

# **VTLS-VITAL Final Research Report**

**13 March 2006**

## **An IAC Initiative for the Creation of Common Information Infrastructure for Digital Collections to Support Teaching, Learning and Research**

### ***Executive Summary***

Fundamental changes in the means by which scholarship is created, packaged and distributed over networks have precipitated the need to re-engineer core library services that manage the life cycle of scholarly information. Content housed in digital formats challenges research libraries to redefine workflows and procedures used to catalog, search, access, build and preserve electronic collections. The Integrated Access Council (IAC) plans, advises, and commissions work groups on emerging digital library initiatives including the life cycle management of digital content, integrated access to collections, and learning technologies. According to IAC research on digital repositories [http://www.library.yale.edu/iac/documents/DR\\_Review\\_final\\_27Sept05.pdf](http://www.library.yale.edu/iac/documents/DR_Review_final_27Sept05.pdf), repositories can serve as (1) homes for digital collections (2) infrastructure for digital preservation (3) safe harbors for faculty output, and (4) mechanisms for supporting scholarly publishing, open access and institutional branding. To achieve the goal of resource integration the outputs from scholarly content flows i.e. faculty and student output need to be channeled into a reservoir or digital repository system that can provide common services to link this content to other library services and to university collections. In addition, repository content would take on added value if it were passable to external resources or functions like e-learning systems. This capability could motivate faculty and researchers to deposit learning objects produced in virtual learning spaces and personal electronic collections into a centralized repository system. Correspondingly, a common infrastructure of repository services and systems would help the Library to achieve its goal of content and service integration.

In July 2005 IAC initiated the VTLS/VITAL project to investigate digital repository software that supports new modalities for teaching, learning and research and demonstrates the potential to provide common information infrastructure for digital collections. IAC recommended that the Yale Library prototype a common framework for digital collections based upon open standards and technology that are consistent with Information Technology Services infrastructure to support e-learning and research. The open source digital library software Fedora (Flexible Extensible Digital Object Repository Architecture) is specifically designed to create infrastructure for digital collections that interface with the Service Oriented Architecture of the Sakai E-Learning System, recently adopted by ITS to support instruction and research. With the assistance

of the commercial vendor, VTLS, and their product, VITAL (a digital library application based upon Fedora), IAC proposed to establish a prototype digital repository environment for the Library that would accommodate a wide variety of digital collections, regardless of format. VITAL was selected because it is built on the Fedora framework but promised advanced functionality with minimal initial development required by the Library. In addition, recognizing that Yale would require significant customization of any repository software, VTLS was seen as an attractive potential development partner while the Fedora framework would allow the Library to develop additional library partnerships and contribute to the Fedora consortium. Finally, IAC supports the commercial/open source model upon which VITAL is built.

The strategic goals of the IAC VTLS/VITAL research project were threefold.

The first was to deploy VITAL and the FEDORA repository environment to test it as a potential home for collections within the Yale libraries. Homes for digital collections can serve as a base from which (1) interdisciplinary collaboration may be supported and (2) the potential reuse of digital content to create learning objects for teaching, learning and research can occur.

The second goal was to explore the possible infrastructure for digital preservation of the Library's electronic collections. The Library's department of Manuscripts and Archives committed resources to help investigate this dimension of the research.

The third goal involved helping librarians better understand how other services in the digital environment interface with a digital repository. As an illustrated example, collection builders articulated the technical requirements for migrating and maintaining their collections in a digital repository environment. Participants identified the need for additional technical training for non-technical librarians in the areas of web-services and XML technologies.

As a formal component of the project, XX sets of collection builders generated use cases outlining user requirements for accessing and interacting with each particular set of collection objects. For the library these use cases serve as one source of input to help establish base line requirements for other digital repository systems that will be evaluated in the future. The use cases aided the collection builders in their assessments of VITAL system functionality.<sup>1</sup>

The VTLS/VITAL project participants experienced barriers along the way related to software, documentation and hardware problems. Yale's eagerness to begin the research project on development code that had incomplete documentation contributed to slow

---

<sup>1</sup> The use case approach indirectly assessed VITAL functionality. The environment (configuration of a VITAL component) and the context (use case) within which a VITAL function was deployed influenced the success or failure of the function to meet the use case goal. An observation is reported that an insufficient time was allocated to understanding the failure of a function within the context of the use case. This could have the unintended consequence of concluding that a VITAL function or feature does not operate correctly. The potential for false positives in the project were high do to time constraints, limited vendor support when problems were encountered, minimal training and poor documentation. Is it a useful question to consider if these factors lead to a premature and bias assessment of VITAL?

progress of the collection builders in the early period of the research. In addition a server disk failure which did not have backups also diminished the time participants could work on all aspects of the projects.

The VTLS/VITAL research findings must be understood within the context that there is a mismatch between Yale Use Cases serving as collection exemplars and the current state of digital repository software. Lagoze et al (November 2005)<sup>2</sup> characterized the state of digital repository software as adolescent in nature. That is to say they are works in progress like the adolescent that struggles to mature by passing through periods of incompleteness, disappointment and frustration. If we view digital repository software, as a work in progress, than we understand the source of the frustration and disappointment we had with VITAL. No product in our market either from a commercial vendor or an open source offering could meet all of the idiosyncratic and complex requirements outlined in our use cases. VITAL must be evaluated within this context as an evolving product that now meets some basic Use Case requirement and has the potential to mature like the adolescent, into an enterprise system if given additional time and resources. Architecturally, the product has great potential.

The results of the use cases decisively indicate that the VITAL software, out-of-the-box, did not satisfy the use case requirements established by the collection builders and their advisors. The architecture of the product makes clear how and where changes or additions may be made. Who best to make the changes and when remains to be determined. Project participants found significant gaps in all major VITAL features including:

1. VITAL Manager Client,
2. VITAL Batch Ingest Tool,
3. VITAL Access Portal and
4. VITAL Workflow tool called VALET.

Consequently, the project group does not recommend that the Library deploy VITAL 2.0 as an interface for managing its digital collections without additional development. However, the collection builders differentiated VITAL from its underlying software application, FEDORA, and its Service Oriented Architecture (SoA). Collection builders and some Advisory and Working Group members recognize Fedora's potential to meet their highly differentiated and sophisticated collection building requirements. These conclusions suggest three alternative scenarios that could advance the Library's investigation of repository systems to manage and preserve digital collections:

1. The Library could engage in a development partnership with VTLS to build additional VITAL functionality that would meet the requirements outlined in the

---

<sup>2</sup> Lagoze et al 'What is a Digital Library Anymore, Anyway? Beyond Search and Access in the NSDL',DLIB Magazine Novemver 2005 Volume 11 Number 11

use cases. Alternatively the Library could purchase consulting services from the Harris Corporation that has expertise in writing Fedora applications.

2. The Library could launch a follow-on pilot of native Fedora to build a series of specialized digital collections, providing internal library support for programming and development or exploring consortial development models.

The library could investigate other open source and commercial software solutions for digital repository software. Together, the VITAL research use cases, our incomplete understanding of user requirements and the adolescent nature of digital repository systems suggests that a one size fit all solution cannot be fashioned now, for the Library, as common information infrastructure. A mosaic of repository services may be needed to provide a collection building infrastructure.

The project leaders acknowledged the efforts of Carl Grant President of VTLS and the organization for its willingness to be a partner in this research investigation. By entering into this collaboration VTLS demonstrated its commitment to serve Libraries and the cultural heritage community. The resources that VTLS extended to Yale during the research were generous and supportive. In addition, VTLS' commitment to develop software that is compatible with an open source offering like Fedora differentiate and distinguishes VTLS from other vendors in this market space. In this context, VTLS serves as a good example where commercial vendors can become contributors to open source projects like Fedora that benefit our community while still being able to profit by their work.

## ***Introduction to the IAC VTLS/VITAL Project***

### **The Context:**

The past decade has seen fundamental changes in the format and lifecycle workflows of scholarly information that are used to support daily activity—teaching, learning and research—at Yale and its Libraries. New ways to publish, discover and access scholarly communication have broadened the scope and nature of core library services like cataloging, reference and preservation. These services must now be more tightly woven into external instructional systems. To support faculty and students in the digital world, the Library needs new infrastructure for digital collections that not only integrates content but extends the usefulness of digital content to facilitate interdisciplinary collaboration and the creation of digital learning objects that are polymorphic and therefore re-usable” in multiple pedagogical contexts. The Library must explore how it will adopt and integrate new infrastructure to support its role in the evolving information ecology of the academy and of the wider society.

### **The Concept:**

According to IAC research on institutional repositories [http://www.library.yale.edu/iac/documents/DR\\_Review\\_final\\_27Sept05.pdf](http://www.library.yale.edu/iac/documents/DR_Review_final_27Sept05.pdf), digital repositories can serve as (1) homes for digital collections (2) infrastructure for digital preservation (3) safe harbors for faculty output, and (4) mechanisms for supporting scholarly publishing, open access and institutional branding. To achieve the goal of resource integration the outputs from scholarly content flows i.e. faculty and student output need to be channeled into a reservoir or digital repository system that can provide common services to link this content to other library services and to university collections. In addition, repository content would take on added value if it were passable to external resources or functions like e-learning systems. This capability could motivate faculty and researchers to deposit learning objects produced in virtual learning spaces and personal electronic collections into a centralized repository system. Correspondingly, a common infrastructure of repository services and systems would help the Library to achieve its goal of content integration.

In July of 2005 the Integrated Access Council (IAC) initiated the VTLS/VITAL project to investigate digital repository software that supports new modalities for teaching, learning and research and demonstrates the potential to provide common information infrastructure for digital collections. IAC recommended that the Yale Library prototype a common framework for digital collections based upon open standards and technology that are consistent with Information Technology Services infrastructure to support e-learning and research. The open source digital library software Fedora (Flexible Extensible Digital Object Repository Architecture) is specifically designed to create infrastructure for digital collections that interfaces with the Service Oriented Architecture of the Sakai E-Learning System, recently adopted by ITS to support instruction and research.

The ability to share services or functions across applications differentiates Fedora from DSpace and is the principal reason for rejecting DSpace and instead evaluating Fedora as

common infrastructure for Yale's digital collections. Any SoA application that speaks Web Services can communicate and share data across applications. Therefore the Fedora application could easily serve as a content pipe for Sakai. In the Sakai environment, digital objects can be processed and re-used to build new learning objects for a particular classroom exercise. This potential functionality made the selection of Fedora a timely, justifiable and strategic approach that was consistent with the aforementioned IAC goals. With the assistance of the commercial vendor, VTLS, and their product, VITAL (a digital library application based upon Fedora), IAC proposed to establish a prototype digital repository environment for the Library that would accommodate a wide variety of digital collections, regardless of format. Under this repository system, project participants can contribute to centrally supported services or build independent collections whose descriptive records can later be federated to facilitate cross-collection searching, access and tool-building.

### **The Process:**

Selecting the appropriate technology is but one component of developing infrastructure to support digital collection building. Robust system and workflow development depends upon the interaction of library culture, processes, and technological functionality. To conduct the VTLS/VITAL project, IAC organizers invited participants from a wide variety of departments within the Library and from other departments on campus. Participants were organized into two groups: a Core Working Group and an Advisory Group. Members of the Core Working Group built collections representing different academic domains at Yale such as medicine, the humanities and in the field of preservation. Advisory group members acted as consultant and assume different roles such as subject specialists and technologist and rendered usability assessments of the resulting repository collections.

Project organizers utilized a group workspace in Sakai to promote collaboration among group members and to disseminate information about the project such as progress reports from the project manager, collection builders' experiences with implementing VTLS/VITAL functions and to document software bugs or other problems. The Sakai site also featured a formal Use Case tool. A use case is a formal specification that describes a user's interaction with a system to achieve a desired goal, i.e. the building of a collection. Use cases are traditionally used in software development to specify discrete system features that may be created and then tested by software users or machine processes to confirm that the feature works properly. In the context of the VITAL project, use cases describe user requirements for the functionality of the repository software and reflect the needs of diverse user communities. Collection builders developed use cases relevant to individual collections, while project organizers developed a strategic use case to represent the organizational objective of the project.

The strategic goal of the project was to determine how efficiently VITAL could be used to ingest and manage Yale digital collections using Fedora as an underlying repository framework. Secondly, the project assessed the VITAL/FEDORA platform based on its capacity to support long-term digital preservation. Seven scenarios representing seven

different collection building modalities (archives, social science datasets, finding aids, image, audio/video objects from the Beinecke Rare Book and Manuscript Library, medical images from the Medical Library, Unicode from Middle-Eastern collections and the Yale Indian Papers project, a scholarly organization affiliated with the University) were represented in the project. The scenarios, instantiated into use cases, were designed to test how well VITAL handles different content types, storage configurations (Fedora-Internal and Fedora-External) and ingest modes or workflows. While these scenarios tested features of the VITAL product, they also lend insight into how this product can be used to provide collection and item (object) views of data, cross collection searching and federation through the OAI protocol.

Collection builders documented key user requirements for searching, accessing and manipulating collections in their domain into the formal use case templates. In addition to developing use cases centered on user experience, participants also conceptualized system use cases to describe the expected outcomes of machine processes such as indexing, record ingest, and batch processing. System use cases allowed collection builders to identify gaps in components of VTLS/VITAL software that were used to create workflows, ingest content, index content, search collections and brand collection interfaces. The use cases provided a framework against which project participants could systematically assess the success/failure of their collection building work and analyze components of the VITAL system functionality.

Appendix C contains all of the use cases generated by the collection builders with support or input from their advisors. The following section contains the use case reports of the collection builders.

## **Report on the Collection Building Experience**

(See Appendix D)

### **Lessons Learned From Use Case Reports:**

Among the factors driving the development of digital repositories are the need for infrastructure to manage the long-term curation of digital collections, the increasing levels of at-risk faculty resources, the need for alternative scholarly publishing models, and the need to repurpose and re-contextualize digital assets. Trends in scholarly communication and instructional technology suggest that digital repositories are becoming increasingly important components in the academy. At Yale University, there is pressing need to find homes for digital collections proliferating on campus and for which the Library has been given or must take stewardship responsibility for providing access and ensuring long-term preservation.<sup>3</sup>

The IAC VITAL/Project represents an important step in evaluating the role of repository environments at the Yale libraries. Collection builders amassed significant information about the software requirements that are needed to provide homes for digital collections and archival systems to preserve digital content over the long run. Without this fundamental infrastructure, digital scholarly output from new Yale initiatives such as Center for Globalization and Yale and the World is in jeopardy of being under utilized or made inaccessible through technological obsolescence or media decay. What then are the lessons learned from the VITAL project? How can these lessons deepen our understanding of: (1) building homes to host complex digital collections (2) creating systems to preserving the digital content for re-use by successive generations of scholars and (3) assessing a vendor's ability to help the Library create and extend open source digital repository software.

The use cases developed by the collection builders in conjunction with their advisors demonstrated that there are no typical digital collections with simple requirements. Collection complexity comes from the extension of the basic building blocks (ingestion of diverse formats either simple or complex, ingestion of simple and complex metadata, sophisticated indexing requirements for metadata search and content retrieval and personalized presentation of content through customization of the user interface) of any digital collection stored in repository software.

### **Ingest Functionality:**

The VITAL software offered two modalities of *ingest*: a batch process and a GUI interface as part of the VITAL Manager Client. The VITAL use cases were very effective in demonstrating the level of maturity of these functions. The collection builders learned

---

<sup>3</sup> See the IAC report “**Review of Digital Repositories**”  
[http://www.library.yale.edu/iac/documents/DR\\_Review\\_final\\_27Sept05.pdf](http://www.library.yale.edu/iac/documents/DR_Review_final_27Sept05.pdf)

very quickly that ingest functions of the VITAL Client Manager such as “Drag and Drop” and auto generation of metadata worked well, but were only developed to the point of effectively ingesting simple objects and creating basic metadata. The lessons learned here were that:

- 1. Yale’s collections require more sophisticated tools to ingest content from a broad spectrum of formats.**
- 2. Ingestion of text in non-Western languages such as Unicode-encoded Arabic text require special XSL style sheets that VTLIS does not provide in their default system.**
- 3. Ingest tools and functions of VITAL need to be extended to handle more than simple Dublin Core metadata and**
- 4. Knowledge of the XML stack like XPATH and XSLT is needed to effectively work with, i.e. extend VITAL tools for batch ingest.**

For example, ingestion with known formats proved possible with either modality. In the Finding Aid use case, EAD/XML files ingested successfully into VITAL. Similarly the Medical Images use case showed that the VITAL batch could readily process TIFF images. However, both the VITAL batch and the VITAL Client Manager could not be used easily by collection builders loading complex objects encoded in UNICODE or objects with MIME types unknown to VITAL like proprietary statistical datasets used in the Social Science Data Sets uses cases. That the underlying architecture of VITAL, FEDORA can ingest diverse formats suggests that it is a viable option for repository software in Yale’s overall information architecture plan. Yet the basic scope of VITAL’s ingest routines suggests that vendors like VTLIS need development partners to create ingest processes for specialized data formats that are typical in Yale’s digital collections. This represents a potential collaboration opportunity for VTLIS and the Yale Library.

VITAL ingest functions for metadata had the same limitations. All the use cases demonstrated that simple Dublin Core Metadata records could be created for an object on ingest. Yet metadata descriptions for Yale’s collections extend beyond simple Dublin Core. The Medical Images use case demonstrated that the VITAL batch configuration files could not be customized to create, upon ingest, qualified Dublin core records for associated TIFF images. Through a workaround in the VITAL Manager Client, this became possible on an image by image basis but impractical, if not impossible, on a production scale that involved thousands of digital objects. For the Medical Library, adopting VITAL as is would mean either abandoning their qualified Dublin Core metadata (at significant cost) or contracting with VTLIS to provide the added functionality. In contrast, the Greenstone Digital Library software, in production at the Medical Library, not only offers this as a basic function but also provides for the extension of this and other predefined metadata sets. Despite these limitations, no collection builder failed to add content or metadata to their repository instance. However, the Unicode Use Case that involved Arabic content, demonstrated that the VITAL batch was primarily designed to ingest text in western languages. (To be fair to VTLIS it is important to note that this is not unlike any other software repository offering in this market space.)

Due to inadequate documentation, the collection builder concluded that “it was not clear where to ingest the text for indexing and searching, when this text is Arabic. Since the OCR step in the digitization workflow for project AMEEL may produce files with Arabic text in either format ... it is important that these files can be part of an ingest procedure.” Despite the difficulty processing Arabic text, this collection builder appreciated the VITAL batch function that auto-generates searchable text as output from ingested PDF files.

### **Indexing and Search Functionality**

The collection builders used the VITAL Access Portal, application, and administrative functions to create additional indexes and to customize their collections for search and presentation of content. Like the VITAL Manager Client, the VITAL Access Portal performed well in the context of simple tasks: to search, retrieve and to present a simple object. However, when use case requirements went beyond basic functionality, collection builders encountered gaps and “bugs” in the software. Lesson learned from our use case requirements for VITAL Access Portal functions were similar to those outlined for the batch function:

- (1) Search, indexing, navigation and presentation of content for a typical Yale collection are highly complex and idiosyncratic. Default VITAL tools for these functions were inadequate but reflected the adolescent nature of digital repository software .**
- (2) Inadequate or incomplete documentation constrained the ability of collection builders to use software tools to customize a user interface for a collection.**
- (3) Again as noted above, inexperience or a lack of training in XML technologies limited collection builder’s capability to take full advantage of VITAL’s XML based tools for search, indexing and presentation.**

Following are some illustrative examples that highlight these lessons. Use cases on Scanned Referenced Publications and the Finding Aids demonstrated important gaps in the ability of a user to navigate search results. To be more effective, the VITAL document viewer should provide a means to search and navigate within a PDF file. In the words of the collection builder:

*“..the search result for a text string in a 50 page pdf file points simply to that pdf file. The user must then download the file and perform the search again within READER to actually get to the result. This not only feels like unnecessary duplication of effort, but adds a significant enough amount of time to each search, that many users would undoubtedly be limited in their ability to fully search a resource”.*

The Finding Aids use case demonstrated that VITAL’s out-of-the-box solution for a Finding Aids repository would serve only elementary level collections where EAD to HTML presentation was satisfactory. The current release of VITAL has no way to

segment or arrange different collections of finding aids. This is a limitation that is likely to be resolved in the next release of VITAL. Yale Finding Aids come from different sources or academic domains. The ability to delineate individual collections is essential. In addition, the VITAL default style sheets do not fully exploit sophisticated navigation within a finding aid based upon xml schema of a finding aid.

While navigation of PDF and Finding Aids files were problematic, the use cases (Finding Aids and Medical Images) showed that the ability to produce custom indexes for a collection was possible with VITAL tools, although complicated. The medical images use case successfully indexed, with some effort, qualified Dublin core metadata fields. Quoting from the collection builder ...”After some project and error, due to my own lack of knowledge of the structure of XPATHS and VTLS’ lack of documentation... I was able to add my metadata to existing indexes...” It is important and instructive to note that adding Indexes to the Finding Aids use case was reported as less problematic. This may be the result of the level of development of other Fedora based Finding Aid applications that VTLS could incorporate into VITAL. If true, this would be a good example how the underlying architecture of VITAL can take advantage of the concept of re-usable modules. It also provides a good example as to how Fedora can expedite development. As written by the collection builder ...”The VITAL access portal Administrative Interface was used to ... create new indexes on specific EAD tags... With changes to the configuration of the Access Portal ... these indexes were made available as ‘Repository’ and ‘Scope and Content’ in the dropdown menus under ‘Advanced Search’”.

### **Presentation and Navigation of Search Results:**

The VITAL Access Portal – application functions were used by collection builders to (1) customize the look and feel of the user interface (search and result-presentation pages) and (2) for image navigation with VITAL’s Image and Document navigation tools. The control over the appearance of html pages in VITAL was found to be complicated. A UNIX editor like “VI” was needed to change a file that controlled text on an html page. Another file was used to control the presentation of the text i.e. the colors, fonts, pictures and layout. While technically appropriate, modification of both files required the understanding of reading and modifying an XML markup language like cascading style sheets. Consequently only a few collection builders made changes to these files to customize the look and feel of their html pages. The important take away here is that the management of page appearance or presentation assumes that function is under the control of a systems administrator not a librarian. This is not an ideal approach and a GUI interface into this function would allow either a librarian or systems person to control the look and feel of collection pages. In addition navigation for images and documents had serious concerns for our collection builders.<sup>4</sup>

VITAL’s image navigator is designed to view high resolution images. As with other VITAL features basic functionality of the image navigator was acceptable. That is to say

---

<sup>4</sup> It also should be noted that the access portal did not have the functionality to produce a preview of TIFF images which was a significant problem for the medical images collection.

collection builders successfully used the tool to view and manipulate simple MrSID and JPEG2000 images. However, the image navigator could not be used to view a multi-part high resolution document. In addition, collection builders had no control over the order of presentation of images. The document navigator had similar issues. The function could handle a simple presentation of a multi-page document as long as it was in a pdf or jpg format. Collection builders found that document navigator could not display multi-page images in TIFF or JPEG200 formats. Like the high resolution image viewer the document navigator randomly defined the order in which images were presented. Therefore a multi-page pdf file of sequential text pages was often presented out of order. To a reader this is confusing and greatly constrains the usefulness of the function.

**Interoperability with other systems:**

The underlying architecture of VITAL (service oriented architecture) promotes the applications interoperability with other systems. This feature of the system was not well explored through the collection builders' use cases. This is unfortunate since many of these features are very useful components of a digital repository system for post-processing metadata search and discovery services. For example, VITAL Client Manager or VITAL batch process automatically produces an OAI record that is potentially accessible to an external OAI harvester. Important metadata services like an information network overlay uses the OAI protocol to harvest and then post process metadata records. For search purposes VITAL can establish a Z39.50 connection to Orbis. VITAL also contains a built in handle server that can be used to create and manage persistent identifies for VITAL objects. A handle server is a fundamental component of any digital repository service on which any number of services (discovery search, preservation) are dependent. It is of importance to note that VTLS' handle module was well received by our community and as an open source product it is being integrated in Fez, a repository interface for Fedora sponsored by the Australian Partnership for Sustainable Repositories. Finally VITAL also offered the SRU/SRW as two means to search for content on the network. SRU allows searching the network based upon URLs and SRW allows searching the network through a web service. In short these were lost opportunities for the Library to learn more about VITAL features which connect their system to content hosted on the greater web. Any experience with these functions would have advanced our understanding how these services could be contextualized for the Yale environment.

## **Conclusions:**

The IAC report entitled “Review of Digital Repositories” recommended that the Library setup a test digital repository using VITAL/Fedora. In the fall of 2005, the VTLS/VITAL project was launched in response to this suggestion. Below are the recommendations from the project about the efficacy of VTLS/VITAL to serve as a foundation for digital repository services at Yale. Lessons learned from the project have deepened our understanding of the complexity inherent in an Information Architecture plan where the digital repository serves as a major component for digital content management. In addition, the deployment of the use case tool effectively engaged users and resulted in an initial articulation of requirements for digital repository functionality at Yale.

Establishing a trusted digital repository and providing for long-term preservation are key requirements in the design of Yale library’s information architecture. These requirements informed the scope of the VITAL/FEDORA project. The aggregation of use cases created by the collection builders and their advisors clearly demonstrated that:

1. **VITAL in its current form is not a sufficient or complete tool for complex digital repository development at Yale but has the potential to be with additional development.**
  - VTLS/VITAL could not meet the objectives or requirements of the strategic use case. As an out-of-the box-solution, VITAL could not satisfy, the entirety of the use case requirements for most domains, and failed in some important respects in every domain examined.
  
2. **FEDORA, in native format, has promise as an architecture to host Yale’s digital collections and to serve as a preservation system.**
  - While VTLS/VITAL does not present a viable solution for complex repositories, use case documentation and the sentiments of collection builders during the process indicate that that Fedora’s service oriented architecture could support some of Yale digital repository and preservation requirements.

However, additional research and training<sup>5</sup> is needed to substantiate this claim. To further develop and test Fedora, collection builders and other staff require additional training to develop a working knowledge of web-services and XML technologies that are needed to build Fedora disseminators or applications. This statement begs the question of whether subject specialists and archivists should be the ones building collections.<sup>6</sup> It may be more efficient for those skills to be

---

<sup>5</sup> Before VITAL, POG another IAC committee recognized the need for training in this area and the library has begun to offer in house classes on XSLT!

<sup>6</sup> We don’t mean to imply that the subject specialist and the archivist should be excluded in any way from collection building. Rather we suggest a role where they define the requirements for a collection’s functional specification that is instantiated by a technologist.

developed in a small, core group of technologists who work with collection builders – so, for example, a subject specialist doesn't have to learn how to write an XSL style sheet, but understands how her collection would optimally be displayed in a DR interface and can communicate those requirements to the IT person who then translates those requirements into an XSL style sheet. This same skill set can also be re-purposed to build Fedora ingest and preservation workflow systems for discrete digital collections hosted by Fedora. If the investigation of native Fedora is a logical next step for the Library then the locus of development is a key issue for the library. Here there are four possible pathways.

- The first is the independent development of Fedora by the Library. As noted above the prerequisites for effective development is dependent upon the availability of resources to create and educate a Fedora development team. (The Collection Builders could serve the base for this group). This would take a substantial investment by the library and require a multiple-year commitment.
- Secondly, is if the Library's Fedora development was done as part of a consortia project. The Yale-Tufts Fedora preservation project could serve as a model. Here two institutions with a common objective, digital preservation, are jointly engaged in Fedora development. Similarly this model has potential to work for the Finding Aids collection where other Universities like the University of Virginia and Rutgers are engaged in similar work.
- The third pathway is to engage VTLS as a custom partner in the development of Fedora at Yale. This is akin to outsourcing some Fedora development thereby reducing risk and reducing time needed to build an application. The research project provided evidence that VTLS is a responsible and flexible vendor that is committed to making VITAL work for Yale. For, example VTLS staff worked over weekends, on their own time, to build a custom configuration of VITAL for the Yale project. In addition, VTLS support staff was responsive to Yale's questions about the VITAL software in proportion to the nature of the engagement. Technically the vendor's work with Australian Research Repositories Online or ARROW demonstrates that VTLS staff can build Fedora applications.
- The fourth pathway is to establish a Library-ITS partnership to develop Fedora. This could potentially help with the technical resource constraints that the library faces. A shared development environment is more economical to the University, promotes knowledge sharing and facilitates the integration of non-library collections with library collections and systems used for resource discovery.

**(3) The outcome of the VITAL project provides evidence that Yale's digital repository framework should be based upon a solution where multiple digital repositories are integrated through a tiered architecture as noted in IAC's report on Digital Repositories.**

The VITAL project use cases represent only a small sampling of Yale entities that may request repository services from the library. For example, there were no use cases that modeled repository requirements for faculty outputs or large scientific datasets generated in a researcher's laboratory. Any generalization or recommendation based upon the outcome of the VITAL research project must be understood in this context. Yet the results of the research provide some evidence that this environment at Yale (like peer institutions) is more reflective of the chaotic bazaar than the highly ordered cathedral. The use case dramatically illustrated Yale's repository requirements for workflow, ingest systems, user interfaces for discovery search and navigation are all over the map.

The Social Science Data Sets and Unicode use cases represent projects at the higher end of the spectrum for complexity where financial and technical resources are extant to work productively with Fedora. At the other end of the spectrum there may be other communities served by the Library which need help just to get their collection(s) into the digital game. The Oldest College Daily, better known as the Yale Daily News serves as a good example. Constrained by financial and technical resources, as a first step, this organization desires a repository system that has a basic ingest and content management functions that can make PDF data accessible over the network. Here VITAL or some other open source or commercial solution may meet their needs. Over time the library brings added value as a service provider to the Yale Daily News by strategically positioning their application so that it maybe enhanced as more sophisticated digital repository tools are developed down the road.

For the Library this suggests that a reasonable goal is to offer a spectrum of digital repository software solutions to its end users that are based upon open standards. These solutions can be integrated by a Yale centric repository interface layer which would provide access to the data from disparate application like Sakai or OAI for metadata harvesting.

## APPENDICIES

### APPENDIX A: The VTLS/VITAL Charge

#### VITAL Digital Repository Project CHARGE

##### Core Working Group:

David Gewirtz, project leader  
Jeff Barnett  
Gail Barnett  
Roy Lechich  
Ernie Marinko  
Matthew Beacom  
Gretchen Gano  
Art Belanger  
Nancy Godleski  
Elizabeth Beaudin

##### Advisory Group:

Meg Bellinger  
Audrey Novak  
Karen Reardon  
Stephen Yearl  
Derek Merleaux  
Kevin Glick  
Rebekah Irwin  
Brian Kupiec  
Martha Smalley  
Emily Horning  
Julie Linden  
Katie Bauer

Duration: August 15 – December 21

Time Commitment: 5%-20% varying from week to week, for Core Working Group members  
5%-20% for Advisory Group members, depending on availability

##### Charge:

The VITAL Digital Repository Project Working Group will conduct a 90-day project of the VITAL digital repository software (August 22 – November 23) in order to determine its suitability for the needs of the Yale Library and, potentially, for larger university needs. VITAL is a digital library application marketed by VTLS (a well-established

vendor of library systems) and based upon the open source Fedora software developed primarily at Cornell and UVA. The project will focus on two of the four types of digital repository defined in the July 2005 IAC Review of Digital Repositories report: “Homes for digital collections” and “Infrastructure for digital preservation.” The primary immediate need is for a flexible system that can provide a reliable long-term repository for a wide variety of content and metadata formats and can offer integrated user interfaces to the diverse content. Projects that have indicated interest in participating in or contributing to such a repository include examples of digital library collections, digital scholarship, and learning objects:

Yale Finding Aids (Beinecke, Divinity, MSSA, Music, etc.)  
Yale Daily News archive (including newspaper interface)  
Images, audio, video (potential backend to Insight and the DL, and possible Medical image colls)  
Digital content from social sciences and sciences (proposed by Gretchen Gano)  
Jonathan Edwards Papers  
Yale Indian Papers Project  
Human Relations Area Files  
OACIS/AMEEL Middle East project (Beth Beaudin has offered to provide test files)  
IMLS McClintock grant (TEI and images)  
Learning Objects from Classes\*v2 and/or ELI program

Integration of external resources and interoperability with external user environments (e.g. Sakai) are additional highly valued characteristics. Potential for future development of the system as an OAIS-compliant preservation archive is an important consideration.

The working group will accomplish the following tasks:

- Install the VITAL software on a Yale server, assisted by VTLS
- Configure the software for use in the project with minimal customization
- Develop use cases that test and exploit technical features of the VITAL package
- Build 3-5 small test collections drawn from the examples above and based upon the use cases (utilizing different content types, storage configurations, and ingest modes or workflows)
- Evaluate the several ingest and collection building mechanisms
- Evaluate user interfaces
- Evaluate metadata management capabilities
- Describe gaps in components of the VITAL software (VITAL Manager, VITAL Portal, VITAL System Administrator) that are used to create workflows, ingest content, brand collection interfaces and to manage the system
- Evaluate performance of the vendor (VTLS)
- Evaluate system operation and management characteristics, including LOE required
- Assess OAI capabilities (if possible)
- Assess interoperability with Sakai (if possible)
- Conduct small-scale usability test (if possible)
- Post weekly updates on progress of the project

- Produce a final report on the findings of the project and recommend next steps to LMT

Assignments (e.g. building collections, testing specific functions) and meeting frequency will be determined by working group members at the initial meeting. The project will begin in earnest on August 29<sup>th</sup> when VTLS will begin three day-long training sessions in various aspects of the system. Monthly meetings of the larger advisory group of stakeholders will be held to discuss issues related to the project. Members of the core working group will call upon staff outside the working group, including members of the advisory group, to assist with appropriate tasks. For example, the core group member(s) responsible for loading and testing finding aids will call upon members of the Finding Aids Task Force. The core group member(s) responsible for evaluating the user interface will call upon Katie Bauer to conduct a usability study.

Frederick Martz  
David Gewirtz  
Meg Bellinger (sponsor)

Yale University Library  
August 8, 2005

## APPENDIX B: The Use Case Template:

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** This is simply an example of a Use Case created from the Use Case template

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Reader has an interest in Use Cases and would like to learn how to use the Use Case Template

**Success end:** Reader understands the use case template.

**Failed end:** Reader is uncertain how to use the template and must consult the discussion list for help.

**Actors:** Reader, VITAL Group

**Primary Actor:** Reader

**Trigger:** Need to understand Use Case Template

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Slow readers will take longer to read the template example - latency can be addressed by skimming.

### Main Success Scenario

1. Reader clicks on Example Use Case link from the [Use Case Catalog](#)
2. Reader reads all elements of Use Case one by one
3. Reader stops at the end of the template.
4. Reader logs impressions

### Extensions

Reader may interrupt reading, and resume later. as a result the use case may go on and on and on and on

### References

### Associated Modules

### Implementations

### Advice and Experience

---



## APPENDIX C: The Collection and Advisory Builders Use Cases

### ARCHIVES:

#### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Load existing pdf's of scanned reference publications, add minimal metadata and index for basic text searching - leave door open for eventual creation of more in-depth EAD markup for advanced searching.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Need by reference archivists and researchers for ability to search full text of publication. Have publication in electronic form - scanned into multiple pdf files with associated OCR text.

**Success end:** Publication is available online for easy searching and retrieval of results.

**Failed end:** Publication is not discoverable by indexing or search and retrieval interface frustrates users.

**Actors:** VITAL Group (VG), VITAL System (VS), Collection Builder (CB), Reference Archivist (RA)

**Primary Actor:** CB

**Trigger:** Need for text searching of publication, existence of electronic version of publication

**Security Concerns:** None

**Logging:** confirmation of ingest of valid files, confirmation of indexing, web access analysis

**Performance Concerns:** linking of index search results to page images within pdf files

#### Main Success Scenario

1. CB creates repository/collection space
2. CB moves pdf files to repository
3. VS performs ingest and logs confirmation of valid or invalid files ingested
4. VS performs indexing and logs details of success or problems
5. CB ensures that ingest and indexing process are successful by examining logs and performing test searches
6. RA performs successful search and retrieval
7. VS creates analyzeable log of RA's web access.
8. CB analyzes web access log and interviews RA to determine levels of success for user-interface and indexing.

**Extensions:** At a later date, CB may wish to create a more detailed EAD markup for the

text index to allow a more advanced level of searching than the current simple full text search.

## References

**Associated Modules:** Creating of Repository/Collection and ingest by Manager Module, indexing by System Module, testing and use by web access module.

## Implementations

## Advice and Experience

---

## Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Create a repository/collection of MPEG2 files of Video Holocaust Testimonies (VHT) with a very high level of control over who may access which files and when.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Need for online access to VHT, existence of MPEG2 files and associated MARC and TEI metadata

**Success end:** VHT are accessible to users with access permission and attempts to view video without proper permission are denied

**Failed end:** Video is not discoverable or viewable, or access is not properly controlled

**Actors:** Video Archivist (VA), VITAL Group (VG), VITAL System (VS), Collection Builder (CB)

**Primary Actor:** CB

**Trigger:** Need for online access to digital video with solid access controls

**Security Concerns:** Serious privacy concerns are raised in creating online access to this collection.

**Logging:** Confirmation of ingest of valid files, confirmation of indexing, web access analysis including security log of any and all attempts to access material.

**Performance Concerns:** access controls may not be as granular as desired

### Main Success Scenario

1. CB creates repository/collection space.
2. CB moves MPEG2, MARC and TEI files to repository.
3. VS performs ingest and logs confirmation of valid or invalid files ingested.

4. VS performs indexing and logs details of success or problems.
5. CB ensures that ingest and indexing process are successful by examining logs and performing test searches.
6. CB assigns administrative control over access/permission module to VA
7. VA defines access permissions.
8. CB and VA test access permissions by verifying that authorized users can discover and access the collection and simulating unauthorized attempts to access the collection.
9. VS creates analyzeable log of RA's web access.
10. CB analyzes web access and security logs to determine levels of success for access control.

**Extensions:** VA may wish to control not only who may access the collection, but from where on the network may they access it and which specific parts of the collection they may access.

### **References**

**Associated Modules:** Creating of Repository/Collection and ingest by Manager Module, Assigning of permissions and access control by Manager Module, indexing by System Module, testing and use by web access module.

### **Implementations**

### **Advice and Experience**

---

## SOCIAL SCIENCE DATASET USE CASES

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to definitively isolate numeric data holdings from among the collections, regardless of the medium that houses them

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants to explore the data holdings on a topic, returning hits that may be housed externally, contained on cd, or contained in the repository.

**Success end:** user can definitively isolate data holdings on a topic from other collection items

**Failed end:** after executing a search that should limit to data holdings, user returns hits for other collection material

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** This is an issue that requires both a consistent technology solution and a consistent metadata treatment of the records. Perhaps there can be a way to build in automated verification or a controlled vocabulary to ensure record consistency.

### Main Success Scenario

1. user wants to find data holdings on a topic
2. user executes a search, imposing a limit for to return only available numeric data associated with the topic/search term
3. search returns hits that include only holdings of numeric data, regardless of the medium (cd/server/external download site)

### Extensions

### References

### Associated Modules

### Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User can view related publications information as a facet of the record display

**Version:** 0.2

**Priority:** MEDIUM

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** researcher finds a record and would like to view a list of publications that have used the data for analysis

**Success end:** researcher views a bibliographic citation for related publications

**Failed end:** researcher find no associated publications information, either because the record is incomplete, or because the interface does not allow for related publications display

**Actors:** researcher

**Primary Actor:** user

**Trigger:** need to become familiar with the characteristics of a given dataset

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Useful characteristics that should be displayed, and the nature of the display should be selected through a process of user testing

### Main Success Scenario

1. user is in [Sakai environment](#) and is looking at her homework assignment
2. user sees that there is a dataset associated with this evening's homework
3. user is able to preview summary statistics for the dataset within the Sakai interface before downloading
4. She could answer basic questions about the character of the dataset without loading it

### Extensions

### References

### Associated Modules

## Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User can preview aspects of the data (see frequencies/mean/mode) in the Sakai courseware environment.

**Version:** 0.2

**Priority:** MEDIUM

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user is a student in stats 101 -- she wants to look at the characteristics of the dataset that is associated with a class assignment before downloading the data to her stats package.

**Success end:** student previews frequencies, mean, mode and other summary statistics associated with a given dataset within the Sakai environment

**Failed end:** student sees only that the dataset is available for download and must download and load the datasets into SPSS, for example, to look at its characteristics

**Actors:** student

**Primary Actor:** user

**Trigger:** need to become familiar with the characteristics of a given dataset

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Useful characteristics that should be displayed, and the nature of the display should be selected through a process of user testing

### Main Success Scenario

1. user is in Sakai environment and is looking at her homework assignment
2. user sees that there is a dataset associated with this evening's homework
3. user is able to preview summary statistics for the dataset within the Sakai interface before downloading
4. She could answer basic questions about the character of the dataset without loading it

## Extensions

## References

## Associated Modules

## Implementations

## Advice and Experience

---

Assess data -- visualize data

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to visualize aspects fo the data files. Rather then reviewing tables, the user sees scatterplots/graphs, etc.)

**Version:** 0.2

**Priority:** LOW

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants to explore the data holdings visually

**Success end:** user can build custom visualizations of a data holding on the fly as a component of the search interface

**Failed end:** User cannot visualize any data holdings in any component of the search interface

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** interpretive need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:**

**Main Success Scenario**

1. user identifies a dataset that she would like to examine in the form of a graph
2. user selects components of the dataset (variables) to include in a visual analysis
3. user views a graph/scatterplot/etc. that has been generated based on her selections

## Extensions

## References

## Associated Modules

## Implementations

## Advice and Experience

---

## Extract Formatted Data

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to extract appropriately formatted data files and/or generate set-up files for the statistical package of her choice on the fly.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - data collection

**Preconditions:** user has found a suitable study record and wants to download data that can be read into a statistical package (say, SPSS)

**Success end:** user can download data formatted appropriately for loading into SPSS and/or user can generate a set-up file for reading the dataset into SPSS

**Failed end:** user cannot read the data she downloads into her preferred statistical package

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** I need to figure out the best way and the proper mechanism for tackling this end-user need. It may not be articulated clearly enough here to make it actionable.

### Main Success Scenario

1. user is viewing a study record and wants to perform analysis with the associated data
2. user downloads the data and loads it directly into her preferred statistical package.

## Extensions

## References

## Associated Modules

## Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to link directly to onlien file access areas in the context of records that are not housed in the repository

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants direct access to data files no matter where they are

**Success end:** user is able to access data files through a limited number of steps

**Failed end:** user cannot readily download data files

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** user will need additional informaiton to obtain data files from outside sources (additional log-ins, will need to navigate other catalogs, etc)

### Main Success Scenario

1. user is viewing a harvested repository record and wants to get to the data
2. user selects file download and is taken seamlessly to the external database
3. user can directly download the data files

## Extensions

## References

## Associated Modules

## Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to definitively isolate numeric data holdings from among the collections, regardless of the medium that houses them

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants to explore the data holdings on a topic, returning hits that may be housed externally, contained on cd, or contained in the repository.

**Success end:** user can definitively isolate data holdings on a topic from other collection items

**Failed end:** after executing a search that should limit to data holdings, user returns hits for other collection material

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** This is an issue that requires both a consistent technology solution and a consistent metadata treatment of the records. Perhaps there can be a way to build in automated verification or a controlled vocabulary to ensure record consistency.

### Main Success Scenario

1. user wants to find data holdings on a topic
2. user executes a search, imposing a limit for to return only available numeric data associated with the topic/search term
3. search returns hits that include only holdings of numeric data, regardless of the medium (cd/server/external download site)

### Extensions

### References

### Associated Modules

## Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to link directly to access the full-text of related publications from the citation information provided in the related publication area of the study catalog record

**Version:** 0.2

**Priority:** MEDIUM

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants direct access to related publications

**Success end:** user is able to access related publication full-text through a limited number of steps

**Failed end:** user cannot readily access full text information from the study record related publications citations

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:**

**Main Success Scenario**

1. user is viewing a data study record and wants to read the related publications associated with it
2. user clicks on a hot link associated with the bibliographic citation of related publication and is directed to the appropriate citation within a full-text database
3. user can download the full-text of the related pub

## Extensions

## References

## Associated Modules

## Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to download all of the files associated with a study record and the downloaded files should retain their file type definitions.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user has found a suitable study record that has associated files

**Success end:** user is able to download all of the associated files

**Failed end:** user cannot readily download the files and/or the files do not retain original file type definitions

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** the files should also exhibit a logical naming structure that links them to the central file study.

### Main Success Scenario

1. user is viewing a study record and wants to access the associated data files
2. user selects files to download and can do so readily from the study record page

### Extensions

### References

### Associated Modules

### Implementations

## Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to download a full-DDI record for the data contained in the repository and for harvested records

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants to capture a full descriptive record of the data she will use

**Success end:** user downloads complete DDI records for the studies of interest

**Failed end:** user cannot readily download descriptive information for a study of interest

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** user will not get complete information about the study

### Main Success Scenario

1. searcher [selects to view the catalog record of a data study](#)
2. searcher can look at a brief record, or choose to examine the full DDI record
3. user sees a clearly displayed, well-formatted DDI display

### Extensions

### References

### Associated Modules

### Implementations

### Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to search variable level information from DDI metadata

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user is looking for data holdings with that contain particular variables

**Success end:** searcher returns a list of data resources that include particular variables

**Failed end:** searcher returns an incomplete and/or unrepresentative list of resources

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** users will miss studies that include relevant variables

### Main Success Scenario

1. searcher uses search interface to search by the names or descriptions of variables
2. searcher returns results that include particular variables
3. user is able to readily compare variable usage between and among the list of studies returned

### Extensions

### References

### Associated Modules

### Implementations

### Advice and Experience

---

Example Use Case

Last changed on 18-Aug-2005 by [Jeffrey Barnett](#)

### Example Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** User should be able to call up both a brief and a long record display to examine the DDI metadata

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project

**Preconditions:** user wants to examine aspects about the study that are contained in the DDI specification

**Success end:** user examines all available DDI-specific metadata

**Failed end:** user examines only partial or incomplete information OR it is poorly formatted

**Actors:** user/searcher

**Primary Actor:** user

**Trigger:** research need

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** user will not get complete information about the study

**Main Success Scenario**

1. searcher selects to view the catalog record of a data study
2. searcher can look at a brief record, or choose to examine the full DDI record
3. user sees a clearly displayed, well-formatted DDI display

**Extensions**

**References**

**Associated Modules**

**Implementations**

**Advice and Experience**

---

FINDING AIDS USE CASES

**OAI Harvest Finding Aids**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

## [Experience](#)

**Goal:** OAI-enable EAD Finding Aids residing in VITAL repository

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - EAD Finding Aids

**Preconditions:** EAD Finding Aids have been ingested into a VITAL/Fedora repository.

**Success end:** OAI harvesters (e.g. MetaLib) are able to find, retrieve, and index the DC metadata related to the Finding Aids in a repository.

**Failed end:** OAI harvesters are not able to locate or retrieve DC metadata.

**Actors:** OAI harvesting systems, Yale's MetaLib system.

**Primary Actor:** OAI harvesting system.

**Trigger:** Need to discover and harvest MetaData for eventual indexing and inclusion in another collection's search results.

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Harvesting may impact performance of repository system.

### **Main Success Scenario**

1. MetaLib's OAI harvester is run and discovers and successfully harvests DC metadata for Finding Aids in the VITAL repository.
2. MetaLib's Xindex (?) uses harvested metadata to include in its cross-collection index.
3. Finding Aids, that reside in the VITAL repository, are included in MetaLib search results and are accessible from those search results.
- 4.
- 5.

### **Extensions**

### **References**

### **Associated Modules**

### **Implementations**

### **Advice and Experience**

---

## **Search Finding Aids -Advanced**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and](#)

## Experience

**Goal:** Advanced search options in the VITAL Access Portal meet the requirements of researchers and librarians who work with EAD Finding Aids.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project -EAD Finding Aid

**Preconditions:** Researcher has specific question to answer and knowledge of complex Finding Aid structure

**Success end:** Researcher constructs complex search and refines results to pinpoint location of specific information needed.

**Failed end:** Search results are insufficient or too broad to be useful.

**Actors:** Researcher, archivist, librarian

**Primary Actor:** Researcher

**Trigger:** Need to locate specific data within the Finding Aids repository

**Security Concerns:** None

**Logging:** System log for later analysis of activity.

**Performance Concerns:** Slow response to search requests could discourage researcher from continuing to use the repository as a resource.

### **Main Success Scenario**

1. Researcher accesses VITAL/Fedora repository Access Portal.
2. Researcher selects 'Advanced Search' option and is able to construct complex queries using multiple keywords or phrases, with boolean operators 'and', 'or', 'not', and 'near' (E).
3. Researcher is able to further refine results by limiting searching to a single repository/collection or to text within specific EAD tags.
4. Researcher receives a relatively small set of search results pinpointing the specific information needed.

### **Extensions Extensions to these Advanced Search options and features include:**

Searching within the displayed Finding Aid (HD),  
Searching with alternate scripts (support Unicode UTF8) (E),  
Flexible proximity - 'near' within 'n' words (HD),  
Use of 'follows' as an operator - similar to 'near',  
Searching within the displayed Finding Aid (HD).

## **References**

### **Associated Modules**

VITAL Web Access Portal

### **Implementations**

### **Advice and Experience**

---

## Search Finding Aids-Simple Search

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Access the VITAL/Fedora repository and discover a variety of information, including Finding Aids and related digital objects held in Yale libraries.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - EAD Finding Aids

**Preconditions:** Student has an assignment and a topic of interest, but not a familiarity with EAD or Finding Aids.

**Success end:** Student finds a manuscript or archive collection of interest, guidelines for its use, and, perhaps, links to other related digital objects.

**Failed end:** Student does not find useful search results or has difficulty navigating search results.

**Actors:** Student

**Primary Actor:** Student

**Trigger:** Classroom assignment or need to find research materials and information.

**Security Concerns:** None

**Logging:** System

**Performance Concerns:** Slow response to searches, and slow downloading of very long documents would discourage further use of this Web resource.

### Main Success Scenario

1. Student accesses the VITAL/Fedora Repository Access Portal.
2. Student enters a single search term or simple keyword search using 'and'.
3. Student receives results set, including Finding Aids for manuscript or archive collections in a Yale library, and, perhaps, images or full text documents linked to the Finding Aids. (See Extension 1. and 2.)
4. Student selects Detail View of a Finding Aid and is presented with basic tools to easily navigate the Finding Aid structure and an easy return-to-search-results option. (See Extension 3.)

### Extensions

1. Search results for a Finding Aid should include:
  - Finding Aid title (E),
  - Repository/location of the collection,
  - A summary or abstract,
  - File size or a warning if file is very large,
  - Number of 'hits' for search terms in the document (E).

2. Search results screen should include options to resort results by title, repository, number of hits (E).
3. Basic navigation options should include:
  - Browser scrolling edit->find-in-page features,
  - A Table of Contents with links to Finding Aid sections.

## References

### Associated Modules

VITAL Web Access Portal

### Implementations

### Advice and Experience

---

## Ingest Finding Aid -Remote

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Ingest a newly created or revised EAD Finding Aid into the VITAL/Fedora repository using the VITAL Web Submission form or VITAL Manager client, create required metadata and dynamically indexing the metadata and full text.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - EAD Finding Aids

**Preconditions:** New or revised EAD Finding Aid is ready to add to the VITAL/Fedora repository.

**Success end:** EAD file successfully ingested and indexed and appropriate metadata created.

**Failed end:** Web submission process (or Manager client) failed to ingest the EAD file/object, along with metadata and indexing.

**Actors:** Collection Builder, Collection Owner, archivist, Sys Admin

**Primary Actor:** Collection Builder

**Trigger:** Need to add new or revised EAD Finding Aid to the VITAL/Fedora repository.

**Security Concerns:** None

**Logging:** Log/report of EAD instances successfully ingested and errors/problems flagged.

**Performance Concerns:** Indexing 'on-the-fly' may impact performance of the repository search and display functions.

### Main Success Scenario

1. Sys Admin configures the Web Submission process based on requirements

- specified by Collection Owners.
2. Collection Builder does project Web Submission(s) and examines results through through the VITAL Access Portal. (System Manager client could be used in place of Web Submission or Access Portal.)
  3. Sys Admin adjusts Web Submission configuration if necessary.
  4. Collection Builder successfully adds new EAD file to the repository and confirms that search, retrieval, and display functions work well for the new file.

### **Extensions**

If possible, configure Web submission workflow to create or include MARCXML, Yale Element metadata, and a PDF version of the Finding Aid as part of the ingest process.

### **References**

### **Associated Modules**

VITAL Web Submission or VITAL Manger client

### **Implementations**

### **Advice and Experience**

---

## **Harvest Finding Aids -Web**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Enable Finding Aids in VITAL/Fedora repository for harvesting by RLG

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - EAD Finding Aids

**Preconditions:** EAD Finding Aids ingested in new repository; RLG harvesting methods understood; similar features enabled in VITAL/Fedora repository.

**Success end:** RLG is able to continue to harvest Yale libraries' finding aids after migration to new repository system.

**Failed end:** RLG no longer able to harvest Yale Finding Aids for inclusion in RLG combined system.

**Actors:** SysAdmin, Colection owners.

**Primary Actor:** Not sure.

**Trigger:** Need to continue to provide functions available in old Finding Aids repository.

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Impact on performance of host sytem while harvesting process runs.

## Main Success Scenario

- 1.
- 2.

## Extensions

## References

## Associated Modules

## Implementations

## Advice and Experience

---

## Ingest Finding Aids-Batch

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Ingest multiple EAD Finding Aids into the VITAL/Fedora repository in a batch job, creating required metadata, and indexing full text at the same time.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - EAD Finding Aids

**Preconditions:** Pre-existing EAD Finding Aids available in previous system or database.

**Success end:** Batch of Finding Aids are successfully ingested into new system/repository and are searchable and accessible through repository's Web portal.

**Failed end:** Batch job failure, appropriate metadata not created, or Finding Aids not indexed correctly and not accessible through Web interface.

**Actors:** SysAdmin, collection owners

**Primary Actor:** SysAdmin

**Trigger:** New VITAL/Fedora repository up and running; Finding Aids ready to be ingested.

**Security Concerns:** None

**Logging:** Log/report of EAD files successfully ingested and problems or failures documented

**Performance Concerns:** Performance of batch ingest process; optimal number of EAD files to include in a batch.

## Main Success Scenario

1. SysAdmin configures batch job options based on requirements specified by collection owners and EAD creators.
2. SysAdmin tests configuration on small but varied batch of EAD files.
3. Collection owners test and examine results, confirm content, full-text indexing, and functionality.
4. SysAdmin adjusts batch job configuration options, if necessary, and then runs the batch job(s) to ingest full set of EAD Finding Aids for further implementation and testing of new functions.

### **Extensions**

If possible, create or include MARCXML, Yale Core metadata, as well as PDF version of EAD Finding Aid during batch ingest.

### **References**

### **Associated Modules - VITAL Batch Ingest**

### **Implementations**

### **Advice and Experience**

---

## **Navigate EAD Finding Aid**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Easily navigate the complex structure of EAD Finding Aids with presentation options and tools provided in the repository Web interface.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project - EAD Finding Aids

**Preconditions:** Researcher found and retrieved an EAD Finding Aid document from the repository.

**Success end:** Researcher found the presentation options and navigation tools easy to use and successfully located specific information within the document; also he/she was able to save or print the document as required.

**Failed end:** Researcher was not able to effectively navigate the large multi-part document with the tools provided.

**Actors:** Researcher

**Primary Actor:** Researcher

**Trigger:** Researcher seeks specific information within a Finding Aid as well as a printed or saved copy of all or part of the document.

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Large EAD files may take too long to open and to navigate in w Web browser.

**Main Success Scenario**

1. Researcher accesses VITAL/Fedora repository, searches, finds, and retrieves a large EAD Finding Aid. (See Extension 3.)
2. Researcher finds that EAD-specific navigation tools in the 'Detail View' reveal the structure of the Finding Aid, as well as the contents, and help in locating required information within the document (See Extension 1 and 2.)
3. Researcher finds options for a 'printer-friendly' (PDF) version of the Finding Aid document and for saving information and notes in a 'bookbag' for future reference. Bookbag contents could include FA title, repository/location, abstract, and reference URL.

**Extensions**

1. The 'Table of Contents' provides links to sections of the Finding Aid document, as well as these navigation features:
  - TOC is persistent - it 'floats' (stays in view) as one scrolls down through the content frame,
  - TOC highlights/flags the section of content currently in view,
  - TOC reflects hierarchical structure of FA with expandable/collapsible section headings,
2. Navigation through matches (an option to jump to next occurrence of a search term) is also highly desirable.

**References**

**Associated Modules**

VITAL Web Access Portal

**Implementations**

**Advice and Experience**

## IMAGES-AUDIO-VIDEO USE CASES:

Added by Ernie Marinko, last edited on Sep 15, 2005

Use Case Name: Search and Display Images

Actors: Patron, Collection Builder, Advisor

### Summary:

To test the search and display of multi-part records that have been ingested and indexed in VITAL\Fedora.

### Basic Course of Events:

1. A collection of images and metadata has been successfully ingested and indexed.
2. Modify the skins (HTML and CSS) that display the search selection boxes and look of the site.
3. The Patron uses the interface.

### Alternative paths:

1. None.

### Exception Paths:

1. Errors. Images are not searchable.
2. Images are not displayed.
3. Images are displayed but not metadata.
4. Multi-part images are not displayed.

### \*Extension Points: \*

1. Upon successful completion the Patron can search and display multi-part items in the web interface.

### Triggers:

1. The Patron wants to search and display multi-part records

### Assumptions:

1. The previous 2 use cases have been completed.

### Preconditions:

1. The images and metadata have been successfully ingested and indexed.

Postconditions:

1. The images and metadata can be searched and displayed in the web interface.

Author: Ernie Marinko  
Date: 09/15/2005  
Use case number: 3  
Use case category: Images

Scenarios:

1. The Patron browses to interface page.
2. Inputs search criteria.
3. System displays list of results.
4. Patron clicks one of the results.
5. The system displays the record.
6. The Patron evaluates that they can access all parts of the multi-part record.

Added by Ernie Marinko, last edited on Sep 15, 2005

Use Case Name: Index Images

Actors: Collection Builder, Advisors

Summary:

To test the indexing of multi-part records that have been ingested into VITAL\Fedora. using the same indexing fields as the BRBL DL.

Basic Course of Events:

1. A collection of images and metadata has been successfully ingested.
2. Using the VITAL Administrative Tools module, images are indexed.

Alternative paths:

1. None.

Exception Paths:

1. Errors. Images are not indexed.

\*Extension Points: \*

1. Upon successfully completion proceed to the "Search and Display Images" use case.

Triggers:

1. The Collection Builder wants to index multi-part records so they can later be searched and displayed.

Assumptions:

1. The indexing program will work.

Preconditions:

1. The images and metadata have been successfully ingested.

Postconditions:

1. The images and metadata are successfully indexed.

Author: Ernie Marinko

Date: 09/15/2005

Use case number: 2

Use case category: Images

Scenarios:

1. A collection of multi-part images from BRBL will be indexed.

Added by Ernie Marinko, last edited on Sep 15, 2005

Use Case Name: Ingest Images

Actors: Collection Builder, Advisors

Summary:

To test the ingest of images and descriptive metadata in a batch mode, of multi-part records using the same record structure as the BRBL DL. Ingesting images from both a share and from a CD and associated metadata from a .txt file.

Basic Course of Events:

1. A collection of images is identified.
2. Using the VITAL batch utility, images are ingested from Rescue Repository and from CDs.
3. Appropriate descriptive metadata is linked to appropriate image.

Alternative paths:

1. None.

Exception Paths:

1. Errors. Images are not ingested.
2. Images are ingested but not linked to metadata.

\*Extension Points: \*

1. Upon successfully completion proceed to the Indexing Images use case.

Triggers:

1. The Collection Builder wants to ingest multi-part records into VITAL.

Assumptions:

1. The batch program will work.

Preconditions:

1. A collection of images had been chosen by the Advisors along with both technical and descriptive metadata.

Postconditions:

1. The images and metadata are successfully ingested into VITAL .

Author: Ernie Marinko

Date: 09/15/2005

Use case number: 1

Use case category: Images

Scenarios:

1. A collection of images from BRBL have been identified. They are multi-part images. One set of metadata need to link to several images. For example a letter has 3 pages. Front and back or 1 page and the front of another. The metadata describes those 3 images but they are really one thing.

## MEDICAL IMAGES USE CASES

### Web Submission with Workflow

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Load PDFs and metadata into vital/fedora via web submission

**Version:** 0.1

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project server

**Preconditions:** Paper documents, workflow processes defined

**Success end:** Digital objects are available through Vital Access Portal; discoverable through indexing of key fields

**Failed end:** Object is not discoverable through indexing; object is intact.

**Actors:** Library Staff (LS), collection builder (CB), SysAdmin (sa), Collection Team (CT)

**Primary Actor:** Library Staff

**Trigger:** Paper Documents made available

**Security Concerns:** Normal ACL controls

**Logging:** System generated log emailed to library staff

**Performance Concerns:** Completion in reasonable amount of time.

#### Main Success Scenario

1. CB logs in to establish authority
2. CB configures workflow files.
3. CB writes web pages.
4. LS scans item and created PDF.
5. LS logs on to establish authority.
6. LS initiates record upload by completing step 1.
7. LS complete following steps when notified.
8. LS loads into repository when last step is complete
9. LS uses portal to retrieve objects by indexes.

#### Extensions

#### References

#### Associated Modules

Web submission, Administration, Web Portal

#### Implementations

#### Advice and Experience

---

Batch Submission

Last changed on 13-Sep-2005 by [Arthur Belanger](#)

## **Batch Submission**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Load existing metadata and images into vital/fedora

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project server

**Preconditions:** Existing images and metadata, qualified DC in XML format

**Success end:** Digital objects are available through Vital Access Portal; discoverable through indexing of key fields

**Failed end:** Object is not discoverable through indexing; object is intact.

**Actors:** Library Staff (ls), collection builder (cb), SysAdmin (sa)

**Primary Actor:** Collection Builder

**Trigger:** Notification of data ready to be uploaded

**Security Concerns:** Normal ACL controls

**Logging:** System generated log emailed to library staff

**Performance Concerns:** Completion in reasonable amount of time.

## **Main Success Scenario**

1. Collection builder logs in to establish authority.
2. Collection builder prepares batch configuration file
3. Collection builder prepares object model file
4. Collection Builder uploads image and metadata files to ingest directory.
5. Collection builder initiates batch ingest.
6. cb requests index creation
7. sa defines indexes
8. cb, ls use portal to retrieve objects by indexes.

## **Extensions**

Ingest failure: Checkpoint/Restart

## **References**

Normal ACL controls as documented (here)

## **Associated Modules**

Batch submission, Administration, Web Portal

## **Implementations**

## **Advice and Experience**



## UNICODE USE CASES

### Unicode Use Case: VerifyRTL

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Verify of correct RTL orientation of ingested Unicode-compliant text files

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Systems Rep or Collection Builder notifies Q/A reviewer that batch of Unicode-compliant text files have been loaded

**Success end:** Q/A reviewer is able to select, display, and verify that a selection of newly added text files correctly show RTL orientation for display of Arabic

**Failed end:** Q/A reviewer is not able to either 1)review a selection of newly added text files or 2) sees that files lack RTL or are shown using LTR

**Actors:** Systems Rep (SR), Collection Builder (CB), Q/A reviewer (QA)

**Primary Actor:** Q/A reviewer

**Trigger:** Need to verify correct display of contents, based on ?? language code in metadata ??

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Slow response time in search and display will cause frustration and postpone other Q/A steps

#### Main Success Scenario

1. QA opens access to Fedora repository
2. QA starts search for batch contents by using Advanced Search or Expert Search
3. QA checks that text file correctly displays in RTL
4. QA checks that Unicode contents display using correct code page, e.g. UTF-8, or Arabic
5. QA repeats steps 2-4 for representative sampling of new batch
6. QA notes and reports any issues for review by CB

#### Extensions

Q/A reviewer may wish to return to review same batch; saved queries desirable

#### References

#### Associated Modules

IngestUniBatch; IngestUniImages; VerifyUniImages

#### Implementations

#### Advice and Experience

---

## Unicode Use Case: IngestUniImages

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Accept batch of processed Unicode-compliant image files

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Outside processing facility prepares image files during scanning and processing workflow steps; Unicode-compliant text files also accompany these files which are .pdf or .jpeg image files

**Success end:** Systems Rep or Collection Builder is able to add batch to repository with no errors, i.e. no rejected objects.

**Failed end:** System rejects an object; system loads files to incorrect parent object.

**Actors:** Systems Rep, Collection Builder, Q/A reviewer

**Primary Actor:** Systems Rep

**Trigger:** Need to accept file to complete parent and child object datastreams

**Security Concerns:** None

**Logging:** log on success: number of objects ingested, parent object metadata; log on failure: report failure type, report failure location in code if possible.

**Performance Concerns:** consequences of stopping batch job prior to completion

### Main Success Scenario

1. SR or CB opens batch submission window
2. SR or CB fills in batch details
3. SR or CB initiates batch job
4. SR or CB reviews successful log report

### Extensions

Systems notifies CB or QA that batch is ready for review

### References

### Associated Modules

VerifyUniImages;

### Implementations

## Advice and Experience

---

### Unicode Use Case: IngestRemoteBatch

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Accept objects in batch mode for processed Unicode-compliant text files and related image files into Fedora repository from remote server site

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Project partner Bibliotheca Alexandrina in Alexandria, Egypt prepares objects during full workflow, i.e. scanned and processed image of text page in .pdf or .jpeg along with Unicode-compliant text files accompany .pdf or .jpeg image files. Objects are staged on a server remote to Fedora repository.

**Success end:** Systems Rep or Collection Builder is able to add batch to repository with no errors, i.e. no rejected objects.

**Failed end:** System fails to connect; system fails to authenticate to established security protocol; system rejects an object; system loads files to incorrect parent object.

**Actors:** Systems Rep (SR), Collection Builder (CB), Q/A reviewer (QA), Remote unit (RU)

**Primary Actor:** Systems Rep

**Trigger:** Need to accept object into complete parent and child object datastreams

**Security Concerns:** None

**Logging:** log on success: number of objects ingested, parent object metadata; log on failure: report failure type, report failure location in code if possible.

**Performance Concerns:** consequences of bad or refused connect to remote server or interrupted connect to remote server; consequences of stopping batch job prior to completion.

#### Main Success Scenario

1. SR or CB opens batch submission window
2. SR or CB fills in batch details, including authentication parameters to connect securely to remote server
3. SR or CB initiates batch job
4. SR or CB reviews successful log report

5. SR or CB disconnects from remote server
6. SR or CB sends notification to QA and RU

**Extensions**

Systems notifies CB or QA that batch is ready for review

**References****Associated Modules**

an associated content verification case

**Implementations****Advice and Experience****Unicode Use Case: IngestUniBatch**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Accept batch of processed Unicode-compliant text files into Fedora repository

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Outside processing facility prepares text files during OCR workflow step; Unicode-compliant text files accompany .pdf or .jpeg image files

**Success end:** Systems Rep or Collection Builder is able to add batch to repository with no errors, i.e. no rejected objects.

**Failed end:** System rejects an object; system loads files to incorrect parent object.

**Actors:** Systems Rep (SR), Collection Builder (CB), Q/A reviewer (QA)

**Primary Actor:** Systems Rep

**Trigger:** Need to accept file to complete parent and child object datastreams

**Security Concerns:** None

**Logging:** log on success: number of objects ingested, parent object metadata; log on failure: report failure type, report failure location in code if possible.

**Performance Concerns:** consequences of stopping batch job prior to completion .

**Main Success Scenario**

1. SR or CB opens batch submission window
2. SR or CB fills in batch details
3. SR or CB initiates batch job

#### 4. SR or CB reviews successful log report

##### **Extensions**

Systems notifies CB or QA that batch is ready for review

##### **References**

##### **Associated Modules**

VerifyUniBatch; IngestImageBatch

##### **Implementations**

##### **Advice and Experience**

#### **Unicode Use Case: VerifyUniBatch**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Verify batch of processed Unicode-compliant text files

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Systems Rep or Collection Builder notifies Q/A reviewer that batch of Unicode-compliant text files have been loaded

**Success end:** Q/A reviewer is able to select, display, and review a selection of newly added text files

**Failed end:** Q/A reviewer is able to either: 1)select, 2)display in correct code page, or 3)review a selection of newly added text files

**Actors:** Systems Rep (SR), Collection Builder (CB), Q/A reviewer (QA)

**Primary Actor:** Q/A reviewer

**Trigger:** Need to verify file addition and view correct display of contents

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Slow response time in search and display will cause frustration and postpone other Q/A steps

##### **Main Success Scenario**

1. QA opens access to Fedora repository
2. QA starts search for batch contents by using Advanced Search or Expert Search
3. QA checks that text file correctly loaded to parent
4. QA checks that Unicode contents display using correct code page, e.g. UTF-8, or

Arabic

5. QA repeats steps 2-4 for representative sampling of new batch
6. QA notes and reports any issues for review by CB

### **Extensions**

Q/A reviewer may wish to return to review same batch; saved queries desirable

### **References**

### **Associated Modules**

IngestUniBatch; IngestUniImages; VerifyUniImages

### **Implementations**

### **Advice and Experience**

---

## **Unicode Use Case: IngestUniImages**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Verify batch of processed imaged files (these may be .pdf or .jpeg)

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** VITAL project Example

**Preconditions:** Systems Rep or Collection Builder notifies Q/A reviewer that batch of image files have been loaded

**Success end:** Q/A reviewer is able to select, display, and review a selection of newly added text files

**Failed end:** Q/A reviewer is able to either: 1)select, 2)display, or 3)review a selection of newly added image files

**Actors:** Systems Rep (SR), Collection Builder (CB), Q/A reviewer (QA)

**Primary Actor:** Q/A reviewer

**Trigger:** Need to verify file addition and view correct display of contents

**Security Concerns:** None

**Logging:** manual

**Performance Concerns:** Slow response time in search and display will cause frustration and postpone other Q/A steps

### **Main Success Scenario**

1. QA opens access to Fedora repository
2. QA starts search for batch contents by using Advanced Search or Expert Search
3. QA checks that text file correctly loaded to parent, and corresponding text file
4. QA checks that Unicode contents display using correct code page, e.g. UTF-8, or Arabic
5. QA checks that image display is legible.
6. QA repeats steps 2-5 for representative sampling of new batch
7. QA notes and reports any issues for review by CB

### **Extensions**

QA reviewer may wish to return to review same batch; saved queries desirable; CB may wish to verify QA's findings, saved queries desirable.

### **References**

### **Associated Modules**

IngestUniBatch; IngestUniImages; VerifyUniBatch

### **Implementations**

### **Advice and Experience**

## YALE INDIAN PAPERS PROJECT (YIPP) USE CASES

### Image Discovery for YIPP Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Locate original document images based on the text of the transcript

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** Yale Indian Papers Project

**Preconditions:** Images of primary documents and project transcript documents have been ingested and indexed as a single object. The accession number of the document has been recorded and indexed

**Success end:** The located document correctly matches the transcription

**Failed end:** The matching document is not found

**Actors:** YIPP collection user, YIPP staff

**Primary Actor:** YIPP collection user

**Trigger:** Need to locate a specific YIPP document image

**Security Concerns:** access is only available to fully accessioned documents

**Logging:** automatic

**Performance Concerns:** Slow transaction response may distract user between searches

#### Main Success Scenario

1. User finds YIPP via Web
2. System presents search for available documents
3. User enters text to be found and identifies document of interest
4. User clicks on [icon within] text transcript entry
5. System displays correct document image
6. System records document access

#### Extensions

Staff may not have completed ingest of requested document ... an explanation is presented to the user. The system notifies staff that the document was requested.

#### References

#### Associated Modules

Vital access module

#### Implementations

#### Advice and Experience

## Image Discovery from Index YIPP Use Case

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Locate original document images based on the accession numbers recorded in the project index documents.

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** Yale Indian Papers Project

**Preconditions:** Images of primary documents an project index documents have been ingested and indexed. The accession numbers of both the document and index entries have been recorded and indexed

**Success end:** The located document correctly matches the index entry

**Failed end:** The matching documented is not found

**Actors:** YIPP collection user, YIPP staff

**Primary Actor:** YIPP collection user

**Trigger:** Need to locate a specific YIPP document

**Security Concerns:** access is only available to fully accessioned documents

**Logging:** automatic

**Performance Concerns:** Slow transaction response may distract user between submissions

### Main Success Scenario

1. User finds YIPP via Web
2. System presents index of available documents
3. User scans index and identifies document of interest
4. User clicks on [icon within] index entry
5. System displays correct document
6. System records document access

### Extensions

Staff may not have completed injest of requested document ... an explanation is presented to the user. The system notifies staff that the document was requested.

### References

### Associated Modules

### Implementations

### Advice and Experience

---

Image Load for YIPP Use Case  
Last changed on 21-Aug-2005 by [Jeffrey Barnett](#)

## **Image Load for YIPP Use Case**

[Main Scenario](#) [Extensions](#) [References](#) [Associated Modules](#) [Implementations](#) [Advice and Experience](#)

**Goal:** Load Images of the original papers for subsequent indexing and transcription

**Version:** 0.2

**Priority:** HIGH

**Status:** DRAFT

**Scope:** Yale Indian Papers Project

**Preconditions:** Images have been scanned to Archival standards, and basic metadata has been captured including accession number

**Success end:** Images and metadata are transferred without error, and can be retrieved by accession number

**Failed end:** Images or metadata is corrupt, missing, or cannot be retrieved.

**Actors:** YIPP staff

**Primary Actor:** YIPP staff

**Trigger:** Need to conserve and organise YIPP documents

**Security Concerns:** Activity limited to authorised staff only

**Logging:** automatic

**Performance Concerns:** Slow transaction response may distract staff between submissions

### **Main Success Scenario**

1. Staff logs in and established YIPP authority
2. Staff identifies prepared image and metadata files
3. System ingests and indexes image and metadata
4. Ingest is automatically logged
5. A project retrieval of submitted files using accession number is successful

### **Extensions**

Staff may interrupt processing, and resume later. as a result the staff may need to reestablish authority

### **References**

### **Associated Modules**

### **Implementations**

### **Advice and Experience**



## APPENDIX D: Collection Builders and Advisors Use Case Reports

Use Case: Scanned Referenced Publications

Author: Derek Merleaux -- Manuscripts and Archives

Here are my feedback notes for the VITAL use cases. I told David that you had agreed to communicate this to the group. I will be out of town after Tuesday, but I will be checking email so if you have questions - please feel free to send them to me and I'll do my best to answer them.

The following is a step by step response to the "Main Success Scenario" section of the use case for Scanned Reference Publications.

1. CB creates repository/collection space

Individual instance of repository was created for the CB by admin staff - access by client and web access tool was uneventful

2. CB moves pdf files to repository

CB didn't have time to become familiar with latest version of Batch Ingest tool so only the client was used. PDF files were moved into the repository instance through the client using both the Watch Folder method and the Drag-and-Drop method. Neither functioned seamlessly, the Watch Folder did not actually function at all. The drag and drop functioned with some bugs - most notably that the interface reported most transfers status as failed, but consulting the web access page showed that the objects were ingested to the repository. (see clientLoadFailed.jpg)

3. VS performs ingest and logs confirmation of valid or invalid files ingested

See above for details of logging/status report failure. The client did display in addition to the pdf object, a DC datastream and a Full Text data stream.

4. VS performs indexing and logs details of success or problems

Indexing appeared to perform successfully although the CB did not access any logs on this...

5. CB ensures that ingest and indexing process are successful by examining logs and performing test searches

Test searches indicated that the indexing was successful

6. RA performs successful search and retrieval

Searches were successful, retrieval was mostly successful. Using the web access page, the pdf object could be easily downloaded as a file. Using the page viewer proved a bit frustrating. The load time for the viewer was extremely slow and the viewer did not offer any advantage over simply downloading the pdf and viewing it in Acrobat Reader. In fact, the lack of integrated searching and table of contents navigation made the viewer useful only if one wished to page through the entire file in a linear manner. The viewer also displayed some instability and produced errors (see imageVwr\_error.jpg) in one

instance the thumbnail images on the bottom of the screen began to be replaced by broken image boxes when they were clicked and the other parts of the navigation froze up.

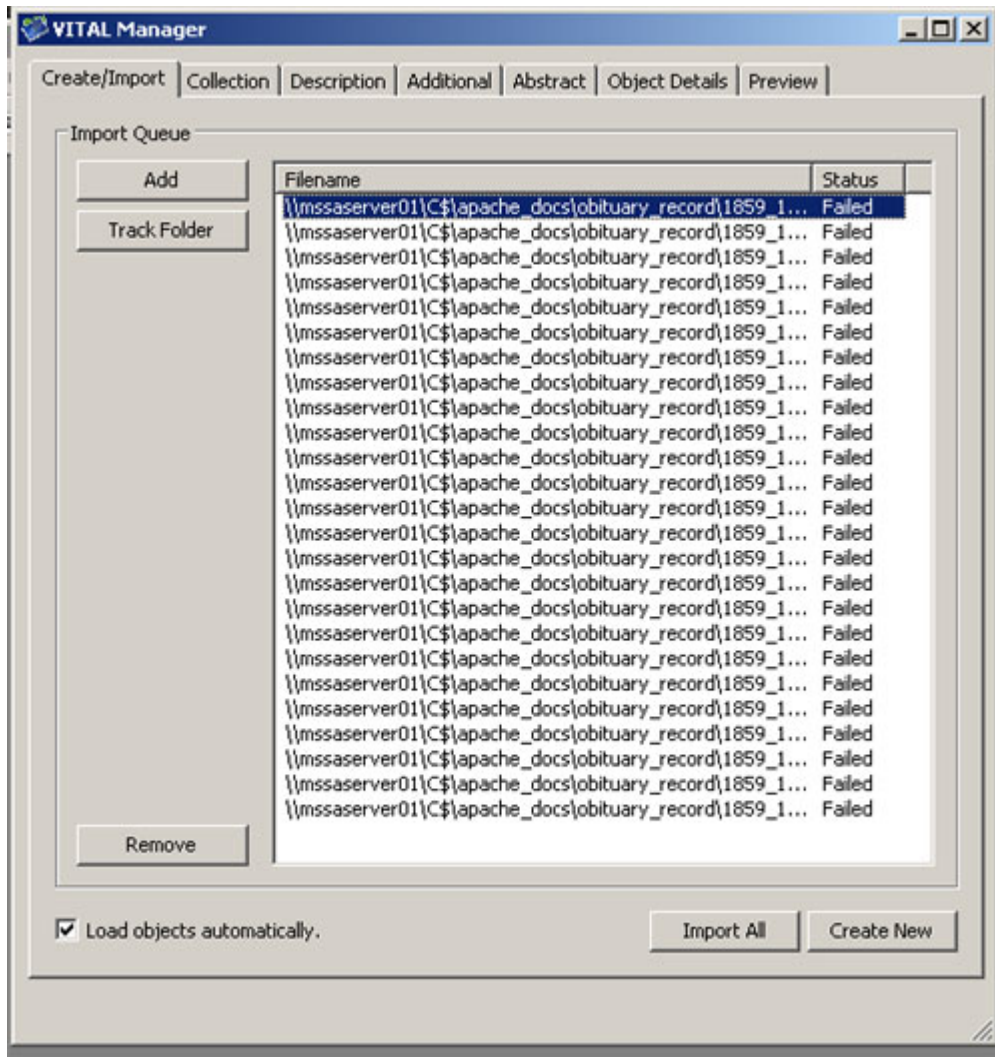
7. VS creates analyzable log of RA's web access.

(David, I didn't have time to examine this log to determine whether the log is there and is "analyzable" - Kevin did indicate that he and Neil would take a look at this since we are sharing the repository instance)

8. CB analyzes web access log and interviews RA to determine levels of success for user-interface and indexing.

Anecdotal responses to the user-interface and indexing reveal that the lack of granularity in the search results is underwhelming to say the least. The search result for a text string in a 50 page pdf file points simply to that pdf file. The user must then download the file and perform the search again within Reader to actually get to the result. This not only feels like unnecessary duplication of effort, but adds a significant enough amount of time to each search, that many users would undoubtedly be limited in their ability to fully search the resource.

I was not able to follow through on the other use case that I proposed, Access Control - video testimony. This use case was predicated mostly on capabilities that were attributed to VITAL in the training session in September, but that turned out not to be quite ready or existing. The hope was that this use case would allow collection level access control with a high level of security and a reasonable amount of flexibility in assigning varying levels of access to different users and to different locations. I still think this would be a very interesting and potentially quite useful exploration, it just became apparent that VITAL was not ready for it.





Given the large number of email messages to be ingested, using the Vital Manager is not practical. The Vital Batch Tool was used and tested. The tool includes a batch ingesting program, import.pl, and configuration file and model file for every object file to be loaded.

### **Issues/Gaps encountered in collection building**

File structure (of message objects) is not supported by the tested version of Vital and Fedora, not sure if it will be possible with the future releases. The access portal or Web view does not show related messages together. But it is possible to extract such information from the file source paths and create a tree view.

### **Problems encounter in collection building**

The batch loading program requires a configuration file and model file pair for each object to be ingested. And the batch tool does not provide a program to generate such pair automatically for each object file.

### **Enhancements that would improve collection building functionality**

More flexible configuration and model file definition and importing programs would definitely improve the collection building functionality

### **Examples from Collection Building**

<http://moonpie.cis.yale.edu:8888/access3>

The following document summarizes the findings of the collection builder and advisors as they established a collection of social science numeric datasets using VITAL/FEDORA as a part of the Yale IAC VITAL/FEDORA digital repository software project. The aim of the project as a whole is to determine the suitability of the VITAL/FEDORA product for the needs of the Yale Library and, potentially, for larger university needs. The dataset group developed a set of use cases and created a small test collection using the VITAL manager and the batch tool for the purposes of this project. The datasets collection VITAL use cases outline the basic functionality required for ingesting, indexing, searching, managing and manipulating numeric data collections in the context of a digital repository. The test collections associated with this pilot are typical examples of records and digital files held in the Yale Social Science Data Archive and searched and accessed through the Yale statistical data catalog, Statcat.

The summary narrative supports two main conclusions:

1) Considering the significant customization that is required to ingest, index, display and provide access to numeric data collections, **the dataset collection builders do not recommend using the VITAL software<sup>1</sup> to manage this collection.** Instead, we recommend assembling this collection in native FEDORA.

*a.* Use of VITAL by collections that involve ingest of multiple file formats with multi-faceted relationships and with multiple metadata formats will require additional programming and tool development to establish an ingest workflow appropriate for a-typical collections or for additional repository services.

**2) The main challenge for the datasets group in the adoption of VITAL would be to understand the extent to which the vendor will support ongoing custom programming and tool development required to establish, maintain and build services for a numeric data repository.**

This summary narrative begins with a brief collection description; outlines typical characteristics of the collection including file formats, metadata, and other records issues characteristic of this numeric data collection; presents and discusses a matrix of use cases that demonstrate the relationship between user functionality and ingest requirements; describes the ingest workflow; and finally lists the gaps we discovered when utilizing the VITAL manager and the batch tool. Together the process of generating use cases, building a test collection, and interacting with and documenting our experience with using the VITAL tools led us to the conclusions we outline above.

### **Collection description**

<sup>1</sup> The datasets group was able to evaluate VITAL Manager, its indexing and interface customization functionality, the VITAL batch ingest tool and the underlying FEDORA database. We did not evaluate the self-submission tool nor other functionalities such as cross-collection searching.

The Social Science Data Archive (SSDA) is the repository and reference center at Yale for machine-readable data sources in the social sciences. The SSDA owns and maintains a major collection of data from academic surveys, public opinion surveys, government agencies, international organizations, and related groups. SSDA codebooks and reference services are available through the Social Science Libraries and Information Services; the Social Science Statistical Laboratory provides technical assistance for dataset users. The SSDA holdings are restricted to use by the Yale community.

StatCat is Yale's statistical data catalog. It includes information about numeric datasets in the Yale Social Science Data Archive, data available to the Yale community from ICPSR, selected datasets available in the Yale University Library, and selected data sources on the Internet. Entries in the current SQL database represent an abbreviated version of the social science data metadata standard, [DDI<sup>2</sup>](#).

Statcat has several types of record entries:

- Metadata records describing the social science data archive SSDA. These records have multiple files associated: codebooks in .pdf or text format, ascii text data files, and set-up scripts for proprietary statistical software. These files are the core content of the electronic data archive.
- Harvested DDI records from the ICPSR data catalog which use the DDI metadata schema in XML format
- Internet or other external references to additional library sources. These records include links or call numbers that point to cd locations in the physical library collection.

### *Use cases and Ingest Functionality Matrix and Discussion*

*Collection builders developed use cases to capture and isolate standard repository functionalities that support the data researcher’s information seeking activities. We broke the basic repository functionality down into sets of use cases that address the following broad categories: search, record display, file access, download files, and assess data. This set of “user experience” oriented use cases represents only the most fundamental user needs for obtaining information about a numeric data collection and for downloading and obtaining actual data files for analysis. These use cases were described in use case templates and posted on the Sakai project site (see Appendix A: use cases). A complimentary set of machine/ingest tasks must inform the collection-building process to ensure that the content and structure of the resulting repository meets the criteria specified in the various user experience use cases. Though the machine ingest tasks were not fully explicated using the use case templates on the Sakai site, the machine ingest use cases are outlined as a part of Appendix B: Use Case Matrix. Use cases that were tested during the project period are underlined in the matrix.*

<sup>2</sup> See <http://www.icpsr.umich.edu/DDI/users/dtd/index.html>

In addition, collection builders created a map to facilitate indexing of the DDI metadata to enable searching within the VITAL access portal. **Appendix C: Indexing** lists the DDI fields that were selected for additional indexing beyond the Dublin core set. These fields were identified based on the advanced search parameters available on the ICPSR catalog search interface. A test configuration of the VITAL index tool to add index terms was attempted during the project period: searching on the newly indexed terms proved successful. However the complete set was not indexed and searched upon in the VITAL access portal. Collection builders directed the bulk of effort toward undertaking and

evaluating the ingest process.

### Ingest Workflow

To ready for the import of both descriptive metadata and digital files either using the VITAL manager or the batch, the records from the current SQL database were processed in the following ways:

- DDI XML was generated from the database entries.
- In the case of ICPSR records, we retrieved and incorporated the original DDI records from the ICPSR harvest. Under the current system, the ICPSR records are harvested periodically and then the DDI XML is used as a basis for generating the SQL database entries.
- Once DDI records existed for the studies used for the test collection, Dublin Core records were generated, using an XSL transform.
- Finally, to test the VITAL batch tool and to integrate the DDI, Dublin core, and file information, custom scripts were used to generate FOXML (native FEDORA file format) to pull all the pieces together and facilitate ingest.

Four ingest scenarios were explored:

- (1) Import of SSDA datasets using the VITAL manager
- (2) Import ICPSR DDI metadata via the VITAL command-line batch-processing tool
- (3) Import of SSDA datasets via the VITAL command-line batch-processing tool
- (4) Import of SSDA datasets and ICPSR DDI metadata directly into the underlying Fedora implementation

In all scenarios involving SSDA data the import required the generation of DDI XML metadata from data contained in the StatCat database. All four scenarios involved the creation of a Dublin Core XML record, which was generated via an XSL transform of the DDI XML metadata. Attempts to ingest data produced the following results:

- (1) The import of SSDA records via the VITAL manager exposed some flaws in the ingest workflow. See "Workflow Gaps using VITAL Manager and batch tool" below for further details.
- (2) Import of ICPSR DDI metadata via the command-line batch tool was successful. The resulting VITAL/Fedora objects consisted of a Dublin Core datastream and a DDI datastream. No data files were associated with these objects.
- (3) Import of SSDA records via the command-line batch tool proved impossible. SSDA datasets contain an arbitrary set of data files and codebooks of various file types. The batch ingest process is not sufficiently flexible to accommodate ingest of these datasets.
- (4) Import of SSDA datasets and ICPSR DDI metadata into the Fedora repository necessitated the creation of a FOXML (i.e., Fedora ingest format) file for each record. This approach required custom programming, but allowed for batch import of the

complex SSDA datasets into the repository.

### Workflow gaps using VITAL Manager and the batch tool

In general, we found importing with both the manager and the batch to be straightforward. However we experienced several shortcomings:

- Auto generated Dublin Core -- In VITAL manager, one cannot use externally created Dublin Core records to generate the information in Dublin Core record that is automatically generated within VITAL upon ingest. The Dublin Core records that are auto-generated contain only the title of the resource. However, when we utilized the batch tool and imported FOXML, we were able to retain, display, and search on all of the information contained in the externally generated DC.
  - We discovered a workaround in the functionality of the VITAL manager. Once a record is imported, it is possible to invoke a “regenerate Dublin Core” function within the VITAL manger. This command allows for the internally generated Dublin Core to be populated with the information contained in an imported DC datastream.
- Metadata structure – We must fine-tune the structure of our metadata prior to ingest. A main example of this modification includes determining where to place the external URLs for the harvested records and for the Internet resources. Ideally, those links should appear in both the DDI and the DC metadata with the understanding that the DC records would form the basis for cross-collection searching and for OAI harvesting from the repository to other aggregators.
- Batch tool -- Out of the box, the batch import function will only really work routinely with the ICPSR harvested records and not with the SSDA records that have actual files associated with them that must be brought into the repository. The numbers of files associated with any one SSDA record as well as the file formats do not occur in any regular pattern. Making the batch import of SSDA records as well as their associated files work properly will require additional programming/scripting.
- Mime type assignment -- Considering the VITAL manager functionality “out of the box”, we experienced issues with mime type assignments because our formats did not have standard definitions in the existing system.
  - Many of the associated files for the SSDA numeric data collection have unusual formats not already available through the VITAL manager.
  - The files in SSDA have variable numbers of files and variable kinds of file formats, rendering it difficult to use the existing batch utility.

An overall assessment of the dataset ingest workflow reveals that regardless of the repository software we choose, initial migration will require making custom adjustments to the collection’s metadata and undertaking additional programming to facilitate the

ingest of the multiple file formats and the irregular patterns associated with the data files and their study records. It is clear that an ‘out of the box’ repository software will not provide all the tools necessary to migrate and manage the Statcat/SSDA collection at first. Once the initial migration is achieved, however, routine operations such as the use of the VITAL manager to add occasional records manually or employing the batch tool to refresh the harvested records from ICPSR could be feasible. However, the dataset group is concerned about the extent to which the VITAL tools will serve as a suitable base for developing additional repository services such as those described in the “assess data” use cases. The need for customized programming to support initial ingest coupled with a concern about the extensibility of the VITAL software to provide future customized services led the dataset group to conclude that managing the numeric data collection using the VITAL software is not recommended. In addition, this group suggests that adoption of the VITAL software by the Yale library should be contingent upon further exploration with the vendor to determine what level of support for ongoing development will be available to Yale collection builders.

## VITAL Project Collection Building Experience for EAD Finding Aids

### Background and Goals:

The Yale University Library has had a Finding Aids database and Web interface for some time now. A recently formed task force was charged with defining functional requirements for a new Finding Aids database/repository system. Because this task force completed its functional requirements list just before the start of the VITAL project, there have been two goals for Finding Aids Collection Building in VITAL/Fedora:

- First, to test the features and functions offered by VTLS’ VITAL software and to evaluate the value they add in building a new repository/database of EAD Finding Aids.
- Second, to determine how well a Fedora repository, with additional VITAL user interfaces and tools, could meet YUL’s requirements for a new Finding Aids system.

Use cases for the EAD Finding Aids collection in VITAL were written to include many of the functional requirements defined by the task force as ‘essential’ or ‘highly desirable’. The Finding Aid use cases can be described this way:

- Two are related to ingesting EAD/XML files into the repository.
- Two involve indexing, search options, and results sets.
- One is about presentation, ‘EAD-specific-navigation’.
- Two were related to communication with other systems, via OAI access over the Web.

## VITAL Features Tested:

EAD Finding Aids files were successfully ingested with the VITAL 'Batch Ingest' tool and with the VITAL Manager client. Each file was found in search results a few minutes after it was added to the repository. In addition, a very brief DC record was created for each EAD/XML file ingested. Although it did not seem to be possible to change the VITAL configuration to add additional (expanded DC) fields to this brief record at ingest time, later use of the 'Regenerate Dublin Core Metadata' feature worked to add to the DC record.

The VITAL Access Portal was used to search and retrieve the Finding Aids added to the repository. It was found that each EAD file was fully indexed and searchable shortly after it was added to the repository. (The VITAL re-index process runs every two minutes, to handle recent additions or changes to the repository.) By default, VITAL includes a full text index as well as specific indexes on titles (for EAD documents, titleproper and subtitle), EAD head, EAD abstract, and file name, creator, and subjects from the DC record.

The VITAL Access Portal Administrative interface was used to view indexing options and to create new indexes on specific EAD tags. Indexes were successfully added for /ead/archdesc/scopecontent and /ead/eadheader/filedesc/publicationstmt/publisher. With changes to the configuration of the Access Portal configuration (slightly modifying the 'skin' and the 'locale'), these indexes were made available as 'Repository' and 'Scope and Content' in the dropdown menus under 'Advanced Search'.

## Issues and Gaps:

The style and presentation of Finding Aids within the VITAL Access Portal and the lack of desired Finding Aid navigation tools in the final display were the biggest problems observed by the members of the Finding Aids Requirements Task Force (the advisory group for this collection building effort). VITAL does provide the structure and necessary XSL stylesheets for EAD-to-HTML conversion 'on the fly'. With a VITAL/Fedora repository, it would no longer be necessary to create and maintain both an EAD/XML and HTML version of each Finding Aid document. However the ead2html.xsl stylesheet included, by default, with VITAL is very basic and does not provide for an attractive or functional presentation in the Access Portal. Considerable additional effort would be required to determine which required presentation and navigation features can be provided by rewriting the ead2html.xsl stylesheet and if there are some advanced Finding Aid navigation features that would require programming efforts in addition to XSL coding.

Multiple distinct collections w/in a single repository are not possible in this release of VITAL. This could be a problem for a Finding Aids repository because the different collections (BRBL, Divinity, MSS/A, etc.) would probably need to each have their own stylesheets, as they do now.

## VITAL Project: Medical Images

The Medical Library currently has 3 image collections consisting of about 2,000 images, tiffs with derivative jpegs, and metadata implemented in Greenstone. The images are cataloged with qualified Dublin Core and can be accessed through an attractive and functional user interface.

I had 3 objectives for the VITAL project:

1. Determine the feasibility of migrating our existing metadata and images to VITAL/Fedora using the batch ingest function.
2. Determine if the Access Portal could provide an acceptable search interface for our collections
3. Evaluate the web submission tool as a way to manage our workflow.

### **Batch Ingest**

I selected 5 images and metadata records from our Peter Parker Collection for the test. The VTLIS documentation for batch ingest was very basic and incomplete. I had to spend a significant amount of time trying to understand the process and configuration files by studying the examples VTLIS provided and modifying them to work with my data. I had to significantly modify the format of my metadata in order for it to be ingested. As I only had 5 records, I did this by hand using a text editor. If we are to proceed with a complete conversion, I would write a script to convert the rest of our metadata files to a format that could be ingested into VITAL/Fedora.

I wanted to put my metadata into the VITAL DC datastream. Even after the format of the data passed muster, I discovered that VITAL only accepted my DC elements that were unqualified, the title, identifier, type and rights; it ignored all of my qualified fields.

I then took a different approach. I defined another datastream for my metadata; I called it QDC. Though the ingest process was successful, the contents of the QDC datastream were not visible in the Access Portal. I had to hunt down and modify the stylesheet that the AP was trying to use to display the data as well as the language file that defined the field labels. Once this was done, I could see my metadata in the QDC datastream.

There is a function in the VITAL Manager called “Regenerate Dublin Core”. I tried it on one of my records and now my metadata is available to the user without having to click on the QDC datastream link. The use of the regenerate DC data function makes the metadata more easily accessible but it is not feasible to use this function on the 2,000 records we currently have as it must be applied separately to each individual object.

### **Access Portal**

As I just mentioned, I had to modify stylesheet and language files used by the Access Portal (AP) in order to display my metadata. In order for the AP to be useable, I had to index my metadata fields. After some project and error, due to my own lack of knowledge of the structure of XPATHs and VTLs' lack of documentation on the subject, I was able to add my metadata to existing indexes by adding XPATH statements to the index definitions. I worked primarily on the subject index.

The AP does not support thumbnail preview of tiff images. If we wanted to have a preview available in the AP, we would have to use jpeg images instead of or in addition to tiffs.

Although the AP provides reasonable search options, it does not allow the browsing of the collection by any particular fields other than title without doing a search. There is no way to browse a particular index to see what might be available. These are functions that are built into Greenstone and are relatively easy to expand to fields other than those natively supported by the out of the box code.

In short, I believe that the AP is not only aesthetically displeasing without significant modification but is also limited in its functionality. Though its appearance can be changed, its functionality cannot.

### **Web Submission**

This software was not available to us until very late in the project. We only had access to the open source version of the software. I had hoped to have time to use it to create a workflow for our processes but I had none. I am unable to make any judgment on its appropriateness to our needs. As it is open source code and we are looking for software to manage our workflow, we will probably look at its functionality and suitability to our needs. I will also look at the feasibility of modifying the final stage, submission to the repository, for use with Greenstone instead of Fedora.

## **Collection Builder's report: Unicode text**

### **Collection:**

Project AMEEL, a 4-year granted project funded by the Department of Education and Yale, plans to build a new digital repository holding scanned images with OCR output from Arabic printed texts, beginning first with journals from Iraq and later from the rest of the Middle East. Project AMEEL is a ground-breaking project focusing on

solving Arabic OCR challenges and on providing transparent cross-collections searching. Its repository will need to connect remotely to other repositories of existing digital content held by universities and libraries, as well as commercial entities such as Brill Publishers, in Leiden, The Netherlands.

The VTLS / Fedora experiment coincided with the selection of a repository solution for Project AMEEL. The functionality of VTLS / Fedora needs to be Unicode-compliant as most of the digital objects held in the new AMEEL repository will contain Unicode text in indexed fields and will need to display both Western languages and Arabic using the correct encoding and LTR or RTL orientation via the Internet.

## **Use Cases:**

At the time of writing the use cases for this collection, ingest of data via batch and review of batch data were the top goals. However, due to lack of documentation the focus changed to ingesting digital objects via the Vital Manager. Therefore, all ingest and review was completed manually.

## **Best Features:**

The Vital Manager functioned well. There were no crashes and the client caused no problems with the operating system of the workstation on which it was installed. For Western text, the search function performed correctly.

The Vital Manager and Vital Access displayed Arabic text correctly in all fields available via the Vital Manager, e.g. title, creator, etc.

## **Problems:**

### **1. Display order in Vital Access: Full View**

Based on the manual ingest using the Vital Manager, each digital object entered had a datastream consisting of several objects, each with a unique PID and time stamp assigned by the Vital Manager. At ingest, I made sure that the datastream was entered in logical order, i.e. for a scanned article from a journal; the pages were entered in the order published in the journal. Also, when entering label names for the datastream objects, I used unique names that corresponded to the actual page number in the article scanned.

Because of the PID and data stamp along with the unique label names, I hoped for some control when displaying the datastream. However, when viewing the article and its datastream objects in the Vital Access: Full View, no consistent order was noted. Thus, page 10 displayed before page 1 as seen in this sample:

**Title** al-Qānūn fī al-tibb li-Abī `Alī al-Husayn ibn `Abd Allāh Ibn Sīnā.

**Creator** Ibn Sīnā (Avicenna, d. 428/1037)

**Description** the modern critical edition for القانون في الطب (Canon of Medicine) from Ibn Sina (Avicenna, d. 428/1037). The manuscript, copied in 1006 H./1597-98 A.D., is found in the Medical Historical Library at Yale University and is cataloged as Cushing Arabic ms. 5.

**Identifier** vital:34

**Fedora PID** vital:34

**Creation Date** 2005-11-28T15:18:31.594Z

**Modified Date** 2005-11-28T15:18:31.628Z

**Copyright Type** Other Enforced Copyright

**Copyright URL** <http://www.library.yale.edu/htmldocs/copyright.html>

Documents			
Label	Date	Format	Download
p10_JPEG 2000 Image	November 28, 3:47 pm		Download

After noting this, I tried to control the order of the display by clicking OFF the “Publish datastream for public viewing” and relying on the Document Navigator.

(Also, please note in the example above: some customization is needed to turn off the “Download” option dependent on the Intellectual Property permissions acquired for each object. In the case of Project AMEEL, this could be at the title or the article level for each journal.)

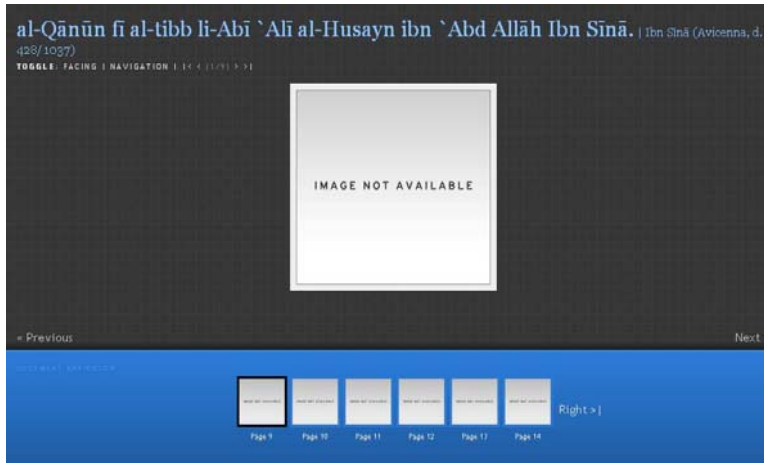
## 2. Formats handled by Document Navigator.

In my testing, I created digital objects whose datastreams were: JPEG2000, JPEG, PDF, and TIFF.

The images saved as JPEG were displayed correctly based on the XML document created to control the navigation. While the navigation for JPEG images worked well, the Document Navigator does not include any highlighting feature so that the patron can easily find the desired string within the search results.

The Document Navigator did not display JPEG2000 images (as seen in the example below) but the Navigator did recognize the existence of the XML document for navigation. TIFFs were not displayed in the Vital Access: Full View.

Since we hope to use PDFs in the workflow for Project AMEEL, I created a digital object with PDFs in the datastream and an XML document to control navigation. This did not work. As with the JPEG2000 object, the Document Navigator opened and navigated from one datastream object to another but did not display the PDF.



### 3. Assumption: PDFs are always multi-page.

From the VITAL Digital Management System User's Guide, labeled version 2.0 and dated October 31, 2005.

#### **5.4.3 Importing Multipage, PDF Documents**

To view multipage documents that are in PDF using the Document Navigator, you do not have to import the PDF document any differently than you would any other type of object. Because VITAL handles the paging of PDF documents transparently, a PDF document does not have to be imported along with a structural XML file. In the VITAL Access Portal, once you access the Full View screen for the PDF document, you will only need to click the View link, and the document will open automatically in the Document Navigator.

In the case of Project AMEEL, we will create new digital objects in PDF-with-text-behind format. In order to provide quick throughput via the Internet, each page scanned will be a single PDF. As explained above, the Document Navigator did not display the PDF.

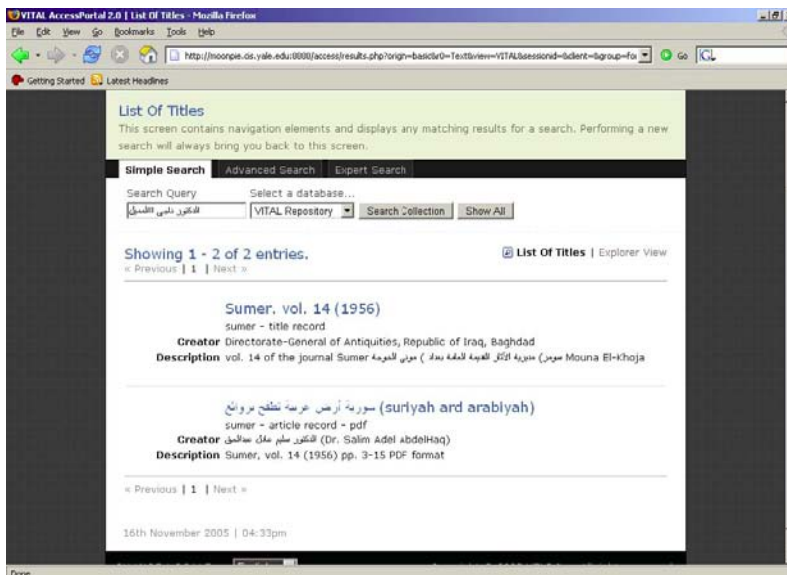
From the Vital Access: Full View, the single PDF could be selected and displayed. But since Vital assumes that PDFs are always multi-page, the display showed “Previous” and “Next” links that did not navigate anywhere.



While the presentation of the Document Navigator is quite polished and attractive, it does not recognize or display inherent PDF functionality, e.g. search, print, zoom, etc.

#### 4. Searching in Arabic

Inconsistent results: Arabic text for indexing and searching was added to particular digital objects. When searching via Vital Access, search results were inconsistent. For the most part, if one word in Arabic was entered into the Search Query box, the results were good. However, if two words or more were entered, the results were incorrect. This is most likely because of how Vital is truncating the search string in conjunction with the RTL orientation of an Arabic string.



#### 5. Batch ingest question: how to handle Arabic, Unicode, and encoding

It was not clear where to ingest the text for indexing and searching, when this text is in Arabic. Or, in which format. See below: examples of Arabic in encoded format and in Unicode character format entered into the Abstract field via the Vital Manager. Text strings from the passages in the examples below were used in search queries and did not produce consistent or correct search results.

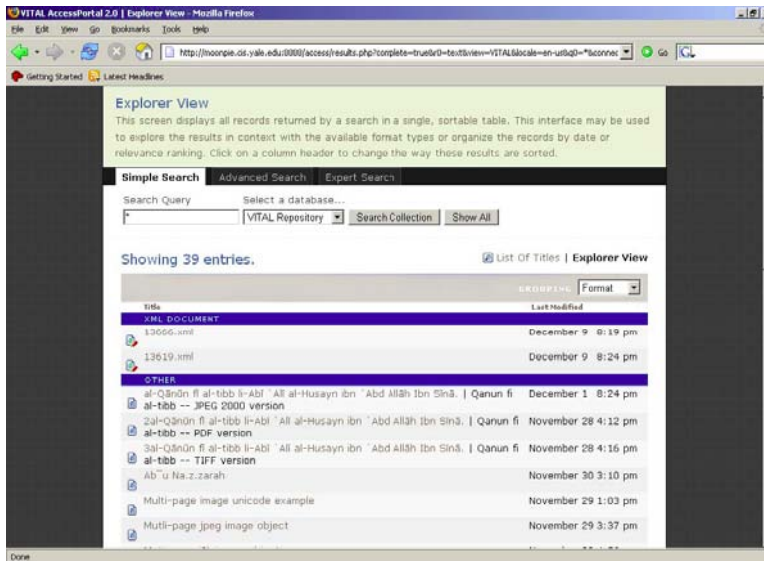
Since the OCR step in the digitization workflow for Project AMEEL may produce files with Arabic text in either format in the examples below, it is important that these files can be part of an ingest procedure. It is further extremely important to create custom disseminators for this type of OCR output. For example, it would be of tremendous help to create a disseminator that would display OCR output as a separate and displayable digital object or as an object in a datastream.



## 6. Usability

The documentation received at the end of October did not include any explanation to help improve the usability of the display in Vital Access. For example, in the example below, the Explorer View offers some flexibility in sorting the collection. Yet, there was no explanation to understand why XML documents received higher priority or why all other digital objects were labeled merely as “Other”.

Also, in the specific case of transliterated titles for non-Western scripts or for vernacular scripts, an extra field in the digital object description is required to sort titles correctly. In the example below, “al-Qanūn” (starting with ‘Q’) should sort after “Multi-page image” (starting with ‘M’).



## Conclusions:

1. VTLS / Fedora as it exists at the time of this writing does not offer enough features to be the front-end client of choice for Project AMEEL.
2. The Vital Manager worked well and suggests that with more time for experimentation that batch ingest would be possible for adding new digital objects into a Fedora repository for Project AMEEL.
3. The interface of Vital Access both in Full and Explorer views needs more improvement and the ability to customize disseminators than currently possible to meet the needs of Project AMEEL.
4. Native Fedora remains a potentially good solution for Project AMEEL's new repository.
5. A hybrid model for Yale University Library projects seems most appropriate for the future. VTLS / Fedora would serve small projects where English or Western languages are used. For the purposes of Project AMEEL, however, we hope to proceed with a customized front-end using native Fedora as the repository and the middleware to permit connectivity with other remote collections.

Submitted on: January 16, 2006

Submitted by: Elizabeth Beaudin

### III. Report on the Collection Building Experience: YIPP The Goals and Outcome of use case(s)

- Load Images of the original papers for subsequent indexing and transcription – Successful
- Locate original document images based on the accession numbers recorded in the project index documents – Successful

- Discover and retrieve information about Connecticut tribes based upon a subject and tribe – Successful
- Locate original document images based on the text of the transcript – Successful
- Navigate from original transcript to annotation using internal deep links to Fedora – Successful
- Navigate from external Project Documentation to Collection content using external deep links to Fedora – Successful

### **Features/ VITAL Function Matrix tested in use case**

- VITAL Manager
  - Load text and images
  - Automatic text indexing
  - Automatic Metadata creation (dates and subjects)  
DC and FOXML
  - Incremental Object datastream compilation
  - Integrated datastream editing
  - Integrated XML editing
- VITAL Access Portal
  - Access control via Tomcat
  - Automatic recognition and display based on appropriate mime type
  - Multipage Document Viewing  
PDF  
JPG  
JPEG 2000
  - Simple and advanced searching
- VALET Document Submission
  - Workflow management to prepare content
  - Background loading from desktop content
  - Automatic datastream packaging
  - Automatic Metadata creation (author, subject, date)
  - Integrated rights management

### **Issues/Gaps encounter in collection building**

- The collection is too small to justify and independent repository, need a method to manage content on small scale.
- Lack of control over presentation sequence makes discovery experience confusing .
- Lack of heirarchy ("part of") limits documentation of important relationships.
- Lack of control over Datastream PID makes multipart document annotation difficult.

## **Problems encounter in collection building**

- Many functions not well documented.
- Late system availability limited dialog with content owners
- VALET document management has no incremental build capability

## **Enhancements that would improve collection building functionality**

- Need a means to prevent indexing of certain datastreams

## **Examples from Collection Building**

This document shows that individual [datastreams](#) within a fedora object can be referenced directly, even from documents not themselves included in the repository. This is a reference to an annotation concerning [Cossatuck](#). Here is a [map](#) of the area.

Here is a screen shot:

YIPP	December 14, 2005, 4:41 pm	XML Document	<a href="#">Download</a>
Index of Documents.txt	December 14, 2005, 4:41 pm	Text Document	<a href="#">Download</a>
HTML Document	December 14, 2005, 4:59 pm	HTML	<a href="#">Download</a>
CSL1669.7.8b.JPG	December 14, 2005, 4:41 pm		<a href="#">Download</a>
CSL1669.7.8c.JPG	December 14, 2005, 4:41 pm		<a href="#">Download</a>
CSL1669.7.8a 2.JPG	December 14, 2005, 4:41 pm		<a href="#">Download</a>
<a href="#">Multipage View (View)</a>	December 14, 2005, 4:51 pm	Datastream	<a href="#">Download</a>
1IP1 10 scholars.doc	December 14, 2005, 4:41 pm	Datastream	<a href="#">Download</a>

