



The Mile High City

DENVER

ellucian
LIVE | 2016

Navigating the Banner XE Technology Stack

Brian Schoudel, Dir Application Administration

John Tanner, Enterprise App Admin Specialist

University of Illinois

April 19, 2016

Session 36090

Session rules of etiquette



- Please turn off your cell phone/pager
- If you must leave the session early, please do so as discreetly as possible
- Please avoid side conversation during the session

Thank you for your cooperation!

Agenda



- 1** Past/Present/Future with Banner XE

- 2** Banner XE architecture

- 3** Banner XE modification process

- 4** Tips for Troubleshooting XE

- 5** Summary

ellucian
LIVE | 2016



Past/Present/Future using XE



University of Illinois



- Three campuses (Urbana-Champaign, Chicago, Springfield) www.uillinois.edu
- Urbana-Champaign – 44K students
- Chicago – 28K students
- Springfield – 5 K students
- **Production Banner database 1.6 TB**
- **5.8 million spriden recs**



Current Banner Prod Environment



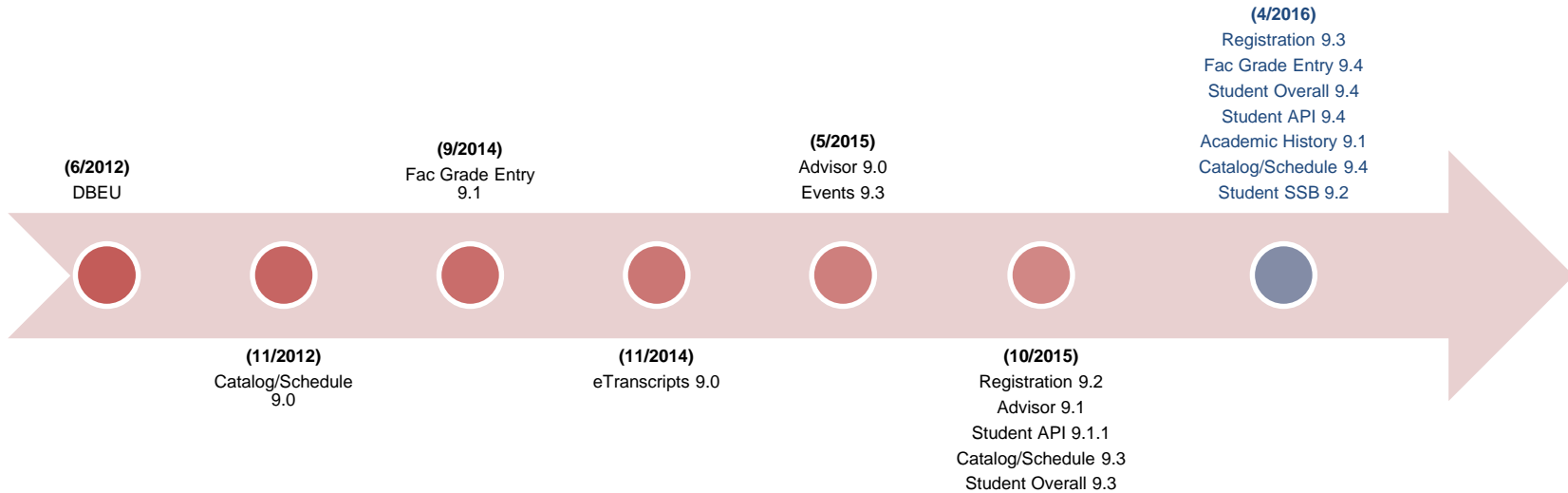
- **Been live since 2001**
- **Licensed for everything but Alumni**
- **Banner 8 releases kept up to date (Gen 8.8 this month)**
- **OFM 11g for app servers on RHEL 6**
- **Oracle 11.2.0.4 Sun M9000 for database server**
- **Multi-Campus VPD for Fin Aid, Academic Hist, Web Tailor, Course Catalog and Schedule**
- **FGAC used throughout Student and HR to segregate by campus, college, dept, access, etc..**

Current XE prod environment

- Database Extension Util (DBEU)
- Student Faculty Grade Entry 9.3 (Active)
- Student Course Catalog 9.3
- Student Class Schedule 9.3
- Student Advisor Self Service 9.1
- Student Advisor 9.1
- Student API 9.1.1
- Student Overall 9.3
- Student Registration 9.2 (Limited)
- Student eTranscript 9.0
- General Events Mgmt 9.3 (Active)
- Banner DB Upgrade 9.1



Banner XE timeline



Other apps in test/beta...



- **Ellucian Solution Manager 1.5**
- **Application Navigator 1.3.1**
- **Banner Finance Transformed Forms 9.0 (Beta)**
- **Banner HR/Payroll Transformed Forms 9.0**
- **BEIS SSO Manager**
- **Ellucian Identity Services (SAML authentication)**
- **Ellucian Integration Hub (Beta)**

Miscellaneous Notes on XE



- **Need to stay current with Banner 8**
- **It is modular but beware of dependencies**
- **You can implement only the apps you want (can also continue to use Banner 8 and ease into it)**
- **Been slower to adopt than we initially thought.**
- **Really version 1.0, 1.1, ... (not 9.0, 9.1, ...) of a new application (should get better over time)**

ellucian
LIVE | 2016



Banner XE Architecture



Banner XE Architecture



- **Why WebLogic?**
 - Oracle Site License
 - 24X7 Oracle Support
 - WebLogic Familiarity (Existing Forms/Self-Service Environment)

Banner XE Architecture

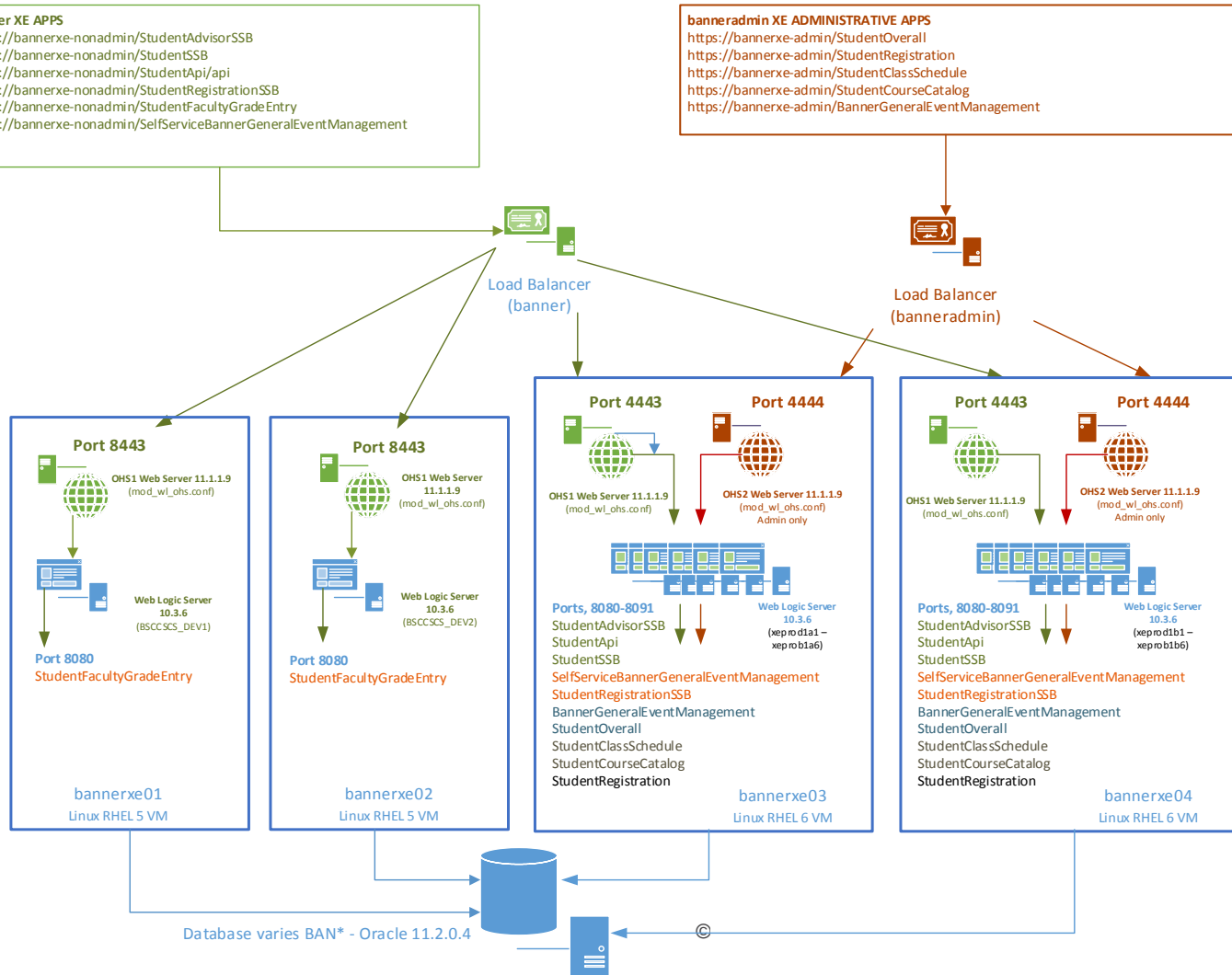
- **Environment Setup**

- Four Red Hat Enterprise 6.7 Linux Servers
 - Duplicate Hardware setup in Development, QA, and Production
 - 64 GB RAM
 - 16 CPU's
- WebLogic Server 10.3.6.0.13
- Oracle HTTP Server 11.1.1.9.0 (2 OHS instances per server)
- Server Iron Hardware Load Balancer



banner XE APPS
 https://bannerxe-nonadmin/StudentAdvisorSSB
 https://bannerxe-nonadmin/StudentSSB
 https://bannerxe-nonadmin/StudentApi/api
 https://bannerxe-nonadmin/StudentRegistrationSSB
 https://bannerxe-nonadmin/StudentFacultyGradeEntry
 https://bannerxe-nonadmin/SelfServiceBannerGeneralEventManagement

banneradmin XE ADMINISTRATIVE APPS
 https://bannerxe-admin/StudentOverall
 https://bannerxe-admin/StudentRegistration
 https://bannerxe-admin/StudentClassSchedule
 https://bannerxe-admin/StudentCourseCatalog
 https://bannerxe-admin/BannerGeneralEventManagement



Database varies BAN* - Oracle 11.2.0.4

Banner XE Architecture

- **Web Tier Setup**

- OHS1 used for all non-administrative applications.
No firewall restrictions.
- OHS2 used for all administrative applications.
Firewalled to campus users.
- Multiple Development/Test environments running simultaneously.
- Banner instances differentiated with -SIDNAME
- Web Tier mod_wl_ohs.conf used to route application to specific WebLogic managed servers.



Banner XE Architecture

- **Managing Banner databases**

- Modification configuration files to maintain –SIDNAME
- XE
 - instance.properties (shared_configuration directory)
 - XEAPP_configuration.groovy (application logging)
 - weblogic.xml (extracted from war)
 - web.xml (extracted from war or manually edit CreateWar.groovy to update self-service datasource.)



Banner XE Architecture

- **Managing Banner databases (continued)**

- Modification configuration files to maintain –SIDNAME

- WebLogic

- mod_wl_ohs.conf

```
<Location /BannerGeneralEventManager-BANDEV>  
    SetHandler weblogic-handler  
    WebLogicCluster managedserver01:8088,managedserver02:8089  
</Location>
```

- ssl.conf

```
RewriteRule ^/BannerEventManager-BANDEV/login/auth/* https://authentication-provider [R]
```



Banner XE Architecture



- **Issues managing multiple Banner databases**
 - Configuration file management.
 - Creation of scripts to centrally manage and maintain mod_wl_ohs.conf and Banner instances between application servers.
 - “Banner Already Open in Another Window” error.
 - Manual extraction and editing if WEB-INF/weblogic.xml cookie-path.
 - Managed server application deployment tracking.
 - WebLogic scripts export server status and imported to website via php.

Banner XE Architecture



- **Issues managing multiple Banner databases (continued)**
 - Managed server application deployment tracking.
 - WebLogic scripts export server status and imported to website via php.

Managed Server Section

bannerxeserver

Server Name	Server State	Server Listen Address	Server Listen Port	Server Health Status
AdminServer	RUNNING	bannerxeserver/ipaddress	7001	Component:ServerRuntime State:HEALTH_OK MBean:AdminServer ReasonCode:[]
xedev1a_1	RUNNING	bannerxeserver/ipaddress	8080	Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_1 ReasonCode:[]
xedev1a_2	RUNNING	bannerxeserver/ipaddress	8082	Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_2 ReasonCode:[]
xedev1a_3	RUNNING	bannerxeserver/ipaddress	8084	Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_3 ReasonCode:[]
xedev1a_4	RUNNING	bannerxeserver/ipaddress	8086	Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_4 ReasonCode:[]

Banner XE Architecture



- **Issues managing multiple Banner databases** (continued)
 - Managed server application deployment tracking.
 - WebLogic scripts export server status and imported to website via php.

Deployment Section

Server Name	Application	Component Type	Session Connections
xedev1a_1	Banner9DS-BANDEV	JDBCDataSourceRuntime	
xedev1a_1	Banner9DS_BANDRMS1	JDBCDataSourceRuntime	
xedev1a_1	Banner9SsbDS-BANDEV	JDBCDataSourceRuntime	
xedev1a_1	StudentAdvisorSSB-BANDEV	WebAppComponentRuntime	0
xedev1a_1	StudentApi-BANDEV	WebAppComponentRuntime	0
xedev1a_1	StudentSSB-BANDEV	WebAppComponentRuntime	0

Banner XE Architecture



- **Issues managing multiple Banner databases** (continued)
 - Managed server application deployment tracking.
 - WebLogic scripts export server status and imported to website via php.

Server Status Section

Memory Status	Currently Allocated	Currently Available	Max Allowed	Percentage Free
AdminServer	500	195	500	39
xedev1a_1	4776	2597	7282	70
xedev1a_2	4749	1793	7282	59
xedev1a_3	4926	3616	7282	82
xedev1a_4	4868	2988	7282	74

Banner XE Architecture

- Capacity planning – Future configuration



Forms Servers

4 load balanced Red Hat 5 Servers

2 CPU's per server

64GB RAM

1 Managed WebLogic Server
per server.

512MB max allocated memory
per managed server

Self-Service Servers

4 load balanced Red Hat 5 Servers

8 CPU's per server

16GB RAM

1 WebLogic OHS Apache instance
per server.

Each OHS instance utilizes roughly
1GB of memory.

XE Servers

4 load balanced Red Hat 6 Servers
2 load balanced Red Hat 5 Servers
*initial Faculty Grade Entry implementation servers

16 CPU's per server

160GB RAM

6 Managed WebLogic Servers
per server.

8GB max allocated memory per managed
server.

2 WebLogic OHS Apache instances per server.
**Admin and Self-Service apps are split.

***We are only partially into the
Student XE product at this moment.

ellucian
LIVE | 2016



Banner XE Modifications



Banner XE Modifications

- **Illinois has many local modifications to Ellucian products such as Oracle Forms and Self-Service applications. This need is still required with XE technology.**
- **Requirements**
 - Access to source code for Application Support and Developers.
 - Processes to restrict access to source code.
 - Tools to modify and test changes.
 - Processes to document and distribute changes for technical users.



Banner XE Modifications

Git Repository Setup



○ Illinois git origin server

- Red Hat Enterprise Linux Server release 6.7
- ssh key exchanged with Ellucian to access source git repositories
- Create local master branch for modification
- Git repositories are secured with gitolite
 - ssh keys exchanged with technical team and developers to access repositories
 - Allows setting read/write access to git repositories via gitolite.conf

```
repo banner_student_registration_app
  RW+          = user1
  R            = user2
  - local-master = user3
```
- Modifications to source done via IntelliJ or git bash

Banner XE Modifications

Git Repository Process

- **Create bare repository from Ellucian source.**
 - Create single bare repository or use script to create all available repositories.
- **Create local master branch for modification.**
 - Find latest version and create local master branch
- **Update .gitmodules to point to local origin server**
- **Update project submodules.**
- **Clean the application. **This can be time consuming**



Banner XE Modifications

Git Repository Process (continued)

- Make modifications to the code
- Compile
- Run locally to test changes.
- Include readme file with specific changes for technical teams.
- Package up and deploy to WebLogic Server.





Banner XE Modifications

Modifying XE with code changes

- Simple modification, increasing length of a field from 2 to 4

- Example; Student Schedule App
- Modified
plugins/banner_general_common.git/src/groovy/net/hedtech/banner/general/overall/SectionCrossListSection.groovy
increasing length of cross list column to 4.
- @Column(name = "SSRXLST_XLST_GROUP", nullable = false, length = 2)

Changed to
- @Column(name = "SSRXLST_XLST_GROUP", nullable = false, length = 4)

Banner XE Modifications



Modifying XE with code changes

o Advanced modification, disabling Last Attend Date and Hours Attended

- Example; Student Faculty Grade Entry
- Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
- Before modification

Import Grades

Select Preview **Map** Import Finished

Map Columns

Use the drop down lists to map the data from your spreadsheet to the appropriate columns on the Faculty Grade Entry page.

* Student ID * Term * CRN are required fields. Those must be mapped in order to continue the import process.

Term Code* ✓ CRN* ✓ Student ID* ✓ Midterm Grade ✓
Last Attend Date ✓ Hours Attended ✓

Ignore	Ignore	Midterm Grade	Last Attend Date	Hours Attended
Confidential	Course	Midterm Grade	Last Attend Date	Hours Attended
	Molecular and Cell Biology	F	02/04/2016	
	Molecular and Cell Biology	C	02/04/2016	

Cancel Go Back Next

Import Grades

Select Preview **Map** Import Finished

Map Columns

Use the drop down lists to map the data from your spreadsheet to the appropriate columns on the Faculty Grade Entry page.

* Student ID * Term * CRN are required fields. Those must be mapped in order to continue the import process.

Term Code* ✓ CRN* ✓ Student ID* ✓ Midterm Grade ✓
Last Attend Date ✓ Hours Attended ✓

Ignore	Ignore	Midterm Grade	Last Attend Date	Hours Attended
Confidential	Course	Midterm Grade	Ignore	Ignore
	Molecular and Cell Biology	F		
	Molecular and Cell Biology	C		

Cancel Go Back Next

Banner XE Modifications



Modifying XE with code changes

o Advanced modification, disabling Last Attend Date and Hours Attended

- Example; Student Faculty Grade Entry
- Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
- Modified

`grails-app/controllers/net/hedtech/banner/student/utils/ImportController.groovy`

Original Code Allowed Last Attend Date and Hours Attended to be mapped when importing Midterm Grades

```
def columns = validCourseRosterColumns
if (session.selectedSection?.gradingType == "midterm") {
    columns = columns.findAll {
        it.key != "finalgrade" && it.key != "incompletefinalgrade" && it.key != "extensiondate"
    }
}
```

Banner XE Modifications



Modifying XE with code changes

Advanced modification, disabling Last Attend Date and Hours Attended

- Example; Student Faculty Grade Entry
- Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
- After modification

Import Grades

Select Preview Map Import Finished

Map Columns

Use the drop down lists to map the data from your spreadsheet to the appropriate columns on the Faculty Grade Entry page.

* Student ID * Term * CRN are required fields. Those must be mapped in order to continue the import process.

Term Code* CRN* Student ID* Midterm Grade

Ignore	Ignore	Midterm Grade	Ignore	Ignore
Confidential	Course	Midterm Grade	Last Attend Date	Hours Attended
No	Molecular and Cell Biology	F	02/04/2016	
No	Molecular and Cell Biology	C	02/04/2016	

Cancel Go Back Next

Import Grades

Select Preview Map Import Finished

Map Columns

Use the drop down lists to map the data from your spreadsheet to the appropriate columns on the Faculty Grade Entry page.

* Student ID * Term * CRN are required fields. Those must be mapped in order to continue the import process.

Term Code* CRN* Student ID* Midterm Grade

Ignore	Ignore	Midterm Grade	Ignore	Ignore
Confidential	Course	Midterm Grade	Last Attend Date	Hours Attended
No	Molecular and Cell Biology	F	02/04/2016	
No	Molecular and Cell Biology	C	02/04/2016	

Cancel Go Back Next

Banner XE Modifications



Modifying XE with code changes

o Advanced modification, disabling Last Attend Date and Hours Attended

- Example; Student Faculty Grade Entry
- Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
- Modified

`grails-app/controllers/net/hedtech/banner/student/utils/ImportController.groovy`

Modified Code Ignored Last Attend Date and Hours Attended mappings when importing Midterm Grades by adding `lastattenddate` and `hoursattended` columns to logic. Change highlighted in red.

```
def columns = validCourseRosterColumns
if (session.selectedSection?.gradingType == "midterm")
  columns = columns.findAll {
    it.key != "finalgrade" && it.key != "incompletefinalgrade" && it.key != "extensiondate" && it.key !=
"lastattenddate" && it.key != "hoursattended"
  }
```

Banner XE Modifications

Modifying XE without code changes



○ Modifying XE through CSS and js

- Include bannerSelfService-custom.css and bannerSelfService-custom.js in XE war.
- Allows for Institutional Branding, removing the Ellucian University logo in upper left of browser.

```
.institutionalBranding {  
  background: url("./images/logo.gif") no-repeat;  
}
```
- Allows hiding of columns/buttons without modifying source code. Example, Student Faculty Grade Entry

```
#browseButton {  
  visibility: hidden;  
}
```
- Using browser developer tools makes finding elements easier.

Banner XE Modifications

Modifying XE without code changes

- Modifying XE through CSS and js

Before custom css



The screenshot shows the Banner XE interface. At the top, the navigation bar includes 'ellucian UNIVERSITY' and 'Banner Self-Service'. The 'Grade Entry' tab is active. A table lists courses, with 'Molecular and Cell Biology (MCB) 298' selected. Below this, the 'Enter Grades' section is visible, showing a table with columns for 'Full Name', 'ID', 'Midterm Grade', 'Last Attend Date', and 'Hours Attended'. A search box and pagination controls are also present.

Grading Status	Subject	Course	Section	Title	Term	CRN
	Geography (GEOG)	101	AYE	Global Development&Environment	Spring 2016 - Urbana-Champaign (120161)	33251
	Statistics (STAT)	200	F58	Statistical Analysis	Spring 2016 - Urbana-Champaign (120161)	48327
	Molecular and Cell Biology (MCB)	298	B	MCB Honors Lab Discussion	Spring 2016 - Urbana-Champaign (120161)	63532

Full Name	ID	Midterm Grade	Last Attend Date	Hours Attended
		<input type="text"/>	02/16/2016	<input type="text"/>
		<input type="text"/>		<input type="text"/>
		<input type="text"/>		<input type="text"/>
		<input type="text"/>		<input type="text"/>
		<input type="text"/>		<input type="text"/>

Banner XE Modifications

Modifying XE without code changes

- Modifying XE through CSS and js

After custom css



The screenshot displays the Banner XE interface. At the top, the Illinois State University logo is visible on the left, and user information (John R. Tanner, Univ of Il at Urbana-Champaign) and navigation links (Sign Out, Notifications) are on the right. A 'Tools' button is also present.

The main content area is titled 'Select a Course' and contains a table with columns: Grading Status, Subject, Course, Section, Title, Term, and CRN. The table lists three courses:

Grading Status	Subject	Course	Section	Title	Term	CRN
<input type="checkbox"/>	Geography (GEOG)	101	AYE	Global Development&Environment	Spring 2016 - Urbana-Champaign (120161)	33251
<input type="checkbox"/>	Statistics (STAT)	200	F58	Statistical Analysis	Spring 2016 - Urbana-Champaign (120161)	48327
<input type="checkbox"/>	Molecular and Cell Biology (MCB)	298	B	MCB Honors Lab Discussion	Spring 2016 - Urbana-Champaign (120161)	63532

Below the table is a pagination control showing 'Page 1 of 1' and '5 Per Page'. The 'Records: 3' indicator is at the bottom right of the table area.

The 'Enter Grades' section below the table has a search bar 'Search By Full Name or ID' and a table with columns: Full Name, ID, and Midterm Grade. The table is currently empty. A pagination control at the bottom shows 'Page 1 of 1' and '25 Per Page', with 'Records: 5' at the bottom right.

On the right side of the interface, there is a sidebar for 'Molecular and Cell Biology 298, Section B'. It includes course details, a warning '5 Grades Remaining', and statistics: 'Eligible: 5', 'Registered: 16', and 'Graded Midterm: 0'. Course dates are '01/19/2016 - 05/04/2016'. Primary and secondary instructor fields are also present.

At the bottom of the page, there are 'Reset' and 'Save' buttons.



Banner XE Modifications

Git/Grails Challenges

- Learning and integrating new modification process into existing procedures
- Maintaining changes to application plugins that are used by multiple XE applications
- **Windows specific issues**
 - Some groovy files needed to be changed to accommodate Windows paths. “grails not found”
 - Setting fileformat=unix when editing some files
- **Local workstation resources when cleaning, compiling, and running apps locally. Need a good amount of RAM if workstation isn’t dedicated to modifications.**

ellucian
LIVE | 2016



Tips/Troubleshooting XE

Skills needed to support XE



- Existing expert knowledge of Banner 8
- Java => Groovy and Grails
- Web 2.0 (HTML5, CSS, JavaScript, AJAX, jQuery, JSON)
- Java Middle Tier (tomcat or WebLogic)
- Oracle database
- Git (source control)
- IDE (IntelliJ or Eclipse)
- Persistence and time...

Resources



- **eCommunities/BORACLE**
- **Get involved in early adopter/beta efforts (good for contacts)**
- **Literature (Grails in Action/Groovy in Action)**
- **Ellucian training on XE development**
- **Lot of free resources in terms of Git, HTML5, Javascript, JQuery and Java**
- **Groovy and Grails is a little spotty on google**

Browsers



- Works well with Chrome and Firefox (haven't tested much with Safari)
- Don't recommend IE. IE8 JavaScript was REALLY slow. Faster now but have seen issues with things not displaying properly
- Keeping in mind that Chrome and Edge (cannot run Banner 8 Forms). Firefox targeting end of the year.

Browsers for troubleshooting



- **Chrome Developer Tools are your friend!!☺** (*View-> Developer -> Developer Tools*)
- **Assists with performance, response headers, CSS manipulation, cookies, errors, etc..**
- **For Firefox there is** (*Tools -> Web Developer*)
- **Other tools include Firebug, Fiddler, SAML tracer (setting up SAML), etc...**
- **Something for every browser**

Chrome Dev Tools



1.) Click on Search Terms Drop Down

2.) Review Background AJAX calls

3.) Evaluate Performance

Name	Method	Status	Type	Initiator	Size	Time	Timeline - Start Time
81LJZQN28cUzOipQ5miDiwmAYEqwMh2kcrHgTXKwWwHG.gif	GET	200	gif	o9sPOb93Nk5ikL3D9EkXapEx79U3mRdXM5altTSE5e4.js:23	1.8 KB	33 ms	
getTerms?searchTerm=&offset=1&max=10&_=1453306169020	GET	200	xhr	o9sPOb93Nk5ikL3D9EkXapEx79U3mRdXM5altTSE5e4.js:23	1.2 KB	5.58 s	

Chrome Dev Tools cont...



The screenshot shows the Chrome DevTools interface. The browser window displays a registration page with a search box for terms. The Network tab is active, showing a list of requests. The request `getTerms?searchTerm=&offset=1&max=10&_=1453306169020` is highlighted. A red arrow points from the text '2.) Click on "Preview" tab then view payload from "getTerms" request' to the Preview tab of the selected request. The Preview tab shows a JSON array of term objects.

2.) Click on "Preview" tab then view payload from "getTerms" request

```
[{"code": "120161", "description": "Spring 2016 - Urbana-Champaign"}, {"code": "120161", "description": "Spring 2016 - Urbana-Champaign"}, {"code": "120160", "description": "Spring 2015 - Urbana-Champaign"}, {"code": "120160", "description": "Spring 2015 - Urbana-Champaign"}, {"code": "120158", "description": "Fall 2015 - Urbana-Champaign"}, {"code": "120158", "description": "Fall 2015 - Urbana-Champaign"}, {"code": "120155", "description": "Summer 2015 - Urbana-Champaign"}, {"code": "120155", "description": "Summer 2015 - Urbana-Champaign"}, {"code": "120151", "description": "Spring 2015 - Urbana-Champaign"}, {"code": "120151", "description": "Spring 2015 - Urbana-Champaign"}, {"code": "120150", "description": "Winter 2014-2015 Urbana-Chmpgn"}, {"code": "120150", "description": "Winter 2014-2015 Urbana-Chmpgn"}, {"code": "120148", "description": "Fall 2014 - Urbana-Champaign"}, {"code": "120148", "description": "Fall 2014 - Urbana-Champaign"}, {"code": "120145", "description": "Summer 2014 - Urbana-Champaign"}, {"code": "120145", "description": "Summer 2014 - Urbana-Champaign"}, {"code": "120141", "description": "Spring 2014 - Urbana-Champaign"}, {"code": "120141", "description": "Spring 2014 - Urbana-Champaign"}, {"code": "120138", "description": "Fall 2013 - Urbana-Champaign"}]
```

1.) Highlight "getTerms" request

Finding source code from url

URL: <https://host/StudentRegistrationSSB/ssb/classRegistration/getTerms>

A screenshot of an IDE (Eclipse) showing the source code for a Groovy controller. The project structure on the left shows the path to the controller file. The main editor displays the code for the `getTerms()` method. Two red arrows point from the URL components to the corresponding parts of the code: one from `classRegistration` to the controller name in the project tree, and another from `getTerms` to the method definition in the code.

```
ClassRegistrationController.groovy - [banner_student_registration_ssb_app] - banner_student_registration_ssb_app /localgit/apps/bann...
Project
  banner_student_registration_ssb_app (~/.localgit/apps/...
    .idea
    .settings
    org.codehaus.groovy.eclipse.preferences.prefs
    grails-app
      conf
      controllers
        net.hedtech.banner.student.registration
          ClassRegistrationController
          ClassSearchController
          ContactCardController
          CourseSearchController
          CourseSearchResultsController
          PlanController
          PrepareRegistrationController
          RegistrationController
          RegistrationHistoryController
          SearchResultsController
          TermController
        domain
          i18n
          services
          taglib
          utils
          views
  Persistence
  Web

ClassRegistrationController.groovy x
ClassRegistration
ClassRegistrationController
ClassRegistration Views

570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599

def registrationEvents retrieveRegistrationEvents(params?.term
render registrationEvents as JSON
}
}

def getTerms() {
  def terms
  if (session.getAttribute("selectedRole")?.facultyIndicator) {
    terms = selfServiceLookupService.fetchSelectableFacultyTermsf
      params.max.toInteger(), (params.offset.toInteger() *
  } else if (session.getAttribute("selectedRole")?.administratorInd
    terms = selfServiceLookupService.fetchAllTermsForRegistrar(pa
      (params.offset.toInteger() * params.max.toInteger())
      params.max.toInteger())
  } else {
    terms = selfServiceLookupService.fetchSelectableTerms(params.
      (params.offset.toInteger() * params.max.toInteger())
      params.max.toInteger())
  }

  render terms.collect {
    [code: it.code, description: it.description.encodeAsHTML()]
  } as JSON
}
```

Chrome Dev Tools CSS fun!!



- 1.) Goal – modify the appearance
- 2.) Click “Elements” tab and highlight “select2-results” class for example
- 3.) Elements on page automatically highlight and css shows up on right

Chrome Dev Tools CSS fun cont..



The screenshot shows a browser window with the URL `https://banner-dev.apps.uillinois.edu/StudentRegistrationSSB-BANDRI/ssb/term/termSelection?mode=registration`. The page content includes a "SELECT A TERM" section with a search box and a dropdown menu. The dropdown is open, showing a list of terms: "Spring 2016 - Urbana-Champaign", "Winter 2015-2016 Urbana-Champaign", "Fall 2015 - Urbana-Champaign", "Summer 2015 - Urbana-Champaign", "Spring 2015 - Urbana-Champaign", "Winter 2014-2015 Urbana-Champaign", "Fall 2014 - Urbana-Champaign", "Summer 2014 - Urbana-Champaign", and "Spring 2014 - Urbana-Champaign".

The Chrome DevTools interface is open at the bottom. The "Elements" panel shows the HTML structure of the dropdown menu. The "Styles" panel shows the computed styles for the selected element, including a red color for the text.

3.) Changes magically appear!!

1.) Click Add

2.) Add some css

Add css changes to XE app



- Create or modify
\$BANNER_HOME/xe_app/instance/config/css
/bannerSelfService-custom.css file
- `ul#select2-results {
 color: red;
}`
- Save the changes and rebuild and deploy
- We've done this to effectively hide fields, columns, etc..

Application Log File



- Comes with every application and can be customized in terms filename/directory/logging level
- First place to start with issues
- Inside file StudentRegistrationSsb_configuration.groovy

```
log4j = {  
    def String loggingFileDir = "/u01/app/oracle/apache_logs" ← custom logging directory  
    def String logAppName = "StudentRegistrationSSB-BANDR1" ← custom log file name  
    def String loggingFileName = "${loggingFileDir}/${logAppName}.log".toString()  
    appenders {  
        rollingFile name:'appLog', file:loggingFileName, maxFileSize:"${10*1024*1024}", maxBackupIndex:10, layout:pattern(  
conversionPattern: '%d{[EEE, dd-MMM-yyyy @ HH:mm:ss.SSS]} [%t] %-5p %c %x - %m%n' )  
    }  
}
```

Application Log Error



Sorry, the server received an internal error.
Error Message: No Institution code specified in the URL



specific error

place in code where
error occurred

```
urbxdev1b-oracle-> tail -100 StudentRegistrationSSB-BANDR1.log
[Wed, 10-Feb-2016 @ 09:43:11.874] [[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'] ERROR org.codehaus.groovy.g
GrailsExceptionHandler - MepCodeNotFoundException occurred when processing request: [GET] /StudentRegistrationSSB-BANDR1/ssb/registration
No Institution code specified in the URL. Stacktrace follows:
No Institution code specified in the URL
    at net.hedtech.banner.db.BannerDS.setMepSsb(BannerDS.groovy:642)
    at net.hedtech.banner.db.BannerDS.getConnection(BannerDS.groovy:81)
    at net.hedtech.banner.student.registration.RegistrationController.getGtvsdaxForSession(RegistrationController.groovy:66)
    at net.hedtech.banner.student.registration.RegistrationController.registration(RegistrationController.groovy:18)
    at weblogic.servlet.internal.StubSecurityHelper$ServletServiceAction.run(StubSecurityHelper.java:227)
```

XE Application debugging



- XE comes with log4j built into it
- Can modify logging level across board (careful with VERY large log files with trace or debug)
- Can enable logging for individual packages or classes
- Can customize log file name and directory
- Changes made to [appname]_configuration.groovy then rebuild and deploy
- JMX extensions for dynamic logging (haven't tried yet)
- Customized logging by modifying code/rebuild/redeploy

Global Log Change



- Global log change using root logger in [appname]_configuration.groovy
- Log levels (off, fatal, error, warn, info, debug, trace, all)

```
case 'production':
```

```
    root {
```

Output to defined appender or default "stdout"

log level

```
        error appLog'
```

```
        additivity = true
```

```
    }
```

Targeted log change



e.g. having authentication issues??

```
// ***** Security framework classes *****  
  
trace 'net.hedtech.banner.security'  
trace 'net.hedtech.banner.db'  
trace 'net.hedtech.banner.security.BannerAccessDecisionVoter'  
trace 'net.hedtech.banner.security.BannerAuthenticationProvider'  
off 'net.hedtech.banner.security.CasAuthenticationProvider'  
trace 'net.hedtech.banner.security.SelfServiceBannerAuthenticationProvider'  
trace 'grails.plugins.springsecurity'  
trace 'org.springframework.security'  
trace 'org.apache.http.headers'  
trace 'org.apache.http.wire'
```

Custom log change



Using `getTerms` again as example, add following sample code to `ClassRegistrationController.groovy` then run locally

```
def getTerms() {  
  
    def terms  
    if (session.getAttribute("selectedRole")?.facultyIndicator) {  
        terms = selfServiceLookupService.fetchSelectableFacultyTermsForRegistration(StudentSsbControllerUtility.getPdm(), params.searchTerm,  
            params.max.toInteger(), (params.offset.toInteger() * params.max.toInteger()) - (params.max.toInteger() - 1) - 1)  
    } else if (session.getAttribute("selectedRole")?.administratorIndicator) {  
        terms = selfServiceLookupService.fetchAllTermsForRegistrar(params.searchTerm,  
            (params.offset.toInteger() * params.max.toInteger()) - (params.max.toInteger() - 1) - 1,  
            params.max.toInteger())  
    } else {  
        terms = selfServiceLookupService.fetchSelectableTerms(params.searchTerm,  
            (params.offset.toInteger() * params.max.toInteger()) - (params.max.toInteger() - 1) - 1,  
            params.max.toInteger())  
    }  
  
    log.error "** log test for eLive 2016 **"  
    log.error "${terms}"  
    log.error "** end log test for eLive 2016 **"  
  
    render terms.collect {  
        [code: it.code, description: it.description.encodeAsHTML()]  
    } as JSON  
}
```

add debug code

Custom logging output



- View output in log or stdout
- Output of “**terms**” is nicely displayed automatically

```
Configuring Spring Security Core ...  
... finished configuring Spring Security Core  
| Server running. Browse to http://localhost:8080/StudentRegistrationSsb  
| Error 2016-01-27 14:13:05,437 [http-bio-8080-exec-4] ERROR registration.ClassRegistrationController - ** log test for eLive 2016 **  
| Error 2016-01-27 14:13:05,446 [http-bio-8080-exec-4] ERROR registration.ClassRegistrationController - [Term[id=670, code=120161,  
description=Spring 2016 - Urbana-Champaign, searchDescription =SPRING 2016 - URBANA-CHAMPAIGN, version=2 ], Term[id.....  
| Error 2016-01-27 14:13:05,447 [http-bio-8080-exec-4] ERROR registration.ClassRegistrationController - ** end log test for eLive 2016
```

Oracle AWR reporting



- AWR (Automated Workload Repository)
- **Note – does require diagnostics pack license**
- Database snapshots taken every hour by default (or manually)
- The awrrpt.sql SQL script generates an HTML or text report that displays statistics for a range of snapshot Ids.
- I like to isolate in a dev/test environment and run something
- Essential for use in load testing
- Great for “generically” tracing a connection pool

AWR in action/Reg XE Load testing



Disaster!!!



Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
latch: row cache objects	13,754,847	120.4K	9	79.9	Concurrency
library cache: mutex X	29,846,296	21.1K	1	14.0	Concurrency
DB CPU		9554		6.3	
cursor: pin S wait on X	620	52.3	84	.0	Concurrency
cursor: mutex X	158,431	50.1	0	.0	Concurrency

BEFORE

AFTER
JDBC Caching

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		273.8		80.2	
library cache: mutex X	120,845	40.4	0	11.8	Concurrency
control file sequential read	31,001	16.7	1	4.9	System I/O
log file sync	2,311	13.3	6	3.9	Commit
cursor: pin S wait on X	99	4.6	46	1.3	Concurrency
rdh file sequential read	801	3.2	4	1.0	User I/O

AWR features



	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	729	09-Nov-15 08:31:22	220	2.1
End Snap:	730	09-Nov-15 09:06:16	222	2.3
Elapsed:		34.89 (mins)		
DB Time:		34.68 (mins)		

← AWR header

SQL ordered by elapsed time →

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 113.3% of Total DB Time (s): 2,081
- Captured PL/SQL account for 22.5% of Total DB Time (s): 2,081

2 queries 67% of activity!!

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
891.96	1	891.96	42.86	31.69	70.74	836s3y8c478us	RPEDISB	select (((SUBSTR(SPRIDEN_LAST...
497.56	763	0.65	23.91	94.83	5.38	795ba0tcuytub	RPEDISB	select X.RPRADSB_FUND_CODE , X...
336.35	763	0.44	16.16	42.06	59.11	bad15xphtzh7s	RPEDISB	declare LV_PARM_ONE_UP number ...
277.86	255,161	0.00	13.35	8.48	93.86	3gtccw2qh6ry5	RPEPELL	SELECT RCRAPP4_C_INST_1_TOT_FA...
120.89	2,805	0.04	5.81	54.39	46.45	fxzvtbn0qbbmq	RPEDISB	SELECT SFRSTCR_CRN, SFRSTCR_TE...
118.18	2,805	0.04	5.68	10.25	91.37	dj4fzymf1gy8y	RPEDISB	SELECT SHRTCKN_CRN, SHRTCKN_TE...

SQL Statistics

- SQL ordered by Elapsed Time
- SQL ordered by CPU Time
- SQL ordered by User I/O Wait Time
- SQL ordered by Gets
- SQL ordered by Reads
- SQL ordered by Physical Reads (UnOptimized)
- SQL ordered by Executions
- SQL ordered by Parse Calls
- SQL ordered by Sharable Memory
- SQL ordered by Version Count
- Complete List of SQL Text

← SQL report options

Summary



- **It's a luxury to be able to use Banner 8 while you get ramped up on XE (now's the time!!)**
- **Ease your way into it/get comfortable with the technologies and architecture – there is a learning curve**
- **Make some mods if only for the practice of seeing it all put together**

ellucian
LIVE | 2016



Questions??



Thank you!

Brian Schoudel – brians@uillinois.edu

John Tanner – jrtanner@uillinois.edu

Please complete the online session evaluation form.

Session ID 36090

We need your help

Feedback Survey

Access session surveys by using the survey widget on the mobile app or by logging into your session schedule builder at

<http://tinyurl.com/elive2016surveys>

Brian Schoudel – brians@uillinois.edu

John Tanner – jrtanner@uillinois.edu

Session ID 36090