deposits packed data in output buffer (dst), and returns number of bytes actually packed.
GLOBALS:  None
STATICS:  outftp, ftpn

Packs the integer data into the output buffer.
Calls pack()
*****************************************************************************/

int  packit(src, dst, num)
 int *src;
 unsigned char *dst;
 int num;
```

```
{
 int    i;
 int    state=16;
 int    dr, d2r, dr1=0;

 outftp = dst;
 for (ftpn = 0, i = 0; i < num; i++)
 {
   if (!i)
      dr = src[i];
   else
      dr = src[i]-src[i-1];
   d2r = dr-dr1;
   switch(state)
   {
      case 16: /* Note this was originally, incorrectly, (dr<128 && dr>-128) */
         if (dr<128 && dr>-127)
         {
            state = 8;
            pack(0x8000, 4);
            pack(dr, 2);
         }
         else
         {
            pack(src[i], 4);
         }
         break;
      case 8:
         if (dr>=128 || dr<=-127)
         {
            state = 16;
            pack(0x81, 2);
            pack(src[i], 4);
         }
         else
            if (d2r<8 && d2r>-8)
            {
               state = 4;
               pack(0x80, 2);
               pack(d2r, 1);
            }
            else
            {
               pack(dr, 2);
            }
            break;
      case 4:
         if (d2r>=8 || d2r<=-8)
         {
               pack(0x8, 1);
               if (dr>=128 || dr<=-127)
               {
                      state = 16;
                      pack(0x81, 2);
                      pack(src[i], 4);
               }
               else
               {
                      state = 8;
                      pack(dr, 2);
               }
            }
            else
            {
```

```
              pack(d2r, 1);
            }
            break;
        default:
          return(-99);
          break;
    }
    dr1 = dr;
  }
  return( (ftpn+1)>>1 );
}


/***************************************************************************
unpack() function

INPUTS:
  i - The number of nibbles to unpack from the packed data stream.
OUTPUTS:
  The 16-bit integer value of the packed data
GLOBALS:
  None
STATICS:
  outftp

Returns the next nibble, byte, or word of packed data based on the size which is passed.  Since
byte order for words is different on Z8002, it swaps the order of the bytes before returning a
word value.

unpack() is local to this module, called by unpackit()
***************************************************************************/

int   unpack(i)
  int i;
{
  unsigned int       j;

  if ( !(ftpn & 1) )                    /* on an even boundary */
  {
    switch (i)
    {
        case 1:
          j = (*outftp & 0xf0)>>4;      /* pull from high nibble first */
          if ( j > 7 )
            j |= 0xfff0;
          break;
        case 2:
          j = *outftp++;                /* pull a whole byte */
          if ( j > 127 )
            j |= 0xff00;
          break;
        case 4:
          j = *outftp++;                /* pull first low byte */
          j |= ((int)(*outftp++))<<8;   /* or in high byte */
          break;
    }
  }
  else                                  /* starting in the middle of the byte */
  {
    j = ((*outftp++) & 0x0f);           /* pull from low nibble first */
    switch (i)
    {
        case 1:
          if ( j > 7 )
```

```
          j |= 0xfff0;                  /* extend sign */
        break;
      case 2:
        j <<= 4;                        /* shift up a nibble */
        j |= (*outftp & 0xf0)>>4;       /* and grab low nibble from high nibble */
        if ( j > 127 )
          j |= 0xff00;                        /* extend sign */
        break;
      case 4:
        j <<= 4;                        /* shift what we have up to be nibble 3 */
        j |= (*outftp & 0xf0)>>4;       /* add in nibble 4 to get low byte complete */
        j |= ((int)(*outftp++ & 0x0f))<<12;   /* add in nibble 1 */
        j |= (*outftp & 0xf0)<<4;             /* add in nibble 2 */
        break;
    }
  }
  ftpn += i;
#ifdef NOT_80x86
  if ( i == 4 )
    swab( (char *) j, (char *) j, 2);         /* put into Z8002 order */
#endif
  return( (int) j );
}

/****************************************************************************
unpackit() function

INPUTS:
  src    Pointer to an input buffer containing packed data.
  dst    Pointer to an output buffer to fill.
  n      Number of integers to unpack
OUTPUTS:
  Unpacked data in the output buffer (dst). Returns number of unpacked bytes.
GLOBALS:
  None
STATICS:
  outftp

Unpacks the packed data from src into dst. Calls unpackit().

****************************************************************************/

int  unpackit(dst, src, n)
  int *dst;
  unsigned char *src;
  int n;
{
  int state=16, size=4;
  int i, dr=0, d2r=0, dr1=0, x;

  outftp = src;
 ftpn = 0;
 for (i = 0; i < n; i++)
 {
   AGAIN:
   x=unpack(size);
   switch(state)
   {
      case 16:
        if ( x == 0x8000 )
        {
          state = 8;
          size = 2;
          goto AGAIN;
```

```
        }
        dst[i]=x;
        if (i)
           dr1 = x-dst[i-1];
        else
           dr1 = x;
        break;
     case 8:
        if ( (x & 0xff) == 0x80 )
        {
           state = 4;
           size = 1;
           goto AGAIN;
        }
        if ( (x & 0xff) == 0x81 )
        {
           state = 16;
           size = 4;
           goto AGAIN;
        }
        dr = x;
        if (i)
           dst[i] = dst[i-1]+dr;
        else
           dst[i] = dr;
        dr1 = dr;
        break;
     case 4:
        if ( (x & 0x0f) == 0x8 )
        {
           state = 8;
           size = 2;
           goto AGAIN;
        }
        d2r = x;
        dr = dr1+d2r;
        dst[i] = dst[i-1]+dr;
        dr1 = dr;
        break;
  }
 }
 return( (ftpn+1)>>1 );
}
```

Annex C
(Informative)

CRC calculation example

The following is a C subroutine in which the algorithm for the CRC calculation is demonstrated.

```
/*********************************************************************

        CRC16 - 16 bit CRC
        CCITT CRC-16 Cyclical Redundancy Check
        polynomial: X^16 + X^12 + X^5 + 1
        (used in XMODEM-CRC communications protocol)

*********************************************************************/

union _crc {
 unsigned char b[2];              /* high byte is b[1], low byte is b[0] */
 unsigned w;                      /* word value */
 };

unsigned crc16(len, start_crc, p)
int len;                          /* length of p */
unsigned start_crc;               /* starting value, initialize to zero */
unsigned char *p;                 /* pointer to memory of which to calculate crc */
{
 union _crc crc;
 int i;
 crc.w = start_crc;               /* set up starting value of CRC */

 while (len-- > 0)
 {
  crc.b[1] ^= *p++;               /* xor value of next byte into HIGH byte of CRC */
                                  /* this is for an 80x86 processor */
  for(i=0;i<8;++i)
    if (crc.w & 0x8000)           /* high bit set?? */
    {
      crc.w <<= 1;                /* left shift one */
      crc.w  ^= 0x01021;          /* XOR value 0x1021 */
    }
    else
    {
      crc.w <<= 1;                /* left shift one */
    }
 }
 return (crc.w);
}
```

The following code fragment will print “0cd3” (hexadecimal) as the CRC value to “Hello World!”:

```
 char hello[] = “Hello World!”;
 unsigned crc;
 crc = crc16(strlen(hello), 0, hello);
 printf(“%04x\n”,crc);
```