

DOCUMENT No. CMI010

Guidelines for Package Exchange Notification Services

ORIGINAL RELEASE DATE 2005-June-27
Revision 1.0a release 2006-March-15

THIS DOCUMENT IS CONTROLLED BY:

AICC CMI Subcommittee

ALL REVISIONS TO THE DOCUMENT SHALL BE APPROVED
BY THE ABOVE ORGANIZATION PRIOR TO RELEASE.

POINT OF CONTACT:

Scott Bergstrom
AICC Administrator
P.O. Box 472
Sugar City, ID 83448-0472

PREPARED ON PC

FILED UNDER CMI010_v1-0a.pdf

Caveats...

© 2005, 2006 AICC
All rights reserved

The information contained in this document has been assembled by the AICC as an informational resource. Neither the AICC nor any of its members assumes nor shall any of them have any responsibility for any use by anyone for any purpose of this document or of the data which it contains.

This page intentionally left blank.

1 **Specification for Learning Technology—**

2 **Package Exchange Notification Services (PENS)**

3 **Abstract:**

4 This specification describes a protocol to support a notification service to announce the loca-
5 tion of content package(s) that are available for transport. The intent is to automate the notifi-
6 cation, transfer and delivery confirmation of content packages between tools or systems that
7 generate content and systems that manage, publish or deliver content. The scope of the speci-
8 fication is specifically constrained to the notification request, package transfer and related re-
9 sponses. Specifically outside the scope of this specification are mechanisms for physical
10 deployment of content packages, content management, version control, publication or revoca-
11 tion of content.

12 **Keywords:**

13 CBT, CMI, CMS, content management, content package, e-learning, LCMS, LMS, notifica-
14 tion service, package exchange, PENS.

15 **Introduction**

16 The purpose of this specification is to fill a gap that currently exists between the creation of
17 content packages by “content authors” and the deployment of those content packages on
18 LMSs by “LMS administrators” where learners may ultimately have access to them. Without
19 a specification that addresses this gap, the concept of shared content is incomplete: LMSs do
20 not have a means to obtain newly developed, revised or updated content.

21 This specification aims to provide a mechanism whereby content that is capable of being
22 shared can be deployed and thus actually shared in practice. It describes a notification scheme
23 that will enable a content creator’s authoring system to announce that a content package is
24 available and ready for transport from a location that it will provide.

25 The acronym for this specification is PENS: Package Exchange Notification Services. The
26 PENS data model may be extended in the future to include commands in addition to the cur-
27 rent “collect” command, which is the first service to be defined. Data elements and value
28 spaces can be extended as driven by needs and determined in the future by the community of
29 users.

30 The scope of the specification is specifically constrained to the notification request, package
31 transfer and related responses. PENS addresses neither version control nor content manage-
32 ment; there are no PENS commands to require the recipient to remove, replace, or update ex-
33 isting packages or elements of packages. PENS provides a contact URI (e.g., email address)
34 for the recipient to contact the requestor, but PENS does not prescribe a specific workflow for
35 processing of the transferred package. PENS does not require notifications to the requestor,
36 other than the specific obligatory confirmation. For illustrative purposes, consider a courier
37 service as a conceptual model for PENS. Two parties may use the courier service as a means
38 of requesting pick-up, performing transfer, and confirming delivery. However, it is not incum-
39 bent on the courier to enforce particular post-processing by the recipient. The recipient may
40 decide to use the parcel as notification to remove something, add the parcel to stock, or replace

41 existing stock. All post-delivery processing is determined by the recipient. The recipient sets
42 its own policy and procedures, and may choose to notify the party that requested delivery of
43 specific events as it sees fit to do.

44 **Discussion forum**

45 The PENS working group communicates via the AICC web forums. To view PENS-related
46 documents and comment on them, go to:

47 <http://www.aicc.org/pages/review.htm>

48

49 **Participants**

50 At the time this specification was completed, the working group had the following active par-
 51 ticipants:

52

Mike Bednar (Pathlore)	Leonard Greenberg (Pathlore)
Bob Benedict (Macromedia)	Scott Hamilton (Plateau)
Khammao Chang (Documentum)	Joe Herman (Plateau)
Andrew Chemey (Macromedia)	Tom King (Macromedia)
Ed Cohen (Plateau)	Brian Quigley (Documentum)
Shawn Daniels (Recombo)	Paul Roberts (Questionmark)
Ryan Edgar (Documentum)	Heather Walls (Scientific & Technical Editing Services)
Silke Fleischer (Macromedia)	Derek Zasiewski (Documentum)
Steven Forth (Recombo)	

53

54 **Corrections**

55 Version 1.0a includes the following corrections:

56 **Section 6.2**

57 The package-url-expiry message element requires a trailing 'Z' to comply with the intent of the
 58 PENS specification which specifies ISO 8601 format with UTC.

59

60 In the CMI010 Revision 1.0 document, the field is described as ISO 8601 with UTC, but the
 61 example does not properly show the UTC time zone code ('Z').

62

63 **Erroneous sample value:** 2005-07-22T06:51:29

64 **Correct sample value:** 2005-07-22T06:51:29Z

65

66 The CMI010 Revision 1.0a document includes a correct sample value.

67	Contents	
68	1 Overview	3
69	1.1 Scope	3
70	1.2 Purpose	3
71	2 References	4
72	3 Definitions	4
73	3.1 Abbreviations and acronyms	5
74	4 Conformance	5
75	4.1 Sending implementations (Client Systems)	5
76	4.2 Receiving implementations (Target Systems)	5
77	4.3 Implementation-defined values	6
78	5 Conceptual Model: Informative	6
79	5.1 Description of use case	9
80	6 Data Model	10
81	6.1 Target system for PENS message	12
82	target system URL.....	12
83	6.2 PENS message elements	12
84	pens-version	12
85	command.....	13
86	package-type	13
87	package-type-version.....	13
88	package-format	13
89	package-id	14
90	package-url.....	14
91	package-url-user-id.....	14
92	package-url-account.....	14
93	package-url-password.....	15
94	package-url-expiry	15
95	client	15
96	system-user-id	15
97	system-password.....	15
98	receipt	16
99	alerts	16
100	vendor-data.....	16
101	6.3 Response and Error Messages	17
102	PENS Responses	17
103	PENS Error Messages	17
104	Appendix A: Binding of PENS to a URI	1
105	1 Binding of PENS Message to a URI	1
106	1.1 Examples of PENS Messages Bound to a URI	3
107	Example of collect command (sample not URL encoded, for readability)	3

108 Example of HTML link for collect command, with the processing/results in a new blank
 109 browser window (URL encoded) 3
 110 Example of PENS receipt command (URL encoded) 3
 111 Example of PENS alert command (URL encoded) 3
 112 **1.2 Binding of PENS Error Message to a URI 4**
 113 **2 Sample PENS transaction stream..... 5**
 114 **2.1 Summary..... 5**
 115 **2.2 Actors..... 5**
 116 **2.3 Stages 5**
 117 1. Author creates package and places on web server 5
 118 2. Author sends COLLECT command to LMS passing URL of package for collection.. 5
 119 3. LMS returns RESPONSE to acknowledge that it understood the COLLECT command
 120 5
 121 4. LMS collects package from Author's URL..... 5
 122 5. LMS sends RECEIPT command to Author to say that the package has successfully
 123 been collected..... 5
 124 6. Author returns RESPONSE to acknowledge that it understood the RECEIPT
 125 command..... 6
 126 7. LMS opens package 6
 127 8. LMS sends ALERT command to Author to say that the package has been opened 6
 128 9. Author returns RESPONSE to acknowledge that it understood the ALERT command6
 129 10. LMS deploys package 6
 130 11. LMS sends ALERT command to Author to say that the package has been deployed 6
 131 12. Author returns RESPONSE to acknowledge that it understood the ALERT command
 132 7
 133 **Bibliography..... 1**

134 **Specification for Learning Technology—** 135 **Package Exchange Notification Services**

136 **1 Overview**

137 The scope and purpose of this specification are discussed in 1.1 and 1.2. This specification
138 describes a protocol to announce the location of content package(s) that are available for
139 transport. The intent is to automate the notification, transfer and delivery confirmation of con-
140 tent packages between tools or systems that generate content and systems that manage, publish
141 or deliver content.

142 **1.1 Scope**

143 This specification describes a data model and binding to facilitate the transport of e-learning
144 content between content publishing systems and systems for the delivery of e-learning such as
145 a learning management system (LMS). This specification describes a notification scheme that
146 is a communication between the content publishing (or authoring) system and the server man-
147 aging delivery. It includes data elements to identify the package format and location.

148 The specification does not specify how the systems must behave after the package is trans-
149 ferred. The scope is specifically constrained to the notification request, package transfer and
150 related responses. Specifically outside the scope of this specification are mechanisms for de-
151 ployment of content packages, content management, version control, publication or revocation
152 of content. The specification does not provide any mechanism for systems to search for con-
153 tent packages.

154 This specification relies on content package formats specified by the Advanced Distributed
155 Learning (ADL) Initiative and the AICC; the definition of content package formats is outside
156 the scope of this specification. The binding in Appendix A is modeled on AICC HACP
157 mechanisms as defined in the AICC data model for Computer Managed Instruction (CMI).

158 **1.2 Purpose**

159 The purpose of this specification is to provide a means to notify targeted systems of the exis-
160 tence of content packages that may be of interest to those systems. The notification scheme
161 defines the format of the message that describes the available content package and from where
162 it can be retrieved. The intent is to simplify content authoring and deployment by supporting
163 automation of the transfer of e-learning content between authoring or publishing systems and
164 systems for managing or deploying content.

165 2 References

166 The following referenced documents are indispensable for the application of this specification.
167 For dated references, only the edition cited applies. For undated references, the latest edition
168 of the referenced document (including any amendments) applies.

169 RFC 1738, "Uniform Resource Locators (URL)," December 1994.

170 RFC 2368, "The mailto URL scheme," July 1998.

171 RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax," August 1998.

172 RFC 2616, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.

173 RFC 2817, "Upgrading to TLS Within HTTP/1.1," May 2000.

174 RFC 2822, "Internet Message Format," April 2001.

175 ISO 8601:2000 "Data elements and interchange formats -- Information interchange -- Repre-
176 sentation of dates and times," Edition 2.

177 3 Definitions

178 For purposes of this specification, the following terms and definitions apply. The AICC Glos-
179 sary [A3], should be referenced for terms not defined here.

180 **Content Package:** The data structures and files used to provide interoperability of digital e-
181 learning material with authoring tools, LMS products and run-time environments. A content
182 package typically contains one or more files that list all of the resources included in the pack-
183 age; data to describe the content structure; data describing prerequisite or sequencing rules,
184 descriptive metadata, and the supporting files (and/or pointers to external resources) that con-
185 stitute the e-learning material. Often the components of a content package are combined into a
186 single-file archive format such as a PKZip v2.04g, conformant to RFC1951.

187 **Client (Sending) System:** A system that initiates a PENS request and ensures that a confor-
188 mant content package is available for delivery. Examples of Client Systems include Authoring
189 Tools, Learning Content Management Systems, Content Management Systems, Middleware
190 Systems, Assessment Systems and other systems that can publish content.

191 **Learning Content Management System (LCMS):** A computer system that includes and
192 combines the functions of a Content Repository or Content Management System and a Learn-
193 ing Management System. The Content Repository functions manage the content and permit
194 locating stored content, authoring new content, attaching metadata to content, managing ver-
195 sions of content, etc. The Learning Management System functions administer courses to learn-
196 ers.

197 **Learning Management System (LMS):** A computer system that may include the capabilities
198 to register learners, schedule learning resources, control and guide the learning process, ana-
199 lyze and report learner performance, and schedule and track learners.

200 **Target (Receiving) System:** A system that receives PENS requests and is responsible for the
201 transfer and import of content packages. The target system is also responsible for validating
202 responses, returning error messages and delivering receipts and alerts. Examples of Target
203 Systems include Learning Management Systems, Learning Content Management Systems,

204 Content Management Systems, Content Repositories, Middleware and any other system that
205 can import content for the purpose of managing or delivering content.

206 **3.1 Abbreviations and acronyms**

207 AICC: Aviation Industry CBT Committee
208 CBT: Computer-Based Training
209 CMI: Computer Managed Instruction
210 CMS: Content Management System
211 HACP: HTTP AICC Communication Protocol
212 LCMS: Learning Content Management System
213 LMS: Learning Management System
214 PENS: Package Exchange Notification Services
215 QTI: Question and Test Interoperability
216 SCORM: Sharable Content Object Reference Model
217 URI: Uniform Resource Identifier
218 URL: Uniform Resource Locator
219 URN: Uniform Resource Name

220 **4 Conformance**

221 Conformance to this specification is discussed in 4.1 – 4.3.

222 In this specification, “shall” is to be interpreted as a requirement on an implementation; “shall
223 not” is to be interpreted as a prohibition.

224 **4.1 Sending implementations (Client Systems)**

225 A conforming sending implementation shall send data instances that conform to this specifica-
226 tion and accept responses from target systems as defined in this specification. A conforming
227 sending implementation shall send all required elements.

228 **4.2 Receiving implementations (Target Systems)**

229 A conforming receiving implementation shall accept data instances that conform to this speci-
230 fication and generate the required, valid responses. Data instances that conform to this specifi-
231 cation include all required elements and may include optional elements.

232 4.3 Implementation-defined values

233 The processing and meanings of values that are not specified by this specification (e.g., senti-
234 nel, missing, and empty values) are implementation-defined.

235 NOTE:

236 For example, implementations may specify the processing or meanings of missing, default
237 values or sentinel values. A Target System implementation might specify that in the absence of
238 another value, an empty password value indicates no password is required.

239 5 Conceptual Model: Informative

240 Synopsis of the package exchange notification services (PENS) model:

- 241 • A notification is sent from a content source (such as an authoring tool, CMS or
242 LCMS) to a Target System (central deployment or repository system such as a CMS,
243 LCMS or LMS).
- 244 • The notification announces the availability and location of a content package that is
245 available for transport.
- 246 • The notification represents the first step in initiating the Target System workflow to
247 transfer and import a content package.

248 Notification mechanism details:

- 249 • Suggested notification mechanism binding: HTTP-GET or HTTP-POST of name-
250 value pairs (see Appendix A, “Binding of PENS Message to a URI”).

251 NOTE:

252 According to RFC 2616 (June 1999), section 3.2.1, “The HTTP protocol does not place any a
253 priori limit on the length of a URI. Servers MUST be able to handle the URI of any resource
254 they serve, and SHOULD be able to handle URIs of unbounded length if they provide GET-
255 based forms that could generate such URIs. A server SHOULD return 414 (Request-URI Too
256 Long) status if a URI is longer than the server can handle. (Servers should be cautious about
257 depending on URI lengths above 255 bytes, because some older client or proxy implementa-
258 tions may not properly support these lengths.)” Also see RFC 2817, Upgrading to TLS within
259 HTTP/1.1, as an update to RFC 2616.

260 The definitive reference for HTTP-GET and HTTP-POST is:

261 <http://www.w3.org/Protocols/Overview.html>

- 262 • Notification modes: can be server-to-server, or server via browser window to server
263 (HTTP-GET only).
- 264 • Responsibilities of sender: The content source (herein referred to as the “Client”)
265 shall arrange for the content package to be made available on a staging server. The
266 Client shall be capable of specifying a URI that uses HTTP, or HTTPS (secure HTTP)
267 protocols. The Client may optionally support specifying FTP and FTPS (secure FTP)
268 protocols and the related access credentials.

269 NOTE:

270 It is assumed that the particular configuration of the staging server may be determined by a
271 third party and therefore is not controlled by the content developer (Client). It is further as-
272 sumed that if content is staged on the server via FTP that it does not have to be retrieved via
273 FTP, but could be retrieved via an HTTP alias. Such provisions allow cases such as the transfer
274 to the staging location via FTP and retrieval via an HTTP equivalent or alias to the same loca-
275 tion.

- 276 • Password: If required by the Client's system, the notification may include a password
277 needed to access the content package.

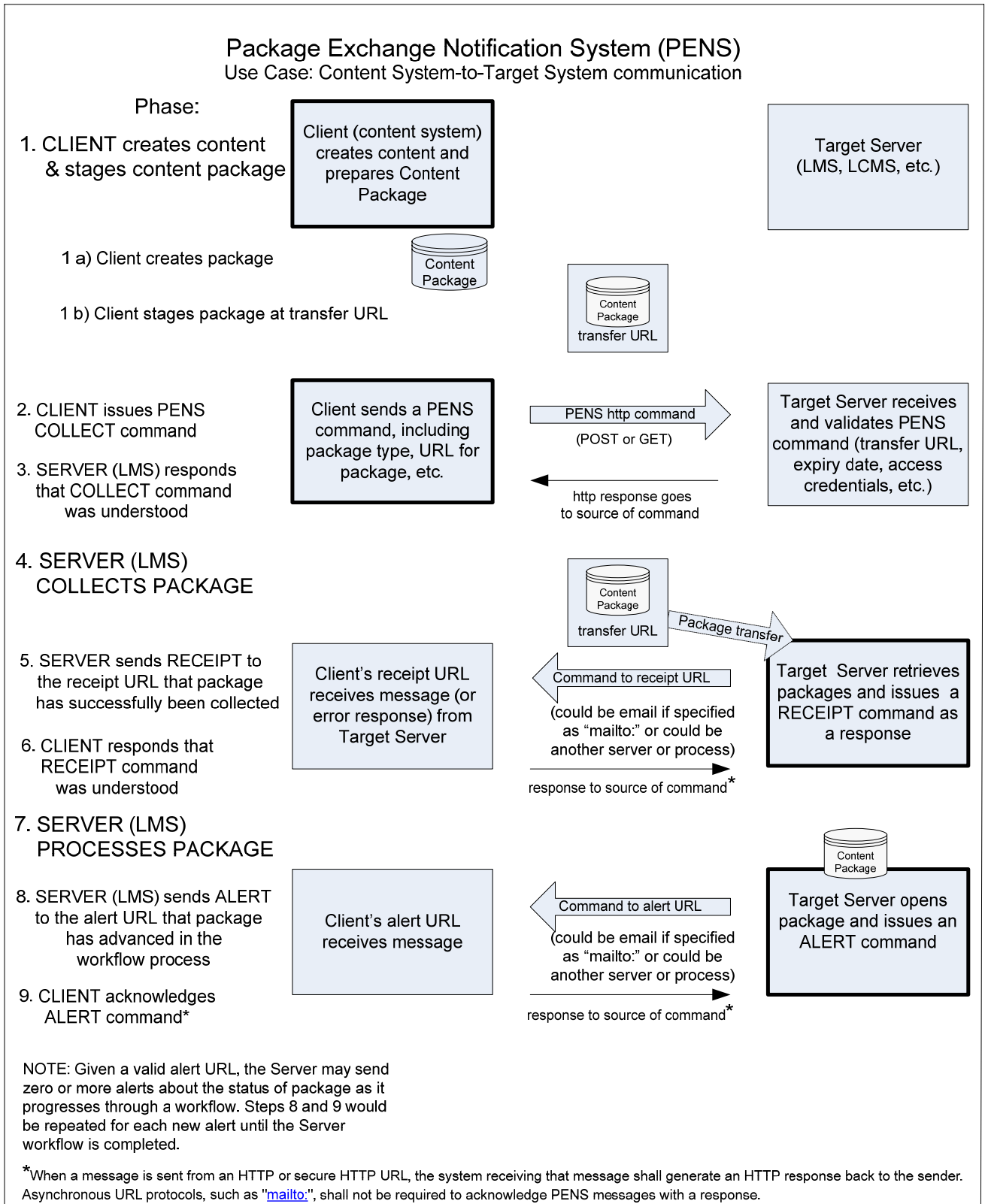
278 **Responsibilities of recipient of notification (Target System):**

- 279 • The notification recipient (herein referred to as the "server") shall be capable of sup-
280 porting both HTTP and HTTPS protocols for the "pull" or "get" transfer of the content
281 package from the URI provided by the Client. The server may optionally support the
282 retrieval of packages specified with FTP or FTPS protocols and appropriate access
283 credentials. In such cases where the server does not support one or more optional pro-
284 tocols, the server is obligated to return the appropriate error message regarding the re-
285 quested protocol.
- 286 • Once the content package is retrieved, store or deploy it (for example, make available
287 via a catalog of resources for a course).

288 **Illustrative use case:**

289 A use case for this specification is shown in Figure 1.

- 290 • Client (authoring system) creates and prepares a content package.
- 291 • Client sends a PENS message to Target System (e.g., an LMS), announcing the avail-
292 ability of the content package.
- 293 • Target System acknowledges PENS message.
- 294 • Target System collects and processes content package.



295
296
297

Figure 1— Conceptual model, Content System-to-Target System communication

298 **5.1 Description of use case**

299 The diagram (Figure 1 above) illustrates the phases that need to occur between a Client Sys-
300 tem and a Target System to issue and process a collect command that transfers a content pack-
301 age.

302 **Preparing the content package and issuing a PENS command**

303 Assume a content author has created some learning content. A content package has been pre-
304 pared and staged by the Client (the authoring system) at a transfer URL (an HTTP site or FTP
305 site) from where it can be collected. The Target System that may ultimately retrieve the cli-
306 ent's content package is typically a CMS, LMS or LCMS product.

307 NOTE:

308 The Client may use FTP or other mechanisms to transfer content to a staging location, yet
309 specify an HTTP URI equivalent as the package-url for retrieval. There is no requirement that
310 the retrieval protocol must match the method used to stage the content. Content staged on the
311 server via FTP could be retrieved via an HTTP alias. Best practices indicate that HTTP is the
312 preferred transport protocol for the package-url.

313 The Client System then sends a message that contains the PENS "collect" command to the
314 Target System (CMS, LMS, LCMS, etc.) via HTTP. This Package Exchange Notification
315 Services (PENS) message includes the elements detailed in the table in 6.2. These PENS ele-
316 ments provide information about the type of content package, where it is available, how to
317 communicate with the client and other parameters. The vendor-data element is available for
318 *optional* data defined by the target system implementation. This implementation-specific data
319 might be used to provide information from the content provider about how a target system is
320 to act on a package after retrieval.

321 Next, the Target System (e.g., LMS) validates the PENS message (transfer URL, content
322 package expiration date, any required access credentials or passwords, etc.) and sends an
323 HTTP response back to the Client System. (An HTTP error code is sent if there was a problem
324 with the PENS message. See 6.3, Response and Error Messages.) This response simply ac-
325 knowledges that the PENS collect command was understood; it does not imply that processing
326 to actually retrieve the package has commenced.

327 NOTE:

328 In another scenario, the Client System could open a browser window to send the PENS mes-
329 sage, and the HTTP response from the Target System would be returned to there.

330 **Collecting the content package**

331 If a server receives a valid collect command, the server can attempt to retrieve the package
332 from the transfer URL via FTP, HTTP or HTTPS. (NOTE: per best practices, HTTP is the
333 preferred protocol for package retrieval.) After attempting to retrieve a package, the server
334 sends a response to the specified receipt URL, either acknowledging successful collection of
335 the package or reporting an error. Receipt responses can be sent to one or more email ad-
336 dresses if so specified in the PENS message.

337 The format of the receipt message sent to the specified receipt URL is not fully defined, but
338 should include relevant data from the inbound Collect Command, such as the package type,
339 package id, and client, plus the receipt message itself.

340 If an error has occurred, an error message should be sent to the receipt URL. The “pens-data”
341 portion of the error message could include extended information about the nature of the error
342 (such as a stack trace of what happened during the attempted communication). See Section
343 6.3 Response and Error Messages for details.

344 **Opening and deploying a content package**

345 The recipient system will proceed with internal processing, such as opening the package, ap-
346 proving the content for release, listing the new content in a catalog, staging the content on a
347 deployment server, etc.

348 NOTE:

349 Internal processing phases, workflow and alert triggers are implementation-specific and are
350 outside the scope of this specification. However, the ‘vendor-data’ may supply useful hints
351 from the content provider about how particular recipient systems are to act on the package sub-
352 sequent to retrieval.

353 If an alert URL has been provided, the server may send alerts to the Client about the status of
354 the package as it progresses through these various processes. For example, an “alert” message
355 may be sent to the content authoring management staff (the Client) so that where the retrieved
356 content is being cataloged, deployed, etc., can be monitored. Alerts can be sent to one or more
357 email addresses.

358 The format of the alert message(s) sent to the specified alert URL is not fully defined, but
359 should include relevant data from the inbound Collect Command, such as the package type,
360 package id, and client, plus the alert message itself.

361 **6 Data Model**

362 This defines the PENS data model that enables the sending of a notification message from au-
363 thoring tools to target systems to announce the description and location of content packages
364 that are available for transfer.

365 This specification does not define which servers are notified, nor does it define what happens
366 to any package that has been retrieved.

367 The table below summarizes the components of the model, which are then defined in the sub-
368 sections indicated.

Component	Required	Data Type	Sub-section
target system URL	Yes	URL	6.1.1
pens-version	Yes	x.x.x (string of three integers separated by periods)	6.2.1
command	Yes	Reserved words, pre-defined character strings	6.2.2
package-type	Yes	Reserved words, pre-defined character strings	6.2.3
package-type-version	Yes	Character string	6.2.4
package-format	Yes	Character string	6.2.5
package-id	Yes	A URI according to RFC 2396	6.2.6
package-url	Yes	URL, URL-encoded string	6.2.7
package-url-user-id	No	Character string	6.2.8
package-url-account	No	Character string	6.2.9
package-url-password	No	Character string	6.2.10
package-url-expiry	Yes	ISO 8601 format expressed as UTC	6.2.11
client	Yes	Character string	6.2.12
system-user-id	No	Character string	6.2.13
system-password	No	Character string	6.2.14
receipt	Yes	Character string	6.2.15
alerts	No	Character string	6.2.16
vendor-data	No	Character string	6.2.17

369

370 ASSUMPTIONS/NOTES:

371 1—The authoring tool has a method for sending the messages/password to the target system
372 URL; automatic discovery of the package by LMSs is out of scope.

373 2—Messages/passwords are sent as clear text; encryption and security issues are out of scope.

374 3—PENS message focuses on the transport and overall package type.

375 4—Receiving system sends an HTTP acknowledgement of message receipt.

376 5—After a package transfer is achieved, receiver responds with an acknowledgement or an er-
377 ror message to a receipt URI.

378 6—How the package is processed upon retrieval is left to the Target System implementation.
379 Notifications of processing errors or events shall be sent to the alert URI.

380 7—Acknowledgment and Error responses shall be in the format specified in 6.3, Response and
381 Error Messages. (Format is similar to AICC HACP error responses.)

382 8—This specification does not define an extension mechanism for the data model. Implemen-
383 ters may create additional data models for package exchange notification. Such models may be
384 used to augment this model to support different communities of practice.

385 6.1 Target system for PENS message

386 This is a sample URL for a system that might receive and process a valid PENS command.

6.1 Target system for PENS message	
target system URL	
<p>Required: Yes</p> <p>Data type: URL</p>	<p>Description: Fully qualified URL of target system that will perform processing.</p> <p>Value space: Valid, fully qualified URI, including transport protocol (e.g., http://)</p> <p>Example: http://acmelearning.lms.com</p>

387 6.2 PENS message elements

388 *Reserved words:*

389 The following words are not utilized in this version of PENS, but are reserved for use in sub-
390 sequent versions as potential candidates for PENS commands.

391 • Delete

392 • Revise

393

pens-version	
<p>Required: Yes</p> <p>Data type: x.x.x (string of three integers separated by periods)</p>	<p>Description: Version of package exchange notification service protocol used by client submitting the package.</p> <p>Value space: Values defined by releases of the specification</p> <p>Sample element value: 1.0.0</p>

command	
<p>Required: Yes</p> <p>Data type: Reserved words, pre-defined character strings</p>	<p>Description: Command for an action that client submitting the package is requesting for the target system to perform. May include the capability to perform a preview of the content in the system's run-time environment.</p> <p>Value space: Fixed values defined by specification.</p> <p>Vocabulary: At this time only "collect" is defined. See 6.2 for reserved words. collect: To retrieve a content package from a designated server.</p> <p>Sample element value: collect</p>
package-type	
<p>Required: Yes</p> <p>Data type: Reserved words, pre-defined character strings</p>	<p>Description: Allowable types of content packages. Other types: AICC assignable unit (aicc-au) or SCORM package (scorm-pif).</p> <p>Value space: Fixed values defined by specification.</p> <p>Vocabulary: aicc-pkg: scorm-pif: lms-qli:</p> <p>Sample element value: aicc-pkg</p>
package-type-version	
<p>Required: Yes</p> <p>Data type: Character string</p>	<p>Description: Identifies the version of the packaging specification relevant for the package to be processed, e.g., for ADL SCORM "scorm-pif" packages, a system might use "1.2" or "2004".</p> <p>Value space: Values defined by associated package spec releases, such as ADL SCORM, AICC CMI, or IMS QTI specifications.</p> <p>Sample element value: 1.0</p>
package-format	
<p>Required: Yes</p>	<p>Description: Identifies a package as being one of the allowable package archive</p>

<p>Data type: Character string</p>	<p>formats or resource types.</p> <p>Value space: Values defined by package archive format or resource types. Reserved values include: “zip”, “url”, “jar”, “war”, and “xml”.</p> <p>Sample element value: zip</p>
package-id	
<p>Required: Yes</p> <p>Data type: A URI according to RFC 2396</p>	<p>Description: Unique identifier required for package; package-id shall be a URI consisting of two parts, a globally unique namespace taken from the URL associated with the product or the service generating the ID plus an ID unique within the service itself.</p> <p>Value space: Any URI according to RFC 2396 with the additional requirement that the URI shall be constructed such that its namespace is the URL associated with the product or service generating the ID and the id of the package is unique within that namespace.</p> <p>Sample element value: http://myurl.com:2631e419-1573-4720-b4c6-8701f960dccc</p>
package-url	
<p>Required: Yes</p> <p>Data type: URL, URL-encoded string</p>	<p>Description: Location of package archive ready for transfer/action.</p> <p>Value space: Valid, fully qualified URL, including transport protocol (e.g., http:// or ftp://) and filename including extension.</p> <p>Sample element value: http://myauthoringtool/mycontentpackage.zip</p>
package-url-user-id	
<p>Required: No</p> <p>Data type: Character string</p>	<p>Description: User id required for system to retrieve package from URL.</p> <p>Value space: Null string or character string.</p> <p>Sample element value: (null string)</p>
package-url-account	
<p>Required: No</p> <p>Data type: Character string</p>	<p>Description: Account required for system to retrieve package from URL.</p> <p>Value space: Null string or character string.</p> <p>Sample element value:</p>

	(null string)
package-url-password	
Required: No Data type: Character string	Description: Password required for system to retrieve package from URL. Value space: Null string or character string. Sample element value: (null string)
package-url-expiry	
Required: Yes Data type: ISO 8601 format expressed as UTC	Description: The package is expected to be available for processing until at least the date and time specified. Value space: Null string or character string. Sample element value: 2005-07-22T06:51:29Z
client	
Required: Yes Data type: Character string	Description: Name or ID for client submitting the content package to the target system. Other examples: PerceptionForWeb; Captivate. Value space: Null string or character string. Sample element value: Authorware7
system-user-id	
Required: No Data type: Character string	Description: User id or sign-on for target system, or a null string. Value space: Null string or character string. Sample element value: tk007
system-password	
Required: No Data type: Character string	Description: Either a URL-encoded password token or the null string. If the target system requires a password and the null string value is passed, then the target system is responsible for prompting for a password for target system. Value space:

	<p>Null string or character string.</p> <p>Sample element value: (null string)</p>
receipt	
<p>Required: Yes</p> <p>Data type: Character string</p>	<p>Description: URL to send acknowledgement receipt after collecting a package; if “mailto:” URL is used, it may include more than one address, with addresses separated by commas per RFC 2368 and RFC 2822.</p> <p>Value space: Any URL, including “mailto:” URL scheme per RFC 2368 and RFC 2822.</p> <p>Sample element value: mailto:name@domain.com</p>
alerts	
<p>Required: No</p> <p>Data type: Character string</p>	<p>Description: URL to send alerts to while processing the package (after the acknowledgment to ‘receipt’ URL). If the alert URL is a “mailto:” URL, it may include more than one address, with addresses separated by commas per RFC 2368 and RFC 2822. The alert response format is the same as that for ‘receipt’.</p> <p>NOTE: Unlike the receipt URL, “alerts” is optional and multiple messages may be sent to the alert URL over an extended period as the package is processed through the host workflow.</p> <p>Value space: Any URL, including “mailto:” URL scheme per RFC 2368 and RFC 2822.</p> <p>Sample element value: mailto:name@domain.com</p>
vendor-data	
<p>Required: No</p> <p>Data type: Character string</p>	<p>Description: Unstructured character string that may be used to transfer vendor-specific data such as processing hints or deployment information.</p> <p>NOTE: Conforming implementations of target systems shall be capable of processing valid PENS commands containing this element, regardless of the system’s ability to parse or act on the value. The value of this element is likely to be unique across implementations. PENS alerts are the mechanism for systems to provide acknowledgements or warnings regarding vendor-data values.</p> <p>Value space:</p>

	Null string or character string, Smallest Permitted Maximum, 4096 characters NOTE: Size may increase if URL-encoding is a requirement of a particular binding. Sample element value: (null string)
--	--

394 6.3 Response and Error Messages

395 Various communication problems can occur between the client offering the package and the
 396 potential receiving (collecting) systems. Responses and error messages provide a means of
 397 acknowledge and communication of processing errors.

398

399 PENS Responses

400 The receipt and alert URI may use http:// or mailto: protocols. If the HTTP protocol is used,
 401 the system receiving the message shall respond with an HTTP response. A response is not re-
 402 quired for mailto: URI for either receipt or alert.

403 Sample responses below are informative only and do not represent requirement specifications.
 404 See Appendix A for URI binding information specifications.

405

406 **Sample HTTP response to COLLECT command:**

```
407     error=0
408     error-text=collect command received and understood
409     version=1.0.0
410     pens-data=
411
```

412 **Sample HTTP response to RECEIPT command:**

```
413     error=0
414     error-text=receipt command received and understood
415     version=1.0.0
416     pens-data=
417
```

418 **Sample HTTP response to ALERT command:**

```
419     error=0
420     error-text=alert command received and understood
421     version=1.0.0
422     pens-data=
423
```

424 PENS Error Messages

425 An error code shall be used to acknowledge successful processing or to indicate a processing
 426 error. In the case of a processing error, the error code shall indicate the class of error as spe-
 427 cifically as possible using either a PENS specific error code or an error code from the underly-

428 ing protocol, as indicated in table1, table 2, and table 3 below. The binding specification for
 429 error code communication is in Appendix A.

430

431

Table 1 —Response Error Codes

Code Number	Error Text
0	No error, successful
other	See errors as listed in Table 3 – Specific Error Codes

432

433 Error codes are integers represented as character strings, in the range of 0 to 65, 535. Unspeci-
 434 fied error codes in the range of 0 to 10000 are reserved for use in future editions of the PENS
 435 specification. Error codes with numbers 10000 and above are reserved for implementation-
 436 defined error messages.

437 NOTE:

438 The error code numbering scheme was established such that PENS codes start at 1000 to avoid colli-
 439 sion with established error codes for underlying protocols (such as 400 series HTTP error codes). These
 440 underlying codes shall be used when they are indicative of the error encountered.

441

Table 2 — Classification of Error Codes

Range	Error Type
100 to 200	
200 to 500	Underlying protocol errors, e.g., HTTP errors
1000 to 1199	General PENS errors
1200 to 1299	PENS syntax errors
1300 to 1399	PENS transport errors
1400 to 1499	Host system errors
1500 to 1599	Acknowledgement and Alert errors
2000 to 9999	PENS parameter errors

442

443 Rationale: Errors are listed in increasing specificity; systems are to respond with highest num-
 444 bered error encountered. Error codes designated as warnings may allow some degree of pack-
 445 age processing (may not be fatal errors).

446

Table 3— PENS-Specific Error Codes

Code Number	Name	Descriptive Text
1101	General error	Unable to parse PENS command
1201	General argument error	Attempt to pass an invalid argument
1301	General retrieve error	Unable to retrieve package
1302	Secure HTTP protocol not supported	Unable to retrieve package via HTTPS
1304	FTP protocol not supported	Unable to retrieve package via FTP
1306	Secure FTP protocol not supported	Unable to retrieve package via FTPS
1310	Invalid or unresponsive package URL	Unable to retrieve package at specified URL due to error in URL or lack of response from URL
1312	Invalid package access credentials	Unable to retrieve package at specified URL due to error with access credential for package URL
1320	Warning - invalid expiry date	Expiration date is non-null and in an improper format
1322	Warning - expired package	Current time indicates expiry date has passed
1420	PENS version not supported	Insufficient permission
1421	Command not supported	Client has requested host to execute an invalid, unknown or unsupported command
1430	Package type not supported	Client has requested host to process an invalid, unknown or unsupported package type
1432	Internal package error	Host unable to process package after successfully retrieving it because of an error with package archive or package contents
1440	Insufficient host space/storage available	Host unable to process package due to local storage space or account restrictions
1500	General acknowledgment error	Unable to communicate with provided acknowledgement URL

1510	Unsupported acknowledgement protocol	Unsupported acknowledgement protocol
1520	Unsupported alert protocol	Unsupported alert protocol
2001	pens-version parameter missing	Message incomplete; PENS version invalid or not specified
2002	command parameter missing	Message incomplete; PENS command invalid or not specified
2003	package-type parameter missing	Message incomplete; package-type invalid or not specified
2004	package-type-version parameter missing	Message incomplete; package-type-version invalid or not specified
2005	package-format parameter missing	Message incomplete; package-format invalid or not specified
2007	package-id parameter missing	Message incomplete; package-id invalid or not specified
2008	package-url parameter missing	Message incomplete; package-url invalid or not specified
2009	package-url-expiry parameter missing	Message incomplete; package-url-expiry date invalid or not specified
2010	client parameter missing	Message incomplete; client submitting package invalid or not specified
2011	receipt parameter missing	Message incomplete; where to send response invalid or not specified

447

448 **Appendix A: Binding of PENS to a URI**

449 **1 Binding of PENS Message to a URI**

450 The PENS message shall be a valid URI [RFC 1738 and RFC 2396].

451 The URI shall consist of 4 components, as described in RFC 2396,

452 `<scheme>://<authority><path>?<query>`

453 The target system URL from the PENS data model (6.1) shall be used to create the `<scheme>`,
454 `<authority>` and `<path>` portions of the URI.

455 This portion of the URI shall be followed by the ASCII character “?” to indicate the start of
456 the `<query>` component of the URI, as per RFC 2396.

457 The query component of the URI shall be an unordered list of message elements.

458 Each element shall contain an element name, as specified in 6.2, followed by the ASCII char-
459 acter “=” followed by the value for the message element. The value for the message element
460 shall be the lexical character encoding of the value from the value space for the elements de-
461 fined in 6.2. Message elements shall be separated by the ASCII character “&”.

- 462 • query = message element * (“&” message element)
- 463 • message element = pens element name “=” lexical element value

464 The entire URI shall be URI-encoded as per RFC 1738 and RFC 2396. Examples of message
465 element values, properly URI-encoded, are shown in the table below.

466 **Table A-1 — Examples of Binding of Individual PENS Message Elements**

PENS Message Element Name	Example URI Binding	Sub-section
pens-version	pens-version=1.0.0	6.2.1
command	command=collect	6.2.2
package-type	package-type=aicc-pkg	6.2.3
package-type-version	package-type-version=1.0	6.2.4
package-format	package-format=zip	6.2.5
package-id	package-id=http%3A%2F%2Fmyurl.com%3A2631e419-1573-4720-b4c6-8701f960dccc	6.2.6
package-url	package-url=http%3A%2F%2Fmyauthoringtool%2Fmycontentpackage.zip	6.2.7
package-url-user-id	package-url-user-id=	6.2.8

package-url-account	package-url-account=	6.2.9
package-url-password	package-url-password=	6.2.10
package-url-expiry	package-url-expiry=2005-07-22T06%3A51%3A29	6.2.11
client	client=Authorware7	6.2.12
system-user-id	system-user-id=tk007	6.2.13
system-password	system-password=	6.2.14
receipt	receipt=mailto%3Aname%40domain.com	6.2.15
alerts	alerts=mailto%3Aname%40domain.com	6.2.16
vendor-data	vendor-data=preview-mode%3Ainstructor	6.2.17

467 NOTE:

468 For the URI binding, the best practice is to use HTTP POST to issue the PENS message. Use of POST
469 avoids a potential issue with two query separators (“?”) in the PENS message URI when the target sys-
470 tem URL itself uses the format

471 <scheme>://<authority><path>?<query>

472 (example: Target System URL is <http://acmelearning.lms.com?partition=staging1>).

473

474 NOTE:

475 For cases when either the alert or the receipt value specifies the “mailto:” protocol the following best
476 practice is recommended for the corresponding return command message in email format.

- 477 1. Use a valid email address of an account associated with the system issuing the response for the
478 sender and reply-to email addresses
- 479 2. Use a suitable human-readable subject line for the email.
- 480 3. Place any optional, implementation-specific information for the recipient in the body of the
481 message *before* the PENS response data, Use a carriage return linefeed pair to separate this in-
482 formation from the PENS response data.
- 483 4. Include all required data element names and the corresponding values of the PENS response in
484 the body of the email. Preferred formatting is a data element name followed by “=” then the
485 corresponding value with a carriage return line feed pair after the value.
- 486 5. As indicated in Section 5.1, the format of the alert message is not fully defined, but should in-
487 clude relevant data from the inbound Collect Command, such as the package type, package id,
488 and client, plus the alert message value itself.
- 489 6. The email format is not fully defined. Implementers are encouraged to use the http: or https:
490 protocol for automated processing of inbound alerts and receipts; machine-based automated
491 processing of inbound email is discouraged.

492

493 **1.1 Examples of PENS Messages Bound to a URI**

494 In the examples that follow, line breaks are not significant; line breaks appear only so that the
495 examples would fit on a printed page.

496 **Example of collect command (sample not URL encoded, for read-
497 ability)**

```
498 http://acmelearning.lms.com/pens?pens-version=1.0.0
499 &command=collect&package-type=aicc-pkg&package-type-
500 version=1.0&package-format=zip&package-id=http://myurl.com:
501 2631e419-1573-4720-b4c6-8701f960dccc&package-url=http://myau-
502 thoringtool/mycontentpackage.zip&package-url-user-id=&package-
503 url-account=&package-url-password=&package-url-expiry=2005-07-
504 22T06:51:29&client=Authorware7&system-user-id=tk007&system-
505 password=&receipt=mailto:name@domain.com
506 &alerts=mailto:name@domain.com
```

507 **Example of HTML link for collect command, with the process-
508 ing/results in a new blank browser window (URL encoded)**

```
509 <a href="http://acmelearning.lms.com/pens?pens-version=
510 1.0.0&command=collect&package-type=aicc-pkg&package-type-
511 version=1.0&package-format=zip&package-id=http%3A%2F%2F
512 myurl.com%3A2631e419-1573-4720-b4c6-8701f960dccc &package-
513 url=http%3A%2F%2Fmyauthoringtool%2Fmycontentpackage.zip
514 &package-url-user-id=&package-url-account=&package-url-
515 password=&package-url-expiry=2005-07-22T06%3A51%3A29
516 &client=Authorware7&system-user-id=tk007&system-password=
517 &receipt=mailto%3Aname%40domain.com&alerts=mailto%3Aname%40dom
518 ain.com" target="_blank">Submit Package</a>
```

519 **Example of PENS receipt command (URL encoded)**

```
520 http://author.com/pens.cgi?command=receipt&pens-version=
521 1.0.0&package-type=scorm-pif&package-type-version=1.2
522 &package-format=zip&package-id=http%3A%2F%2Fmyurl.com
523 %3A994646572378864600-1085069139609&package-url=
524 http%3A%2F%2Fauthor.com%2Fpackages%2F1085069139609.zip
525 &package-url-expiry=2005-05-20T16%3A05%3A39Z&client=LMS
526 &error=0&error-text=package%20sucessfully%20collected
```

527 **Example of PENS alert command (URL encoded)**

```
528 http://author.com/pens.cgi?command=alert&pens-version=
529 1.0.0&package-type=scorm-pif&package-type-version=1.2
530 &package-format=zip&package-id=http%3A%2F%2Fmyurl.com
531 %3A994646572378864600-1085069139609&package-url=
532 http%3A%2F%2Fauthor.com%2Fpackages%2F1085069139609.zip
533 &package-url-expiry=2005-05-20T16%3A05%3A39Z&client=LMS
534 &error=0&error-text=package%20sucessfully%20deployed
```

535 1.2 Binding of PENS Error Message to a URI

536 The Error Message Format is defined in Table A-2; HTTP Response Error Codes are given in
 537 Section 6.3 Table 1; Classification of PENS Error Codes and Descriptive Text representing
 538 specific error situations are listed in Section 6.3 Tables 2 and 3.

539 **Table A-2 — Response & Error Message Format**

<Name>	<Value>
error=	<pens error code number><CR LF>
error-text=	<pens error description><CR LF>
version=	<pens spec version><CR LF>
pens-data=	<pens data> <end of buffer>

540

541 **Example:**

```
542 error=0
543 error-text=
544 version=1.0.0
545 pens-data=
546
```

547 The <value> data is in plain text, and is not URL-encoded.

548 The end-of-line marker is CR LF (carriage return linefeed) per RFC 2616: HTTP/1.1.

549 CR = <US-ASCII CR, carriage return (13)>

550 LF = <US-ASCII LF, linefeed (10)>

551 The hexadecimal values for CRLF acronyms for ASCII systems are OD, OA; hexadecimal
 552 mapping for Unicode systems is OxOD, OxOA (per Unicode Technical Report #13, "Unicode
 553 Newline Guidelines").

554 For http responses, the content type in the HTTP content header should be "Content:
 555 text/plain" with the <CR><LF> pairs as indicated.

556 2 Sample PENS transaction stream

557 2.1 Summary

558 Outline example showing PENS commands and responses during the lifecycle of the success-
559 ful deployment of a package from an Author to an LMS.

560 2.2 Actors

- 561 • Author - <http://author.com/pens.cgi> - Creates packages to send to LMS
- 562 • LMS - <http://lms.com/pens.cgi> - Receives packages from Author to deploy and deliver

563 2.3 Stages

564 1. Author creates package and places on web server

565

566 2. Author sends COLLECT command to LMS passing URL of pack- 567 age for collection

```
568 http://lms.com/pens.cgi?command=collect&pens-
569 version=1.0.0&package-type=scorm-pif&package-type-
570 version=1.2&package-format=zip&package-id=
571 http%3A%2F%2Fwww.author.com%3A994646572378864600-
572 1085069139609&package-
573 url=http%3A%2F%2Fauthor.com%2Fpackages%2F1085069139609.zip&pac
574 kage-url-expiry=2005-05-20T16%3A05%3A39Z&client=Author
575 &receipt=http%3A%2F%2Fauthor.com%2Fpens.cgi
576 &alerts=http%3A%2Fauthor.com%2Fpens.cgi
```

577 3. LMS returns RESPONSE to acknowledge that it understood the 578 COLLECT command

```
579 error=0
580 error-text=collect command received and understood
581 version=1.0.0
582 pens-data=
```

583 4. LMS collects package from Author's URL

584

585 5. LMS sends RECEIPT command to Author to say that the pack- 586 age has successfully been collected

```
587 http://author.com/pens.cgi?command=receipt&pens-
588 version=1.0.0&package-type=scorm-pif&package-type-
589 version=1.2&package-format=zip&package-id=
```

590 http%3A%2F%2Fwww.author.com%3A994646572378864600-
 591 1085069139609&package-
 592 url=http%3A%2F%2Fauthor.com%2Fpackages%2F1085069139609.zip&pac
 593 kage-url-expiry=2005-05-20T16%3A05%3A39Z&client=LMS
 594 &error=0&error-text=package%20sucessfully%20collected

595

596 **6. Author returns RESPONSE to acknowledge that it understood**
 597 **the RECEIPT command**

598 error=0
 599 error-text=receipt command received and understood
 600 version=1.0.0
 601 pens-data=

602 **7. LMS opens package**

603

604 **8. LMS sends ALERT command to Author to say that the package**
 605 **has been opened**

606 http://author.com/pens.cgi?command=alert&pens-
 607 version=1.0.0&package-type=scorm-pif&package-type-
 608 version=1.2&package-format=zip&package-id=
 609 http%3A%2F%2Fwww.author.com%3A994646572378864600-
 610 1085069139609&package-
 611 url=http%3A%2F%2Fauthor.com%2Fpackages%2F1085069139609.zip&pac
 612 kage-url-expiry=2005-05-20T16%3A05%3A39Z&client=LMS
 613 &error=0&error-text=package%20sucessfully%20opened

614 **9. Author returns RESPONSE to acknowledge that it understood**
 615 **the ALERT command**

616 error=0
 617 error-text=alert command received and understood
 618 version=1.0.0
 619 pens-data=

620 **10. LMS deploys package**

621

622 **11. LMS sends ALERT command to Author to say that the package**
 623 **has been deployed**

624 http://author.com/pens.cgi?command=alert&pens-
 625 version=1.0.0&package-type=scorm-pif&package-type-
 626 version=1.2&package-format=zip&package-id=
 627 http%3A%2F%2Fwww.author.com%3A994646572378864600-
 628 1085069139609&package-
 629 url=http%3A%2F%2Fauthor.com%2Fpackages%2F1085069139609.zip&pac
 630 kage-url-expiry=2005-05-20T16%3A05%3A39Z&client=LMS
 631 &error=0&error-text=package%20sucessfully%20deployed

632 **12. Author returns RESPONSE to acknowledge that it understood**
633 **the ALERT command**

634 error=0
635 error-text=alert command received and understood
636 version=1.0.0
637 pens-data=

638 **Bibliography**

639 [A1] AICC CMI001, CMI Guidelines For Interoperability, Version 3.4, April 2001.

640 [A2] SCORM – <http://www.adlnet.org/index.cfm?fuseaction=scormabt>

641 [A3] AICC CRS002, Glossary of Terms Related to Computer Based Training (CBT)

642