

1 INTERNET-DRAFT **There are 5 ISSUES highlighted like this.**
2 <draft-ietf-ipp-notify-send-00.txt>

3
4 Hugo Parra
5 Novell, Inc.
6 Tom Hastings
7 Xerox Corp.
8 February 2, 2000

9 Internet Printing Protocol/1.1: **The 'ipp-notify-send' Notification Delivery Method and Protocol**

10 Copyright (C) The Internet Society (1999). All Rights Reserved.

11 **ISSUE 01 - What should the name of this delivery method and protocol be that we use in the title of this document?**

12 **ISSUE 02 - What should the scheme name be? Consider 'ipp-notify-send' a working title, until we see several schemes. The 'ipp-notify-poll' delivery method is another example. The IETF likes words or well-recognized acronyms, not abbreviations in scheme names, so lets include "notify"?**

13 **ISSUE 03 - Should the scheme name be used in the title?**

14 Status of this Memo

15 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of [rfc2026]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

17 The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

18 The list of Internet-Draft Shadow Directories can be accessed as <http://www.ietf.org/shadow.html>.

19 **Abstract**

20 The IPP event notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1. [ipp-ntfy] requires the definition of one or more delivery methods for dispatching event notification reports to Notification Recipients. This document describes the semantics and syntax of the 'ipp-notify-send' event notification delivery method that is itself a request/response protocol. For this delivery method, an IPP Printer sends (pushes) IPP event Notifications to the Notification Recipients using the protocol defined herein which includes HTTP as a transport.

32 The full set of IPP documents includes:

33 Design Goals for an Internet Printing Protocol [RFC2567]

34 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]

35 Internet Printing Protocol/1.1: Model and Semantics (this document)

36 Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]

37 Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]

38 Mapping between LPD and IPP Protocols [RFC2569]

39

40 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing
41 functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included
42 in a printing protocol for the Internet. It identifies requirements for three types of users: end users,
43 operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A
44 few OPTIONAL operator operations have been added to IPP/1.1.

45 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document
46 describes IPP from a high level view, defines a roadmap for the various documents that form the suite of
47 IPP specification documents, and gives background and rationale for the IETF working group's major
48 decisions.

49 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the abstract
50 operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It defines the
51 encoding rules for a new Internet MIME media type called "application/ipp". This document also defines
52 the rules for transporting a message body over HTTP whose Content-Type is "application/ipp". This
53 document defines a new scheme named 'ipp' for identifying IPP printers and jobs.

54 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
55 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the
56 considerations that may assist them in the design of their client and/or IPP object implementations. For
57 example, a typical order of processing requests is given, including error checking. Motivation for some of
58 the specification decisions is also included.

59 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of gateways
60 between IPP and LPD (Line Printer Daemon) implementations.

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85

Table of Contents

- 1 Introduction4
- 2 Terminology4
- 3 Model and Operation4
- 4 Notification Operations5
 - 4.1 SEND-NOTIFICATIONS OPERATION5
 - 4.1.1 *Send-Notifications Request*6
 - 4.1.2 *Send-Notifications Response*.....6
 - 4.2 NOTIFICATION PROTOCOL URI SCHEME7
- 5 Encoding of the Operation Layer.....7
 - 5.1 NEW ATTRIBUTE TAG.....7
 - 5.2 NEW STATUS CODES7
 - 5.2.1 *unknown-notification-recipient (0xYYYY)*.....7
 - 5.2.2 *unable-to-delivery-notification-report (0xZZZZ)*.....8
 - 5.2.3 *successful-ok-but-cancel-subscription (0XXXXX)*.....8
- 6 Encoding of Transport Layer9
- 7 IANA Considerations10
- 8 Internationalization Considerations10
- 9 Security Considerations10
 - 9.1 SECURITY CONFORMANCE11
- 10 References11
- 11 Author's Addresses12
- 12 Full Copyright Statement.....12

86

87 **1 Introduction**

88 IPP Printers that support the OPTIONAL IPP event notification extension [ipp-ntfy] either a) accept, store,
 89 and use notification Subscriptions to generate notification reports and implement one or more delivery
 90 methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining
 91 tasks to a Notification Delivery Service. The 'ipp-notify-send' event notification delivery method specified
 92 in this document is itself a request/response protocol that may be used in a variety of notification scenarios.
 93 Its primary intended use is for IPP Printers to send (push) event notifications to Notification Recipients
 94 using the IPP Send-Notifications operation over HTTP. However, it may also be used by IPP Printers to
 95 send notification to Notification Services and by Notification Delivery Services to send notifications to the
 96 Ultimate Notification Recipients (see [ipp-ntfy]). Furthermore, this protocol can be extended in the future
 97 to add other operations, such as querying a Notification Recipient for its capabilities.

98 **2 Terminology**

99 This document uses terms such as "attributes", "keywords", and "support". These terms have special
 100 meaning and are defined in the model terminology [ipp-mod] section 12.2.

101 Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED
 102 NOT, and OPTIONAL, have special meaning relating to conformance. These terms are defined in [ipp-
 103 mod] section 12.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

104 This section defines the following additional terms that are used throughout this document:

- 105 **REQUIRED:** if an implementation supports the extensions described in this document, it **MUST**
 106 support a **REQUIRED** feature.
- 107 **OPTIONAL:** if an implementation supports the extensions described in this document, it **MAY** support
 108 an **OPTIONAL** feature.
- 109 Event Notification (Notification for short) - See [ip-ntfy]
- 110 Notification Source - See [ipp-ntfy]
- 111 Notification Recipient - See [ipp-ntfy]
- 112 Subscription object - See [ipp-ntfy]
- 113 Ultimate Notification Recipient - See [ipp-ntfy]

114 **3 Model and Operation**

115 In the IPP Notification Model [ipp-ntfy], one or more Per-Job Subscriptions can be supplied in the Job
 116 Creation operation or **OPTIONALLY** as subsequent Create-Job-Subscription operations; one Per-Printer
 117 Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription
 118 objects becomes the owner of the Subscription object.

119 When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The
 120 "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications

121 when subsequent events occur and the method for notification delivery that the IPP Printer is to use. For
122 the Notification delivery method defined in this document, the notification method is 'ipp-notify-send' and
123 the rest of the URI is the address of the Notification Recipient to which the IPP Printer will send the Send-
124 Notifications operations using HTTP as a transport.

125 The 'ipp-notify-send' event notification delivery method defined in this document is also a client/server
126 protocol. The "client" in this HTTP relationship is the Notification Source described in [ipp-ntfy] while the
127 "server" is the Notification Recipient. The Notification Source invokes the Send-Notifications operation
128 supported by the 'ipp-notify-send' notification protocol to communicate IPP event Notification contents to
129 the Notification Recipient. The Notification Recipient only conveys information to the Notification Source
130 in the form of responses to the operations initiated by the Notification Source.

131 All requests defined for this protocol will be issued as HTTP POST operations and their corresponding
132 HTTP notification responses will be returned in the responses to those HTTP POST operations. Hence,
133 Notification Sources that implement the 'ipp-notify-send' delivery method and protocol will need to include
134 an HTTP client stack while notification recipients that implement this protocol will need to support an
135 HTTP server stack (see section 6 for more details).

136 **4 Notification Operations**

137 The Notification Source composes the information defined for an IPP Notification [ipp-ntfy] and sends it
138 using the Sent-Notifications operation to the Notification Recipient supplied in the Subscription object.

139 The 'ipp-notify-send' delivery method and notification protocol makes extensive use of the operations
140 model defined by IPP [rfc2566]. This includes, the use of a URI as the identifier for the target of each
141 operation, the inclusion of a version number, operation-id, and request-id in each request, and the definition
142 of attribute groups. The Send-Notifications operation uses the Operation Attributes group, but currently has
143 no need for the Unsupported Attributes, Printer Object Attributes, and Job-Object Attributes groups.
144 However, it uses a new attribute group, the Notification Attributes group.

145 In its 1.0 version, the 'ipp-notify-send' delivery method and notification protocol is composed of a single
146 operation, but may be extended in the future as needed (e.g., to find out specific capabilities of an 'ipp-
147 notify-send' Notification Recipient). The operation currently defined is Send-Notifications.

148 **4.1 Send-Notifications Operation**

149 This REQUIRED operation allows a Notification Source to send one or more Notifications to a Notification
150 Recipient using HTTP. The operation has been tailored to accommodate the current definition of IPP
151 Notification [ipp-ntfy].

152 Both Machine-Consumable and Human-Consumable notifications may be sent to a Notification Recipient
153 through this operation.

154 4.1.1 Send-Notifications Request

155 The following groups of attributes are part of the Send-Notifications Request:

156 Group 1: Operation Attributes

157 Natural Language and Character Set:

158 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
159 section 3.1.4.1.

160

161 Target:

162 The URI of the 'ipp-notify-send' Notification Recipient.

163 Group 2 to N: Notification Attributes

164 "human-readable-report" (text)

165 The 'ipp-notify-send' Notification Source **OPTIONALLY** supports this attribute. This attribute is a
166 text string generated by the IPP printer or Notification Delivery Service from the contents of the IPP
167 Notification suitable for human consumption. If the Notification Source supports this attribute, it
168 **MUST** supply this attribute if the Subscription object contains the "notify-text-format"
169 (mimeMediaType) attribute. The text value of this attribute **MUST** be localized in the charset
170 identified by the "notify-charset" (charset) attribute and the natural language identified by the notify-
171 natural-language" (naturalLanguage) attribute supplied in the associated Subscription object that
172 generates this event Notification. The format of the text value is specified by the value of the
173 "notify-text-format" (mimeMediaType) supplied in the associated Subscription object.

174

175 **ISSUE 04 - "human-readable-report" has been added to the [ipp-ntfy] Notification Model document,**
176 **so ok to change this description to be a reference to "human-readable-report" in [ipp-ntfy]?**

177 All of the **REQUIRED** attributes and any of the **OPTIONAL** attributes indicated in [ipp-ntfy] for a Push
178 event Notification, including "notify-text-format-type" (mimeMediaType), if the "human-readable-
179 report" (text) attribute is included, so that the Notification Recipient will know the text format of the
180 "human-readable-report" (text) attribute value.

181 These attributes communicate the same information as the notification attributes by the same name
182 described in sections 7.4, 7.5, and 7.6 of [ipp-ntfy]. The rules that govern when each individual attribute
183 **MUST** or **MAY** be included in this operation precisely mirror those specified in [ipp-ntfy].

184 4.1.2 Send-Notifications Response

185 The 'ipp-notify-send' Notification Recipient returns a status code for the entire operation and one for each
186 Notification Report in the request if the operation's status code is other than "success-ok". If the 'ipp-notify-
187 send' notification listener receives a Notification report that it can't pair up with a subscription it knows
188 about, it can return an error status-code to indicate that events associated with that subscription should no
189 longer be sent to it.

190 Group 1: Operation Attributes

191 Natural Language and Character Set:
192 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section
193 3.1.4.1.

194 Group 2 to N: Notification Attributes

195 "notification-report-status-code" (type2 enum)
196 Indicates whether the 'ipp-notify-send' Notification Recipient was able to consume the n-th
197 Notification Report.

198 4.2 Notification Protocol URI Scheme

199 The 'ipp-notify-send' event notification delivery method uses the 'ipp-notify-send' URI scheme in the
200 "notify-recipients" attribute in the Subscription object in order to indicate the event notification delivery
201 method defined in this document. The remainder of the URI indicates the host and address of the
202 Notification Recipient that is to receive the Send-Notification operation.

203 REPEAT OF ISSUE 02 - What should the scheme name be? Consider 'ipp-notify-send' a working title,
204 until we see several schemes.

205 5 Encoding of the Operation Layer

206 The 'ipp-notify-send' event notification delivery method and protocol uses the same operation layer
207 encoding model and syntax as IPP [ipp-pro] with two extensions:

208 5.1 New attribute tag

209 A new notification attributes tag is defined:

210 notification-attributes-tag = %x07 ; tag of 7

211 5.2 New status codes

212 ISSUE 05 - Should we move the status codes into the Notification Model document in order to have the
213 same status codes for any other delivery method that might be defined?

214 The following status codes are defined:

215 5.2.1 unknown-notification-recipient. (0xYYYY)

216 The Notification Recipient returns this status code in order to indicate that the intended Ultimate
217 Notification Recipient is not known to the Notification Recipient.

218 **5.2.2 unable-to-delivery-notification-report (0xZZZZ)**

219 The Notification Recipient returns this status code in ordre to indicate that it was unable to deliver the event
 220 Notification to the intended Ultimate Notification Recipient.

221 **5.2.3 successful-ok-but-cancel-subscription (0xXXXX)**

222 The Notification Recipient indicates that it no longer wants to receive Notifications for this Subscription
 223 object. Therefore, the Subscription object is canceled. Note: this status code allows the Notification
 224 Recipient to cancel a Subscription object without having to be the owner of the Subscription object. Only
 225 the owner of the Subscription object can cancel a Subscription object using the Cancel-Subscription
 226 operation.

227

228 The encoding for the Send-Notification Request consists of:

229	-----		
230	version-number	2 byte	
231	-----		
232	operation-id	2 bytes	
233	-----		
234	request-id	4 bytes	
235	-----		
236	operation-attributes-tag	1 byte	
237	-----		
238	attributes-charset	u bytes	
239	-----		
240	attributes-natural-language	v bytes	
241	-----		
242	target-attribute	w bytes	
243	-----		
244	notification-attributes-tag	1 byte	- 1 or more
245	-----		
246	notification-attr-list	x bytes	
247	-----		
248	end-of-attributes-tag	1 byte	
249	-----		

250 Where:

251 *version-number* is made up of a major-version-number of %d1 and a minor-version-number of %d0
 252 indicating the 1.0 version of the 'ipp-notify-send' event notification delivery method and protocol.

253 *operation-id*, in the 1.0 version of the protocol, can only be 0x00003, Send-Notification.

254 *request-id* is any 4 byte number provided by the notification source and must be matched by the notification
 255 recipient in the corresponding response to a request. It assists the notification source in associating
 256 operation responses with their corresponding requests. Note that this request id is independent of the
 257 request id embedded in the notification report, which is opaque to the delivery method but assists the
 258 notification recipient order and identity missing or duplicate notification reports.

259 *operation-attribute tag, natural-language-attribute, charset-attribute, target-attribute, and end-of-*
 260 *attributes-tag* have the same syntax and semantics as in [ipp-pro].

261 *notification-attr-list* contains a list of the attributes that make up a single notification (see section 2 above)
 262 encoded using the syntax specified in [ipp-pro].

263 The encoding for the Send-Notification Response consists of:

264	-----		
265		version-number	2 byte
266	-----		
267		status-code	2 bytes
268	-----		
269		request-id	4 bytes
270	-----		
271		operation-attributes-tag	1 byte
272	-----		
273		attributes-charset	u bytes
274	-----		
275		attributes-natural-language	v bytes
276	-----		
277		target-attribute	w bytes
278	-----		
279		notification-attributes-tag	1 byte
280	-----		
281		ntfy-status-code	2 bytes
282	-----		
283		end-of-attributes-tag	1 byte
284	-----		

- 1 or more

285 6 Encoding of Transport Layer

286 HTTP/1.1 [rfc2616] is the transport layer for this protocol.

287 The operation layer has been designed with the assumption that the transport layer contains the following
 288 information:

- 289 - the URI of the target job or printer operation.
- 290 - the total length of the data in the operation layer, either as a single length or as a sequence of
 291 chunks each with a length.

292 It is REQUIRED that an 'ipp-notify-send' Notification Recipient implementation support HTTP over the
 293 IANA assigned Well Known Port XXX (the 'ipp-notify-send' notification protocol default port), though a
 294 notification recipient implementation MAY support HTTP over some other port as well.

295 Each HTTP operation MUST use the POST method where the request-URI is the object target of the
 296 operation, and where the "Content-Type" of the message-body in each request and response MUST be

297 "application/ipp-notify-send". The message-body MUST contain the operation layer and MUST have the
298 syntax described in section 3, "Encoding of Operation Layer". An 'ipp-notify-send' Notification Source
299 implementation MUST adhere to the rules for a client described for HTTP1.1 [rfc2616]. An 'ipp-notify-
300 send' Notification Recipient implementation MUST adhere the rules for an origin server described for
301 HTTP1.1 [rfc2616].

302 An 'ipp-notify-send' Notification Source sends a response for each request that it receives. If a notification
303 recipient detects an error, it MAY send a response before it has read the entire request. If the HTTP layer of
304 the Notification Recipient completes processing the HTTP headers successfully, it MAY send an
305 intermediate response, such as "100 Continue", with no notification data before sending the notification
306 response. The 'ipp-notify-send' Notification Sources MUST expect such a variety of responses from
307 notification recipients. For further information on HTTP/1.1, consult the HTTP documents [rfc2616].

308 An 'ipp-notify-send' Notification Recipient (server) MUST support chunking for HTTP notification
309 requests, and an 'ipp-notify-send' Notification Source (client) MUST support chunking for HTTP
310 notification responses according to HTTP/1.1[rfc2616]. Note: this rule causes a conflict with non-compliant
311 implementations of HTTP/1.1 that don't support chunking for POST methods, and this rule may cause a
312 conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for CGI scripts

313 7 IANA Considerations

314 IANA will be asked to register this 'ipp-notify-send' notification delivery scheme and protocol and will be
315 asked to assign a default port.

316 8 Internationalization Considerations

317 When the client requests Human Consumable form by supplying the "notify-text-format" operation attribute
318 (see [ipp-ntfy]), the IPP Printer (or any Notification Service that the IPP Printer might be configured to use)
319 supplies and localizes the text value of the "human-readable-report" attribute in the Notification according
320 to the charset and natural language requested in the notification subscription.

321 9 Security Considerations

322 The IPP Model and Semantics document [ipp-mod] discusses high level security requirements (Client
323 Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by
324 which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism
325 by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a
326 mechanism for protecting operations from eavesdropping.

327 The Notification Recipient can cancel unwanted Subscriptions created by other parties without having to be
328 the owner of the subscription by returning the 'successful-ok-but-cancel-subscription' status code in the
329 Send-Notifications response returned to the Notification Source.

330 **9.1 Security Conformance**

331 Notification Sources (client) MAY support Digest Authentication [rfc2617]. If Digest Authentication is
 332 supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be
 333 supported.

334 Notification Recipient (server) MAY support Digest Authentication [rfc2617]. If Digest Authentication is
 335 supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be
 336 supported.

337 Notification Recipients MAY support TLS for client authentication, server authentication and operation
 338 privacy. If a notification recipient supports TLS, it MUST support the
 339 TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite as mandated by RFC 2246 [rfc2246]. All
 340 other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described
 341 in HTTP/1.1 [rfc2616]) for client authentication if the channel is secure. TLS with the above mandated
 342 cipher suite can provide such a secure channel.

343 **10 References**

344 [ipp-mod]

345 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
 346 Semantics", <draft-ietf-ipp-model-v11-04.txt>, June, 1999.

347 [ipp-ntfy]

348 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing
 349 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-02.txt>, February 2,
 350 2000.

351 [ipp-pro]

352 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
 353 Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.

354 [rfc2026]

355 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

356 [rfc2616]

357 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
 358 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

359 [rfc2617]

360 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP
 361 Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

362 11 Author's Addresses

363 Hugo Parra
364 Novell, Inc.
365 122 E 1700 S
366 Provo, UT 84606
367
368 Phone: 801-861-3307
369 Fax: 801-861-2517
370 e-mail: hparra@novell.com
371

372 Tom Hastings
373 Xerox Corporation
374 737 Hawaii St. ESAE 231
375 El Segundo, CA 90245
376
377 Phone: 310-333-6413
378 Fax: 310-333-5514
379 e-mail: hastings@cp10.es.xerox.com
380

381 12 Full Copyright Statement

382 Copyright (C) The Internet Society (1999). All Rights Reserved.

383 This document and translations of it may be copied and furnished to others, and derivative works that
384 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
385 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
386 this paragraph are included on all such copies and derivative works. However, this document itself may not
387 be modified in any way, such as by removing the copyright notice or references to the Internet Society or
388 other Internet organizations, except as needed for the purpose of developing Internet standards in which
389 case the procedures for copyrights defined in the Internet Standards process must be followed, or as
390 required to translate it into languages other than English.

391 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
392 successors or assigns.

393 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
394 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
395 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
396 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
397 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.