



Enterprise Rhaptos Request for Proposal

Table of contents

Connexions Overview	1
Goal for this RFP (Enterprise Rhaptos)	2
High Level Tasks	3
Detailed Tasks	3
1. Simplify installation, development, and testing environment	3
2. Virtualization	5
3. Duplicate import pathways for scaling in a virtualized environment ..	6
4. Production and development templates and guides	6
Deliverables	7
Background Information Useful for Preparing a Proposal	9
Proposal Submission Information	11
Rubric for Evaluating Submissions	12

Connexions Overview

Connexions' mission is to provide a global, interconnected, reusable repository of textbooks, teaching materials and learning materials. The impact of open educational resources (OER) becomes exponentially more valuable when the repository of content reaches critical mass, providing an interconnected, reusable, graph of knowledge. In order to do that, we seek to become the platform and repository of choice for open education resources.

Connexions open-source software (Rhaptos) enables the creation, sharing, modification, and vetting of open educational material accessible to anyone, anywhere, anytime via the World Wide Web. Connexions unique features of permanent versions of all materials, a simple, semantic document language, and a powerful lensing system for post-publication quality control, customized tagging, and community-based search and discovery are the engine behind a truly reusable repository of learning.

Today Connexions is one of the most-used open education resources on the web, employed in traditional college and K-12 settings, in distance learning, and by lifelong learners around the globe. The Connexions servers handle over 1,000,000 visitors per month from 196 countries. With its flexibility (Connexions' ability to produce low-cost textbooks was applauded in a May 1, 2007 New York Times editorial) and grassroots orientation (volunteer communities have, for example, translated material into Spanish and Chinese), Connexions aims to become the platform and repository of choice for open education resources.

An overview of Connexions vision and strategy was provide by foundation Richard Baraniuk at TED 2006:
http://ted.com/tedtalks/tedtalksplayer.cfm?key=r_baraniuk

The Connexions code consists logically of three major functions:

1. **Content repository** of modules, collections and lenses: **Modules** treat a single educational topic and are stored using a semantic XML format. Once published, modules have unique and permanent URLs so that the material is available reliably, indefinitely. Authors may create new versions easily to update materials and all versions are permanently available. Non-authors can derive copies to create specialized versions and proper attribution is maintained to the original source. **Collections** are ordered sets of modules to create textbooks, courses, reports, journals etc. **Lenses** are the mechanism for quality control in Connexions where organizations and individuals select content from the repository and tag and comment on it for their community of practice.
2. **Viewing, printing and content feeds**: The Connexions software renders all the content at <http://cnx.org>, including browse and search features, and various ancillary services such as displaying related materials and maintaining publicly available viewing statistics for modules and collections. Automated PDF generation is available for both modules and collections and as new versions are created, the PDF's are automatically regenerated. For collections, the PDF generation includes automatic generation of a table of contents, index, glossaries, and link resolution (page numbers internally, and footnoted URLs for links outside the collection). Print on demand service is available for collections with QOOP's print-on-demand service. RSS feeds are available for new and revised content and new and revised lenses. Standards based discovery and search feeds are also available for OAI-PMH (Dublin Core and IMS) and OpenSearch.
3. **Authoring tools**: Connexions provides authors with shareable workspaces for editing their educational content, web editing capability for modules, collections, and lenses, and import tools for translating works written in Word, OpenOffice or LaTeX into the XML format (CNXML) used by modules.

Connexions (<http://cnx.org>) and Rhaptos: Connexions is a global, open repository of educational materials and a set of tools for contributing to the repository and reviewing content in the repository. The software running cnx.org is a branded version of Rhaptos, the base software that creates a permanent repository, the authoring tools, and the lensing (review) tools.

Connexions sustainability strategy over the next several years depends on creation of an active code and content development consortium, including academic and commercial partners engaged in either participating in cnx.org, the global world-wide open education repository, or in running individual instances of the Rhaptos software serving internal education needs. The deliverables outlined in this RFP are intended to provide the necessary support to enable production and development teams outside of the Connexions core team to run the code and enhance the code features.

Goal for this RFP (Enterprise Rhaptos)

Enterprise Rhaptos will streamline installing, deploying, and scaling Rhaptos' and Connexions' open-source software and enable development teams from the Connexions consortium to enhance and improve the software.

At the same time, all of the improvements will directly improve the efficiency of the core Connexions development team. Each of the high level tasks below address barriers to adopting, deploying, and developing the Connexions/Rhaptos platform.

High Level Tasks

1. Simplify installation, development and testing environment.

Development and installation of Rhaptos and Connexions code is hampered by a system that relies heavily on copying of an existing repository, carefully coordinating with Connexions staff on code installation, and a nearly 100% hand-testing environment. The first priority for Enterprise Rhaptos is to correct each of these three limitations.

2. Virtualize the Connexions platform

Virtualization of the Connexions platform will allow production environments to be scaled as demand increases and decreases, will be fault tolerant regarding hardware malfunction, will support rapid and more realistic testing, and will support providing demonstration instances of the software when needed. Virtualization will also make providing hosting services for Rhaptos instances a cost-effective service.

3. Duplicate import pathways for scaling in a virtualized environment.

One of the chief mechanisms for creating content in Connexions is by importing content from Word or OpenOffice. Because the translation services are provided by large, third-party software services that are expensive to run, it is difficult to service large numbers of simultaneous imports. Connexions repository is growing at a rate that suggests this will become critical in the near future. Even small, independent repositories will often need to upload bursts of content and will benefit from parallelization of these services on an as needed basis.

4. Create production and development templates for configuring the running software, caching services, and monitoring and management services.

Current production configuration at Connexions is done on an as-needed basis by highly skilled IT staff. Proper configuration is critical for balancing workload and appropriately monitoring the system, as well as for proper installation for development and testing.

Templates (persistent configuration information) to create configurable Connexions and Rhaptos instances (or components) used in conjunction with virtualization can be used to start production, development, testing, or demonstration instances.

Detailed Tasks

1. Simplify installation, development, and testing environment

- **Develop test beds of data** and mechanisms for generating test data

- **Create testbeds of data** - (modules, collections, lenses, site content such as the front page and help, accounts) including:
 - an empty repository (no modules, lenses, collections, or user accounts with a minimum set of site content),
 - a test repository with content, accounts, and workgroups, that helps to test out new features. The repository should include all our test modules and help modules and collections, and help lenses along with accounts and workgroups. Our QA manager has a list of modules and collections that use interesting features and some from those should be included also.
- **Create the ability to import and export modules, collections, lenses, and accounts between Rhaptos and Connexions instances**, so that newly developed test data is easy to migrate between instances, so that demonstration data is easy to migrate in and out of demonstration instances, and potentially for use during migration of data during code upgrades. Exporting a collection should export all the component modules also. The import/export does not have to try and handle dependency checking and can expect that for module and collection import accounts for all roles (author, copyright holder, maintainer, editor, translator) will exist on the instance for import. It would be helpful to have the ability to ask on export for a listing of needed accounts. Account export/import does not need to export/import unpublished work.
- **Simplify installation and complete clean separation of Rhaptos and Connexions code.**
 - **Install Issues List** -- (See list: Issues: https://trac.rhaptos.org/trac/rhaptos/wiki/Virtualization_and_Installation/Inception/Issues)
 - **Simplify component installation**
 - Reduce dependence on specific and or customized releases of companion open-source libraries (See list: [Issues](#))
 - Convert to using declarative rather than procedural methods for installation and upgrade so that these can be applied systematically.
 - **Separate Rhaptos and Connexions code clearly and cleanly** (See list: [Issues](#))
 - Merge current Rhaptos installation branch code into trunk
 - Ensure proper styling throughout for Rhaptos only instances, eliminating any Connexions styling that may persist in some products.
- **Testing:** Develop unit and regression tests and develop a nightly build and regression test system for trunk or a specified branch
 - **Complete test framework and nightly build system** based on bitten and FunkLoad.
 - Prior build system work -- <https://trac.rhaptos.org/trac/rhaptos/build> -- documented here -- https://trac.rhaptos.org/trac/rhaptos/wiki/Virtualization_and_Installation/Deliverables/BuildInstallTestArchitecture
 - **Create regression functionality tests** for common operations and external interfaces and API's using FunkLoad and/or Selenium:
 - Publish a module, publish a collection, create a lens and add published content to the lens including tags and comments, create a user account, create a workgroup, import a Word document, import a LaTeX document, browse and search content, view the home page and help pages, checkout a module, checkout a collection, perform an OpenSearch, ask for metadata using the published OAI-PMH (<http://rhaptos.org/docs/apis/metadataharvesting>), ask for Atom

and RSS feeds (<https://trac.rhaptos.org/trac/rhaptos/wiki/RSSFeeds>), view various test modules and collections, view lenses, request (URL-based) module source, module PDFs, collection PDFs and collection multimedia zip files. Connexions will provide access to any tests already developed as guides and starting content.

- **Produce developer API and developer guide** to get new Rhaptos developers started. This should be a compilation of the notes taken during the install and test development process and should be included on the Rhaptos development tracking site (trac). It should not be a major post-facto undertaking.

2. Virtualization

Background: Connexions hired a consultant to develop a virtualized version of Rhaptos and Connexions in the summer of 2008 and the progress was encouraging. Two related factors limited the effectiveness of the project, however. First, instances of Rhaptos/Connexions currently load a complete copy of the Connexions data (modules, collections, lenses, accounts, workspaces, workgroups) in order to operate properly. Loading a complete dataset is inefficient for current development and testing, but was made even worse by slow transfer times between S3 (Amazon Cloud Storage) and EC2 (Amazon cloud computing). The data took 2 hours to load making the system impractical to iteratively develop, and impractical to use for creating development and testing instances, or demonstration instances. This proposal calls for addressing the dataset problem in the "Simplify development and installation" section, and Amazon introduced permanent disk images just after our virtualization milestone, which makes startup from various saved datasets much faster. During Hurricane Ike (Sept. 08), cnx.org ran on EC2 for approximately 24 hours, demonstrating the value of the initial work on virtualizing the system.

Plans from prior (summer 2008) work on installation and virtualization

- [https://trac.Rhaptos.org/trac/Rhaptos/milestone/Virtualization and Installation](https://trac.Rhaptos.org/trac/Rhaptos/milestone/Virtualization+and+Installation)

Work needed to complete the virtualization system

- Finish virtualized architecture using Amazon EC2 and S3
- Produce small, medium and large Amazon Machine Images (AMI's) for startup, ready from the nightly build system. These should be available for startup in the Amazon cloud. An Amazon Web Services test account and budget will be provided.
- Store sample datasets on S3-- empty, test, and complete cnx.org set from Connexions nightly backups.
- Produce instructions to start a Connexions sandbox or demonstration instance using a selected dataset. It should take no more than 15 minutes to get a sandbox or demonstration instance running
- Produce instructions and scripts to start a development instance. It should take no more than 15 minutes to get a development instance running from trunk or a specified code branch with a specified data repository.
- Produce Virtualization Documentation
 - Where to find the published AMI's
 - How to build an AMI for Rhaptos or Connexions.
 - How to update running instances with a fresh data set and backup the dataset from running instances.

3. Duplicate import pathways for scaling in a virtualized environment

Scale the importers in order to increase the ability to import many documents at once.

- **Create a just-in-time deployment service that can be used by the various import/translation pathways** in Connexions that can queue up tasks and start and stop additional instances of services to adjust to the demand. The service should be designed so that it can be used for future queueing and resource allocation tasks such as LaTeX import queueing or PDF generation queueing. May consider using queuing services provided by Amazon's virtualization service. If used, the connection should be abstracted so that other services could be substituted in the future.
- **Word/OpenOffice:** Develop a queue for Word and Open Office import tasks and a programmatic way to start and stop Open Office server instances that uses the just-in-time deployment service.

4. Production and development templates and guides

Produce templates (persistent configuration information) and instructions to create Connexions and Rhaptos instances (or components) used in conjunction with virtualization to start production, development, testing, or demonstration instances. If practical, produce a web service or set of scripts that functions as an automated configuration and launch system. Determining optimum parameters are not a part of the RFP and Connexions staff will provide initial configurations that should be modeled.

- Create initial configuration templates for
 - Development instances for Rhaptos and Connexions (customized Rhaptos), with all services running on a single instance
 - Testing instances
 - Demonstration instances that need a convenient URL.
 - Large, heavily used site -- The following shows the basic production architecture used in Connexions: <http://rhaptos.org/docs/sysadmindocs/production-schematic.png>
 - Small, lightly used site (caching set up and a front and back end)
 - Low-bandwidth template with local caching and/or local frontends and remote backends.
- Produce the following documentation
 - Deployment guide - how to use the templates to create a fully functional system.
 - Production monitoring guide - How to configure nagios and munin to monitor a production site.
 - Backup and Restore guide - How to set up data backups to run periodically, and how to restore (and test and develop) from data backups.
- If time and budget allow, produce an automated system for configuring and launching Rhaptos and Connexions instances.

Deliverables

1. Simplify installation, development and testing environment.

- A. **Data testbeds:**
 - An empty repository and a test repository.
 - Instructions on starting an instance using these repositories.
- B. **Import and Export:**
 - API for importing and exporting modules, collections, lenses, and accounts.
 - Scripts for exporting and importing content between specified Rhaptos and Connexions instances.
- C. **Installing:**
 - Instructions and scripts for building Connexions and Rhaptos instances of the code.
 - Demonstration that the process results in working instances.
- D. **Testing:**
 - Nightly build system that creates Rhaptos and Connexions instances from trunk and runs regression tests.
 - Regression tests for common operations and external interfaces and API's (listed under "Details" section).
- E. **Guides:**
 - Developer API and Developer Guide on trac for new Rhaptos developers.

2. Virtualize the Connexions Platform

- A. Small, medium and large Amazon Machine Instances (AMI's) for startup, ready from the nightly build system. These should be available for startup in the Amazon cloud.
- B. Sample datasets stored on S3-- empty, test, and complete cnx.org set from Connexions nightly backups.
- C. Instructions to start a Connexions sandbox or demonstration instance using a selected dataset.
- D. Instructions and scripts to start a development instance from trunk or a specified branch.
- E. Documentation - location of AMI's and datasets, how to run them and create new ones.

3. Duplicate import pathways for scaling in a virtualized environment.

- A. API for a just-in-time deployment option for the various import/translation pathways in Connexions that can queue up import/export tasks and start and stop additional instances of import services to adjust to the demand.
- B. A queueing system for Word and Open Office import tasks that automatically adjust to import load by increasing the number of OpenOffice servers available for import.

4. Create production and development templates for configuring the running software, caching services, and monitoring and management services.

- A. A set of templates (persistent configuration information) for configuring development, testing, demonstration, large production, small production, and low-bandwidth sites.
- B. Deployment guide using the templates to start various flavors of Rhaptos and Connexions systems.
- C. Production monitoring guide
- D. Backup and Restore guide
- E. If time and budget allow: an automated configuration and launching system that uses the templates to launch fully-functioning, and configured Rhaptos and Connexions systems.

Background Information about Connexions and Rhaptos Development Useful for Preparing a Proposal

Basic development support

- Public SVN repository
 - browsable – <https://trac.rhaptos.org/trac/rhaptos/browser> ,
 - svn client -- <https://software.cnx.rice.edu/svn/>)
- Public access to Trac (<https://trac.rhaptos.org>) (Bugs, Wiki for design docs and discussions, Milestones, Source browsing) Patches can be submitted with tickets. SVN commit access is available for trusted partners and developers.
- Developer documentation - <http://rhaptos.org>

Connexions architectural documentation

- <http://rhaptos.org/docs/architecture/archWhitePaper.odt>
- <http://rhaptos.org/docs/architecture/archWhitePaper.pdf>

Current installation instructions

- <http://Rhaptos.org/docs/installation> (not completely up-to-date)
- See also [Installation Documentation Issues \(https://trac.rhaptos.org/trac/rhaptos/wiki/Virtualization and Installation/Inception/Issues/InstallDocumentationIssues\)](https://trac.rhaptos.org/trac/rhaptos/wiki/Virtualization%20and%20Installation/Inception/Issues/InstallDocumentationIssues)

Plans from prior work on installation and virtualization

- [https://trac.Rhaptos.org/trac/Rhaptos/milestone/Virtualization and Installation](https://trac.Rhaptos.org/trac/Rhaptos/milestone/Virtualization%20and%20Installation)

Connexions architectural components

- OS
 - Debian Linux
- Repository and Authoring
 - Plone/Zope/Zope Object Database (ZODB)
 - PostgreSQL
- Production configuration
 - Squid
- Importers/Translators
 - OpenOffice Server, Tralics
- Print System
 - LaTeX system -- TeXLive
 - Python Image Library (PIL)
 - MathML stylesheets
- Other
 - Javascript libraries - ExtJS and Prototype
 - XSLT and XML processing libraries
 - Sarissa, libxml2, libxslt
 - unFocus History Keeper in the Browse interface
- Production

- Squid -- load balancing and caching
 - Monitoring - Nagios, Munin, Jabber
 - Backup and restore procedure -- custom scripts
- Development
 - Trac
 - Bitten, FunkLoad
 - Selenium

Proposal Submission Information

Assumptions:

- All work will be done in Connexions' svn repository on branches and all documentation will be on the Connexions trac system.
- An Amazon Web Services account will be provided with a monthly budget for testing and development of the virtualization system.
- Reasonable travel costs will be reimbursed for face-to-face meetings with Connexions staff to conduct code and functionality walk-throughs periodically.

Required proposal format: The proposal must contain a (1) Technical section, (2) Qualifications Section, (3) a Time-Cost section,

Technical section: Include a description of how the tasks will be accomplished, time-lines, projected required personnel, and proposed milestones with deliverables

Qualifications section: Describe the qualifications of the firm, the lead architect and manager, and proposed team members. Include references and links to completed projects.

Time-Cost section: Provide a firm estimate of the fees to be charged and expenses incurred, along with the time to complete the milestones and entire project.

Partial bids will be accepted: Although we strongly encourage bids on the full proposal, we will accept bids on one or more of the high level tasks also.

Submission deadline: May 1st, 2009

Award date: May 21st, 2009

Submit Proposal To:

Katherine Fletcher
Connexions, Rice University
PO Box 1892, MS 375
Houston, TX 77005

For additional information or clarification contact:

Katherine Fletcher
kef@rice.edu

Rubric for Evaluating Submissions

Metric Description

1. Estimate credibility

To predict the accuracy of the time/cost estimates and the bid. This will be influenced by the following factors:

A. Overall corporate experience

A.2 References

A.3. Experience of the lead architect/manager

A.4. Experience procuring and managing teams with diverse skill sets, writing detailed specifications, and managing teams to complete projects on time and in budget.

A.5. Experience of the development team(s)

A.6. Experience of management

B. Specific experience

B.1 Virtualization

B.2 Production quality software and streamlining the development process

B.3 Open-source software

B.4 Plone/Zope/XML transforms

2. Proposal Detail

Measure of the detail present in the proposal and the quality of questions asked during the bidding process.

2.1. Time estimates on functionality

2.2. Milestones

2.3. Explanations

2.4. Visuals

2.5. Questions during bid process

3. Time estimate to completion (shorter is better)

4. Development price (lower is better)