

1 INTERNET-DRAFT

Tom Hastings
Xerox Corporation
Bob Herriot
Sun Microsystems, Inc.
Norm Jacobs
Sun Microsystems, Inc.
Jay Martin
Underscore, Inc.
July~~June~~ 13, 1997

11 **Mapping between ~~of~~ LPR/LPD and ~~to~~ IPP Protocols**

12 **<draft-ietf-ipp-lpr-lpd-ipp-map-??.txt>**

13 **Expires Jan 1325, 1998**

14
15 **Status of this Memo**

16 This document is an Internet-Draft. Internet-Drafts are working documents of the
17 Internet Engineering Task Force (IETF), its areas, and its working groups. Note
18 that other groups may also distribute working documents as Internet-Drafts.

19 Internet-Drafts are draft documents valid for a maximum of six months and may be
20 updated, replaced, or obsoleted by other documents at any time. It is
21 inappropriate to use Internet-Drafts as reference material or to cite them other than
22 as "work in progress."

23 To learn the current status of any Internet-Draft, please check the "Iid-
24 abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on
25 ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim),
26 ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

27 **Abstract**

28 This Internet-Draft specifies the mapping between ~~of~~ (1) the commands and
29 operands of the "Line Printer Daemon (LPD) Protocol" specified in RFC 1179 and
30 (2)~~to~~ the operations and parameters of the Internet Printing Protocol (IPP). One
31 of the purposes of this document is to compare the functionality of the two
32 protocols. Another purpose is to facilitate implementation of gateways between
33 LPD and~~to~~ IPP gateway.

34 WARNING: RFC 1179 was not on standards track. While RFC 1179 was
35 intended to record existing practice, in some areas it fell short. However, this
36 specification maps between (1) the actual current practice of RFC 1179 and (2)
37 IPP. This document does not attempt to map the numerous divergent extensions
38 to the LPD protocol that have been made by many implementors.

39

40

TABLE OF CONTENTS

41 **1. INTRODUCTION.....3**

42 **2. MAPPING OF LPD COMMANDS TO IPP OPERATIONS3**

43 2.1 01 - Print any waiting jobs (Printer queue name).....4

44 2.2 02 - Receive a printer job (Printer queue name).....4

45 2.3 03 - Send queue state (short) (Printer queue name [, User Name, ...][, job numbers, ...])6

46 2.4 04 - Send queue state (long) (Printer queue name [, User Name, ...][, job numbers, ...]).....7

47 2.5 05 - Remove jobs (Printer queue name[, User name, ...][, job number, ...]).....8

48 **3. MAPPING OF LPD SUB-COMMANDS TO IPP OPERATIONS9**

49 3.1 01 - Abort job ()9

50 3.2 02 - Receive control file (Number of bytes in control file, Name of control file).....9

51 3.3 03 - Receive data file (Number of bytes in data file, Name of data file).....9

52 **4. MAPPING OF LPD CONTROL FILE LINES TO IPP OPERATION INPUT**
53 **PARAMETERS10**

54 **5. APPENDIX - RELATIONSHIP OF RFC 1179 TO EXISTING PRACTICE4**

55 **6. BIBLIOGRAPHY12**

56

57 Mapping between the LPD and IPP Protocols Mapping

58 **1. Introduction**

59 The reader of this specification is expected to be familiar with the IPP Model and
60 Semantics specification [1], ~~and the IPP Protocol specification [2], and~~ ~~Less familiarity~~
61 ~~with the Line Printer Daemon (LPD) protocol specification [3] is assumed as~~
62 ~~described~~ specified in RFC 1179.

63 RFC 1179 was written in 1990 in an attempt to document existing LPD protocol
64 implementations ~~practice using printers that were mainly fixed pitch character cell printers.~~
65 Since then, a number of undocumented extensions have been made by vendors to support
66 functionality specific to their printing solutions. All of these extensions consist of
67 additional control file directives. This document does not address any of these vendor
68 extensions. Rather it addresses existing practice within the context of the features
69 described by RFC 1179. Deviations of existing practice from RFC 1179 are so indicated.

70 In the area of document formats, also known as page description languages (PDL), RFC
71 1179 defines a fixed set with no capability for extension. Consequently, some new PDL's
72 are not supported, and some of those that are supported are sufficiently unimportant now
73 that they have not been registered for use with the Printer MIB[4] and IPP[1] [2], though
74 they could be registered if desired. See the Printer MIB specification [4] and/or the IPP
75 Model specification [1] for instructions for registration of document-formats with IANA.
76 IANA lists the registered document-formats as "printer languages".

77 ~~Since then a number of page description languages have emerged. Consequently, many~~
78 ~~of the commands that relate to document format specify document formats which have not~~
79 ~~been registered for use with the Printer MIB [4] and the Internet Printing Protocol (IPP)~~
80 ~~[1], [2]. Other LPD commands are intended to work on "text" only formats and so are~~
81 ~~inappropriate for many contemporary document formats that completely specify each~~
82 ~~page.~~

83 This document addresses the protocol mapping for both directions: mapping of the LPD
84 protocol to the IPP protocol and mapping of the IPP protocol to the LPD protocol.

85 **2. Terminology**

86 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
87 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in
88 this document are to be interpreted as described in RFC 2119 [6].

89 The syntax operands of the LPD commands is given appear in braces parens in the sub-
90 headings using ABNF [6?]. Optional input parameters are indicated inside square brackets
91 ([]). Repeated input parameters are indicated with an ellipsis ("...").

92 The following tokens are used in order to make the syntax more readable:

93 LF stands for %x0A (linefeed)

94 SP stands for %x20. (space)

95 **3. Mapping between LPD Commands and IPP Operations**

96 This section describes the mapping between from LPD on-the wire commands and IPP
97 operations. Each of the following sub-sections appear as sub-sections of section 5 of RFC
98 1179.

99 **3.1 Print any waiting jobs (Printer queue name) \1printer\n**

100 Command syntax: %x01 Printer-queue-name LF

101 In LPD, this comment starts the daemon, if it isn't already running. Such an equivalent
102 operation is not provided in IPP, since the IPP Printer is assumed to always be running,
103 where as in LPD, the client makes sure that the daemon is running using this command.

104 If an LPD-to-IPP mapper receives this LPD command, it SHALL ignore it and send no
105 IPP operation.

106 An IPP-to-LPD mapper SHALL send this LPD command after it has finished sending all
107 pending 'Receive a printer job' commands.

108 When recieved, this command is ignored. Under IPP, printers are always assumed to be
109 processing their queue, unless specifically disabled. When passing jobs from an IPP server
110 to an LPD server, this command should be generated after a stream of print request have
111 been transfered to the LPD based server. Under LPD, the assumption is that no process is
112 waiting to process a print queue unless there are jobs in the queue, and LPD has been told
113 to print any wating jobs.

114 **3.2 02--Receive a printer job (Printer queue name) \2printer\n**

115 Command syntax: %x02 Printer-queue-name LF

116 An LPD-to-IPP mapper SHALL map the 'Receive a printer job' command to use either:

- 117 • the Print-Job operation with a single data file or
- 118 • the Create-Job operation followed by a Send-Document operation for each
119 data file document.

120 If a job consists of a single data file document, the PrintJob operation is
121 RECOMMENDED.

122 If a job consists of more than one data file document, Create Job followed by Send-
123 Document for each data file document is REQUIRED. If the IPP Printer doesn't support
124 the Create-Job and Send-Document operations, the LPD-to-IPP mapper SHALL
125 submit each data file as a separate Print-Job operation (thereby converting a single LPD
126 job into multiple IPP jobs).~~reject the job and return an error.~~

127 ISSUE: Ok that I changed so that the mapper shall break a multi-document job into
128 separate jobs, one IPP job for each LPD data file, instead of error return?

129 NOTE: if Create-Job is used, it MUST~~must~~ precede the Send-Document operation even if
130 the LPD control file, which supplies attributes for Create-Job, arrives after all documents.

131 An IPP-to-LPD mapper SHALL map the following IPP operations to~~in~~ this LPD
132 command:

- 133 • Print-Job
- 134 • Print-uri
- 135 • Create-Job followed by Send-Document or Send-URI for each document

136 The mechanism for mapping between an LPD Printer-queue-name operand and the IPP
137 "printer-uri" parameter is not defined in this document.

138 ISSUE: error code conversion.

139 It is my belief that the described method is overly complex. From an implementation stand
140 point, the way I am suggesting below is more consistent, and will require less special case
141 code in an implementation. I would suggest the following:

142 This command is equivalent to an IPP Create Job and a Send Document operation for
143 each job data file within the request. Success or failure return codes as described in RFC-
144 1179 shall be generated equivalent to the return codes generated from performing the IPP
145 Operations. The printer queue name operand is maps to the IPP Printer URL.

146 It's important to note that the IPP protocol requires that the client supply input parameters
147 before submitting the document data. It's also important to note that RFC 1179 does not
148 specify an ordering for passing job document data and job control data between client and
149 server. Most implementations of RFC 1179 pass print job document data files before
150 passing the job control data file. The end result of all of this is that an LPD to IPP
151 gateway is likely to have to receive the entire LPD job before performing any IPP
152 operations.

153 This command is equivalent to the IPP Print-Job operation when only one data file is
154 submitted and is equivalent to the IPP Create Job and Send-Document operations when

155 more than one file is submitted. Success or error codes are produced by the LPD
156 command that the client shall read and the IPP operations returns equivalent status code
157 results. The LPD queue name operand is equivalent to the IPP Printer URI.

158 Unlike LPD, an IPP protocol requires that the client supply input parameters before the
159 document data.

160 See the next section for the mapping of the LPD "second level commands" to IPP input-
161 parameters.

162 **3.2 03--Send queue state (short) (Printer queue name [, User Name, ...][, job**
163 **numbers, ...]) \3printer {user/job ...}\n**

164 Command syntax: %x03 Printer-queue-name *(SP (User-Name / job-number))

165 RGH: the syntax is long enough here that I think it doesn't belong in the header.

166 If the LPD command contains only the Printer-queue-name operand, the LPD-to-IPP
167 mapper SHALL use the Get-Attributes operation of the corresponding IPP Pprinter to get
168 printer-state information and either the Get-Jobs operation of the Pprinter or Get-
169 Attributes to each specified job to get information about all of the jobs. With Get-
170 Attributes, it SHALL get shall request use the "printer-state" and "printer-state-reasons"
171 attributes. With Get-Jobs, it SHALL request use the "number-of-intervening-jobs", "job-
172 originating-user", "job-name", "document-name" (or "document-uri"), and "job-k-octets".
173 (Need to check that this is correct).

174 NOTE: RFC 1179 does not specify what attributes are returned in response to a 'Send
175 queue state' (short) command, but leaves it up to implementation. The IPP attributes
176 specified in this specification reflect existing practice.

177 NOTE: This specification does not specify how the LPD-to-IPP mapper maps: (1) the
178 LPD Printer-queue-name operand to the IPP "printer-uri" parameter or (2) the LPD job-
179 number operand to the IPP "job-uri" parameter, since the format of these URIs is opaque
180 in the IPP protocol and is implementation-dependent.

181 It SHALL format that information as follows:

182 (See PSIS for exact spacing):

183 Printer state/printer state reason (or equivalent)

184 Rank Owner Job Files Total Size

185 1st owner 21 motd, vfstab 355 bytes

186 If the LPD command contains specifies one or more User-name operands or job numbers,
187 the LPD-to-IPP mapper SHALL get all the jobs as above using the Get-Jobs operation on

188 the Printer and then do its own filtering on the returned value of the "job-originating-user"
189 attribute for each job.

190 If the LPD command contains only job-number operands, the LPD-to-IPP mapper
191 SHALL either (1) get all the jobs as above using the Get-Jobs operation on the Printer and
192 then do its own filtering or (2) get each specified job individually using separate Get-
193 Attributes operations (multiple jobs may be requested in a single IPP connection with
194 multiple Get-Attribute operations, one for each job).

195 The IPP-to-LPD mapper shall use the long version of this command. See that command.

196 This command with only the Printer queue name operand is equivalent to the IPP Get-
197 Jobs operation when the client supplies a (short) list of requested attribute names.

198 This command with the Printer queue name operand and one job number is equivalent to
199 the IPP Get Attributes operation when the client supplies a job URI and a (short) list of
200 requested attribute names. Multiple jobs may be requested in IPP in a single connection
201 with multiple Get-Attribute operations.

202 There is no way in IPP to request jobs by user name. The IPP WG removed the "user-
203 name" input parameter during development of IPP. The IPP client will have to filter out
204 jobs specified users.

205 Unless the job URI is easily derived from the supplied information, I would suggest that
206 the same filtering mechanism be used to select print jobs when one a job is specified.
207 Also, I would add the following:

208

209 The response must be returned in the following format: (from a BSD 4.3 LPD server)

210 printer-state-reason (or equivalent)

211 Rank Owner Job Files Total Size

212 1st owner 21 motd, vfstab 355 bytes

213 ...

214

215 **ISSUE: do we want to add make the user-name attribute we deleted last**
216 **meeting?**

217 **3.2 04--Send queue state (long) (Printer queue name [, User Name, ...][, job**
218 **numbers, ...])**

219 Command syntax: %x04 printer-name *(SP (user-name / job-number))

- 220 Same mapping as the 'Send queue state' (short) command. The IPP client supplies a
221 longer list of requested attributes to the Get-Jobs or Get-Attributes operations.
- 222 The LPD-to-IPP mapper should specify additional attributes than the ones listed for the
223 'Send queue state' (short) command. Again the response must be in a format consistent
224 with the output of lpq -l from a BSD 4.3 LPD server.
- 225 NOTE: RFC 1179 does not specify what attributes are returned in response to a 'Send
226 queue state' (short) command, but leaves it up to implementation.
- 227 The IPP-to-LPD mapper shall use this command to get what attributes it can from the
228 LPD server. We should list what this set is. I think the PSIS may help.
- 229 **3.2 05 - Remove jobs (Printer-queue-name[, User-name, ...][, job-number, ...])**
230 **\5printer-user {job/user ...}**
- 231 Command syntax: %x05 Printer-queue-name SP agent *(SP (User-name / job-number))
- 232 The agent operand is the user-name of the user initiating the 'Remove jobs' command.
233 The special user-name 'root' indicates a privileged user.
- 234 The LPD-to-IPP ~~to-LPD~~ mapper shall map this command to ~~use the~~ Cancel-Job
235 operation ~~to cancel a job~~.
- 236 This command with the Printer-queue-name operand and one job-number operand is the
237 same as the IPP Cancel-Job operation when the client supplies just the job URI. Multiple
238 jobs may be canceled in IPP in a single connection with multiple Cancel-Job operations.
239 In IPP only a privileged operator may cancel jobs belonging to another user.
- 240 NOTE: This specification does not specify how the LPD-to-IPP mapper maps: (1) the
241 LPD Printer-queue-name to the IPP "printer-uri" or (2) the LPD job-number to the IPP
242 "job-uri", since the format of these URIs is opaque in the IPP protocol and is
243 implementation-dependent.
- 244 There is no IPP equivalent for the LPD 'Remove jobs' command with just the Printer-
245 queue-name operand supplied, since IPP provides no way to cancel the current job.
- 246 There is no IPP equivalent for the LPD 'Remove jobs' command with a User-name
247 operand supplied, since IPP provides no way to cancel a job that requires root privileges
248 to cancel jobs specified by user name.
- 249 The LPD-to-IPP mapper shall map ~~use this command for a~~ Cancel-Job operation to this
250 command.
- 251 There are some major issues about setting the agent.

252 **4. Mapping between LPD Sub-Commands and IPP Operations**

253 This section describes the mapping between ~~from~~ LPD sub-commands and IPP
254 operations. Each of the following sub-sections appear as sub-sections of section 6 of RFC
255 1179. The operands of the sub-commands appear in parens in the sub-headings

256 **4.1 01 - Abort job ()\1\1**

257 Sub-command syntax: %x01

258 This sub-command is intended to abort any job transfer in process. If an IPP Create-Job
259 operation and/or a Send-Document operation were performed on behalf of the receive job
260 command that is being aborted, an equivalent to the IPP Cancel-Job operation should be
261 issued for the when the client supplies the job URI that was returned by the Printer on
262 which the Create-Job operation was performed. Also, any temporary files created while
263 processing the 'Receive job request' should be cleaned up, and the connection to the client
264 should be closed. Finally, this sub-command is implied if at any time the connection
265 between the LPD RFC-1179-client and server is terminated before an entire print job has
266 been transferred via an LPDRFC-1179 'Receive job request'. or when the client
267 closes the connection.

268 ISSUE: is IPP defined at this point to abort a job whose connection is closed before the
269 job has been fully received. If so, that is an alternate and simpler way to abort the job.

270 **4.2 02 - Receive control file (~~Number of bytes in control file, Name of control file~~)**
271 **\2octet-count file-name\n**

272 Sub-command syntax: %x02 Number-of-bytes-in-control-file, Name-of-control-file

273 This sub-command is roughly equivalent to the IPP Create-Job Send-Document operation.
274 Once the control file has been received, it's contents should be translated, and an
275 appropriate IPP Create-Job operation performed. when the client supplies the job URI
276 returned by the IPP Create-Job operation.

277 However, some information, such as Document-Name go in the Send-Document
278 operation.

279 **4.3 03 - Receive data file (~~Number of bytes in data file, Name of data file~~)\3octet-**
280 **count file-name\n**

281 Sub-command syntax: %x03 Number-of-bytes-in-data-file Name-of-data-file

282 This sub-command is roughly equivalent to the IPP Send-Document operation. If the
283 control file has been previously received, and it's corresponding IPP Create-Job operation

284 performed, an IPP Send-Document operation can be performed using ~~when the client~~
285 ~~supplies~~ the job URI returned by the IPP Create-Job operation.

286 ~~When performing the Send-Document operation is performed,~~ the size of the document
287 ~~data MUST be specified. Unfortunately RFC-1179 alludes to a method for passing an~~
288 ~~arbitrary length data file. This is described as being done by using an octet-count of zero,~~
289 ~~however the description isn't complete, and in practice, no implementations allow sending~~
290 ~~or receiving arbitrary length data files. As in LPD, the client may specify the entire size of~~
291 ~~the document data or may use chunking in order to push an arbitrary and unknown~~
292 ~~amount of data to the Printer.~~

293 **5. Mapping of LPD control file lines to IPP Operation Input Parameters**

294 This section describes the mapping from LPD control file lines to IPP operation input
295 parameters for the Print-Job, Create-Job, and Send-Document operations. Each of the
296 following sub-sections appear as sub-sections of section 7 of RFC 1179. ~~The operands of~~
297 ~~the control file lines appear in parens in the sub-headings.~~

298 ISSUE: somewhere, we need to map the LPD query format to IPP attributes.

299 In LPD text operands have a maximum length of 31 or 99 while IPP input parameters
300 have a maximum of 255 characters. Therefore, no data is lost when mapping/
301 converting from LPD to IPP. However, when mapping from IPP to LPD, there may be some data
302 loss if the IPP parameters exceed the maximum length of the LPD equivalent operands.

303 In the following table, IPP input parameter names are indicated in double quotes (") and
304 input parameter values are indicated in single quotes ('). Values of the IPP "document-
305 format" attribute that could be registered, but are not currently, are indicated with "***".

306 ~~We need the mapping both directions. Where there is a one-to-one mapping, both~~
307 ~~directions are specified. Where IPP has none, the I think that in LPD-to-IPP the attribute~~
308 ~~is ignored, and in the IPP-to-LPD the LPD feature is left unspecified.~~

LPD command	Equivalent IPP input parameter(s)
C Class for banner page	None. LPD default = "job-originating-host" why is this here. If there is no mapping, then leave it out. LPD doesn't actually default to filling in a Chostname in the control file, but instead puts the hostname on the burst page if no class is specified.
H <u>Originating Host</u>	<u>"job-originating-host"</u>
I Indent Printing	None. IPP WG deleted "left margin"
J Job name for banner page	"job-name"

L	Print banner page	"job-sheets" = any but 'none' <u>Absence of an 'L' directive indicates that "job-sheets=none" is set.</u>
M	Mail When Printed	"notification-events" = 'job-completion' and "notification-method" = 'mailto:// <u>Job-originating-user@job-originating-host</u> User Name'
N	Name of source file	"document-name" <u>This is on a per data file basis</u>
P	User identification	"job-originating-user"
S	Symbolic link data	None. "document-uri" in combination with Send-URI operation
T	Title for pr	None. IPP WG deleted "heading" input parameter.
U	Unlink data file	None.
W	Width of output	None. IPP WG deleted "width" input parameter
1	troff R font	None.
2	troff I font	None.
3	troff B font	None.
4	troff S font	None.
c	Plot CIF file	"document-format" = 'CIF' **
d	Print DVI file	"document-format" = 'TeX DVI' **
f	Print formatted file	"document-format" = 'Automatic <u>autosense</u> SimpleText' <u>In practice, this value is often overloaded. It is often used with any format of document data including PostScript and PCL data.</u>
g	Plot file	"document-format" = 'BSDPlotLibrary' **
k	reserved for Kerberized clients and servers	None. <u>This is unimplemented in LPD implementations. It was a place holder for future work that never occurred.</u>
l	Print file leaving control characters	"document-format" = 'Automatic <u>autosense</u> ' none <u>In practice, this is often used as a rough equivalent to the 'f' directive. Again it may mean one of many document formats.</u>

n	Print ditroff output file	"document-format" = 'ditroff' **
o	Print Postscript output file	"document-format" = 'ps' "document-format" = 'PSapplication/postscript'(7) <u>o is recognized by LPD-to-IPP, but never generated in IPP-to-LPD. Rather 'f' is used.</u> <u>This was not implemented in any RFC-1179 implementations until very recently in WinNT.</u>
p	Print file with 'pr' format	None. The IPP WG deleted "headers" = 'title' and "paginate". <u>It therefore is equivalent to 'f' or 'l'</u>
r	File to print with FORTRAN carriage control	"document-format" = 'FORTRAN' **
t	Print troff output file	"document-format" = 'troff' **
v	Print raster file	"document-format" = 'RasterFormat' **
z	reserved for future use with the Palladium print system	None. <u>This was reserved for the MIT Palladium print system, but was never used by that system is unimplemented.</u>

309 **~~6. Appendix - Relationship of RFC 1179 to Existing Practice~~**

310 ~~RFC 1179 was an attempt to document existing practice. However, many~~
311 ~~implementations implement the "f" and "p" commands as 'auto-sense', that is the server~~
312 ~~attempts to determine the document format by examining the document data, rather than~~
313 ~~assuming that the document is simple ASCII. Also at the time of writing, no~~
314 ~~implementations implemented the "o" to indicate the emerging PostScript document~~
315 ~~format. Since then an implementation has supported the "o" command to indicate~~
316 ~~PostScript.~~

317 **6. Bibliography**

318 [1] R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing
319 Protocol/1.0: Model and Semantics", <draft-ietf-ipp-model-02.txt>, July 1997.

320 [2] R. Herriot, S. Butler, P. Moore, R. Turner, "Internet Printing Protocol/1.0: Protocol
321 specification", <draft-ietf-ipp-protocol-00.txt>, July 1997.

322 [3] L. McLaughlin, RFC 1179, "Line Printer Daemon Protocol", RFC 1179, August 1990.

- 323 [4] Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC
324 1759, March 1995~~Printer MIB.~~
- 325 [5] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119,
326 March 1997
- 327 [6] D. Crocker et al., "Augmented BNF for Syntax Specifications: ABNF", draft-ietf-
328 drums-abnf-02.txt.

329 **7. Author's Addresses**

330 Thomas N. Hastings
331 Xerox Corporation
332 701 S. Aviation Blvd., ESAE-231
333 El Segundo, CA 90245
334
335 Phone: 310-333-6413
336 Fax: 310-333-5514
337 Email: hastings@cp10.es.xerox.com
338

339 Robert Herriot
340 Sun Microsystems Inc.
341 2550 Garcia Ave., MPK-17
342 Mountain View, CA 94043
343
344 Phone: 415-786-8995
345 Fax: 415-786-7077
346 Email: robert.herriot@eng.sun.com
347

348 Norm Jacobs
349 Sun Microsystems Inc.
350 1430 Owl Ridge Rd.
351 Colorado Springs, CO 80919
352
353 Phone: (719) 532-9927
354 Fax: (719) 535-0956
355 Email: Norm.Jacobs@Central.sun.com
356

357 Jay Martin
358 Underscore Inc.
359 ????????????????
360 Nashua, NH ?????
361

362 Phone: (603) 889-7000
363 Fax: (603) ???-????
364 Email: jkm@underscore.com
365