

ellucian™

Banner Flexible Registration Administration Guide

Release 8.6
June 2013



Banner®, Colleague®, Luminis® and Datatel® are trademarks of Ellucian or its affiliates and are registered in the U.S. and other countries. Ellucian™, PowerCampus™, Advance™, Degree Works™, fsaATLAS™, Course Signals™, SmartCall™, Recruiter™, and ILP™ are trademarks of Ellucian Company L.P. or its affiliates. Other names may be trademarks of their respective owners.

©2009-2013 Ellucian Company L.P. and its affiliates. The unauthorized possession, use, reproduction, distribution, display or disclosure of this material or the information contained herein is prohibited.

Contains confidential and proprietary information of Ellucian and its subsidiaries. Use of these materials is limited to Ellucian licensees, and is subject to the terms and conditions of one or more written license agreements between Ellucian and the licensee in question.

In preparing and providing this publication, Ellucian is not rendering legal, accounting, or other similar professional services. Ellucian makes no claims that an institution's use of this publication or the software for which it is provided will guarantee compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Prepared by: Ellucian
4375 Fair Lakes Court
Fairfax, Virginia 22033
United States of America

Revision History

Publication Date	Summary
June 2013	New version that supports Banner Flexible Registration 8.6 software.

Banner Flexible Registration 8.6 Administration Guide

Contents

Chapter 1	Introduction	1-1
	Product overview	1-1
	Banner Flexible Registration forms overview	1-2
	Attachment settings	1-3
	Flexible Registration File Attachment Validation Form (SFTVATCH)	1-3
	Flexible Registration File Attachment Association Form (SFFAATCH)	1-3
	Flexible Registration File Attachment Review Form (SFFRATCH)	1-4
	Configuration settings	1-4
	Flexible Registration Configuration Rules Form (SFRACNFG)	1-4
	Catalog settings	1-4
	Flexible Registration Catalog Validation Form (SFTVCTLG)	1-4
	Flexible Registration Catalog Rules Form (SFRACTLG)	1-4
	Discount sets and coupons	1-4
	Flexible Registration Discount Set Validation Form (SFTVDSET)	1-4
	Flexible Registration Discount Conditions Form (SFDACOND)	1-5
	Flexible Registration Discount Rules Form (SFDRRULE)	1-5
	Flexible Registration Coupon Rules Form (SFDRCUPN)	1-5
	Profile settings	1-5
	Flexible Registration Date Format Validation Form (SFTVDATE)	1-5
	Flexible Registration Profile Validation Form (SFTVPROF)	1-5
	Flexible Registration Profile Rules Form (SFRAPROF)	1-5
	Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV)	1-6
	Flexible Registration Person Extended Profile Form (SFGIEPRF)	1-6
	Email notification	1-6
	Flexible Registration Email Notification Form (SFRAEMAL)	1-6
	Additional materials	1-6
	Flexible Registration Additional Materials Validation Form (SFTVAMAT)	1-6
	Flexible Registration Additional Materials Rules Form (SFARAMAT)	1-6
	Flexible Registration Additional Materials Course Association Form (SFARCRSE)	1-7

Flexible Registration Additional Materials Section	
Association Form (SFARSECT)	1-7
Flexible Registration Additional Materials Inquiry Form (SFAIAMAT)	1-7
Purchase order settings	1-7
Flexible Registration Purchase Order to Third Party	
Contract Form (SFBACONT)	1-7
Cart information	1-7
Flexible Registration Cart Inquiry Form (SFAICART)	1-7
Advanced search settings	1-8
Flexible Registration Advanced Search Validation Form (SFTVSRCH)	1-8
Flexible Registration Advanced Search Rules Form (SFRASRCH)	1-8
Relationship settings	1-8
Flexible Registration User Relationship Validation Form (SFTVRLTN)	1-8
Flexible Registration User Relationship Form (SFPARLTN)	1-8
Group Registration	1-8
Flexible Registration Adjustments Inquiry Form (SFFIADAR)	1-8
Banner Flexible Registration web application overview	1-9
Home page	1-9
Search functionality	1-9
Basic search	1-9
Advanced search	1-10
Search results	1-10
Browse by Subject	1-12
Class details	1-12
Schedule Builder	1-15
Weekly View	1-16
List View	1-16
Registration functionality	1-16
Select students	1-16
Select additional options	1-16
Registration results	1-16
Upload Attachments	1-16
Payment options	1-17
Final invoice	1-17
Manage Registration	1-17
Saved Plans	1-17
My Profile	1-17

Chapter 2 Installation 2-1

Hardware and software requirements 2-1



Hardware requirements	2-1
Application server	2-1
Database	2-1
New installation	2-1
Upgrading from a prior version	2-2
Software requirements	2-2
Ellucian solution dependencies	2-2
Oracle dependencies	2-3
Java dependencies	2-3
Download deployment files	2-3
Stage deployment files	2-3
Installation preparation	2-4
Install Banner Flexible Registration on Oracle 11g	2-6
Upgrade to Banner Flexible Registration on Oracle 11g	2-10

Chapter 3 System Configuration 3-1

Application Home	3-1
Why Application Home is needed?	3-1
Application Home folder structure	3-1
Setting up Application Home	3-2
Banner Flexible Registration timeout configuration	3-2
Configure user session timeout	3-2
Token timeout value	3-3
Cart expiration process	3-3
Cart expiration automated process configuration	3-4
Default values	3-4
Cart expiration	3-5
Repeat interval	3-5
Logging	3-6
File Attachments	3-6
Considerations	3-6
Fee Adjustment Process	3-8

Self-Service Banner course details link configuration	3-9
Email notification	3-9
Supported events	3-9
Create emails	3-10
Configure notification by Flexible Registration catalog	3-10
Email execution interval and the JMS queue	3-10
Configure OAS 11g for email notifications	3-11
Debugging	3-22
Banner Flexible Registration Configuration Validation Tool	3-22
Version reporting	3-23
Usage	3-23
Manage plans	3-24
Externalizing resources	3-24
Supported resources	3-25
Resource Folder structure	3-25
Map folders to deployed application	3-26
Set up the resource folder	3-27
Populate the resource folder.	3-28
Empty resource folder	3-28
Fully populated resource folder	3-28
Download resource bundle	3-28
References.	3-29
Chapter 4 Product Administration	4-1
Install in an SSL-enabled environment	4-1
Set up Banner Flexible Registration	4-1
File attachments	4-29
Create attachments	4-29
Uploading of attachments	4-31
Review attachments	4-31
Manage attachments	4-32
New user creation	4-32



General student record	4-34
Overriding Quick Entry Rules	4-34
Custom Override Function API	4-34
Credential Notification	4-35
Acceptable usage indicator	4-36
Relationships.	4-37
Setup	4-38
Creating and Viewing.	4-39
Registration.	4-40
Fee assessment.	4-41
Fee Adjustment Process.	4-42
Discounts and coupons	4-43
Create discount sets	4-44
Create coupon codes.	4-45
Invoicing	4-46
Section charges invoice	4-46
Total fee assessment invoice	4-47
Additional materials.	4-48
Override additional material settings for a course.	4-49
Overriding additional material settings for a section	4-49
Additional materials order fulfillment	4-50
Idle cart expiration	4-50
Payment options.	4-50
External payment	4-52
Purchase order	4-52
Bill me later	4-53
Set up credit card types	4-53
Credit card using the Payment Processor Connection	4-53
Payment Processor Connection URL.	4-54
Home page configuration	4-56
Specify a catalog specific URL	4-56
Create a catalog specific HTML file.	4-57
Modify the default home page content	4-57
Externalize HTML content	4-58
HTML fragment support	4-58

Supported HTML tags	4-58
Supported HTML attributes	4-60
Image support	4-61
JavaScript support	4-62
CSS support	4-62
Support for Flexible Registration Style	4-62
Homepage URL displayed on sign out	4-62
Permalinks	4-63
Create a permalink using the basic search	4-65
Create a permalink using the advanced search	4-65
Restrictions	4-66
Create a permalink on the Landing Page	4-68
Searching	4-68
Basic search	4-68
CRN search in basic search	4-69
Basic search information	4-69
Sort order for search results	4-70
Advanced search	4-70
Operators	4-71
and	4-71
or	4-71
Wildcard	4-71
Term searching	4-72
Maximum search results	4-72
Customizing advanced search fields	4-73
Search processing	4-74
Browse by Subject search	4-75
Authentication	4-76
Single Sign-On (SSO)	4-76
Central Authentication Service (CAS)	4-76
Define CAS managed service	4-79
LDAP configuration on Oracle 11g Application Server	4-80



	Banner Self-Service integration	4-81
	Single Sign-on.	4-81
	Self-Service menu links	4-81
	Manual Web Tailor updates	4-81
	Prohibit student registration through Student Self-Service (Optional)	4-83
	Prohibit students from adding/dropping classes via Registration Tools channel (Optional)	4-85
	Registration locks	4-85
	CAS Enabled Banner Self Service	4-86
	 Luminis integration	4-86
	Luminis authentication	4-86
	Using Luminis IV	4-86
	Luminis single sign-on	4-87
	Luminis IV	4-87
	Luminis V	4-87
Chapter 5	Processes	5-1
	 List of Processes	5-1
	Flexible Registration Fee Adjustment Process	5-3
Chapter 6	Third Party Software Licenses	6-1
	 Sun Microsystems, Inc.	6-1
	Apache License version 2.0	6-2
	 Berkeley Software Distribution (BSD)	6-4
	 Adobe Cairngorm license	6-5
	 Common Public license	6-5
	 Creative Commons Attribution license	6-5
	 Eclipse Public license	6-5
	 Lesser General Public license	6-6
	 MIT license	6-6

Public Domain	6-6
JA-SIG license	6-7

Appendix A Banner Flexible Registration Configuration Validation..... A-1

What configuration settings are checked?	A-1
Execution of the Configuration Validation Process.	A-2
Command line execution.	A-2
Make enhancements to the Configuration Validation Process	A-2
Validation Process output	A-3
Validation check errors.	A-3
Global Configuration Error Messages (sf_glbl_chk.sql)	A-3
Catalog Configuration Error Messages (sf_frc_chk.sql)	A-4
GTVSDAX rules (sf_sdax_chk.sql)	A-6
Term Rules (sf_term_chk.sql)	A-7
Profile Rules (sf_profile_chk.sql)	A-8
Local Check (sf_local_chk.sql)	A-8
System Privileges Check (sf_system_privs_chk.sql)	A-9
LDAP Check (sf_ldap_chk.sql)	A-11
PCI Check (sf_pci_chk.sql)	A-12
Test script patterns	A-13
Wrap all tests with exception handling	A-13
Leverage Banner Validation API calls	A-13

Appendix B Banner Flexible Registration Web Application Services Configuration B-1

Deploy application	B-1
Access web services	B-3
Read WSDL files	B-3
Locate and request URI	B-6



Modify generated addresses	B-7
Validate the configuration	B-9
Validate web service location URI.	B-9
Troubleshooting	B-10

Appendix C Banner Flexible Registration EhCache Configuration.....C-1

EhCache configuration	C-1
ehcache.xml	C-1
Section 1	C-1
Section 2	C-1
Section 3	C-2

Appendix D Banner Flexible Registration Captcha Configuration.....D-1

Captcha configuration	D-1
Troubleshooting	D-1
Captcha display issues on Linux	D-1

Appendix E Banner Flexible Registration Log FilesE-1

Default logging	E-1
Modifying the location of the log files	E-1
Identifying issues using the log files	E-2
Logging Levels and Personally Identifiable Information	E-4
Logging levels	E-4
Personally Identifiable Information	E-5
Reporting issues to Ellucian	E-5



1 Introduction

Welcome to the Banner Flexible Registration application. This guide contains the following information:

- [“Installation” on page 2-1](#)
- [“System Configuration” on page 3-1](#)
- [“Product Administration” on page 4-1](#)
- [“Third Party Software Licenses” on page 6-1](#)
- [“Banner Flexible Registration Configuration Validation” on page A-1](#)
- [“Banner Flexible Registration Web Application Services Configuration” on page B-1](#)

Product overview

Banner Flexible Registration is comprised of both Banner forms that allow you to customize the application and a Rich Internet Application that interfaces with Self-Service Banner and Luminis and allows your students to complete the registration process quickly and easily.

Banner Flexible Registration is a simple, web-based system that allows all of your institution’s learners to search, register, and pay for courses around the clock. This anytime and anywhere self-service environment provides a one-stop solution that can help your institution achieve the service levels it needs, efficiently and effectively.

Learners can take advantage of the rich internet capabilities built into Banner Flexible Registration to quickly and easily search for courses across course number, subject, title, and description as well as access detailed course and section information. During their search, they can easily add potential classes or sessions to a calendar view to plan their weekly schedule and identify any conflicts early on. Shopping cart functionality allows them to easily add courses, review applicable fees, apply any available discount coupon codes, and use the payment method of their choice to settle their account—all in a few simple steps.

Banner Flexible Registration also simplifies some of the complexities surrounding registration. If a course is already full, learners can be automatically waitlisted and receive notification if a seat becomes available. And because they can view their eligibility status, including registration holds, time ticketing, and other conditions that they might need to remedy before registering, students can use Banner Flexible Registration to easily resolve issues that could have involved multiple trips to multiple offices on your campus.

Your institution can build registration catalogs populated with courses that target particular programs and needs, to improve your ability to market your programs effectively. You can also use preconfigured, static permalinks to direct people to certain sessions or groups of classes, improving outreach for your continuing education or community service programs.

Banner Flexible Registration is integrated with your existing Banner® Open Digital Campus and leverages many of the processes your administrators are familiar with today. And with single sign on, end users won't even need to log into the system separately. It is just another part of the Banner solution.

Banner Flexible Registration forms overview

Banner Flexible Registration contains forms that allow you to manage global Flexible Registration settings, create and maintain catalogs, maintain profile rules, establish discounts and coupons, and define additional materials that are available for classes.

The following forms are available in Banner Flexible Registration:

- Flexible Registration Additional Materials Inquiry Form (SFAIAMAT)
- Flexible Registration Cart Inquiry Form (SFAICART)
- Flexible Registration Additional Materials Rules Form (SFARAMAT)
- Flexible Registration Additional Materials Course Association Form (SFARCRSE)
- Flexible Registration Additional Materials Section Association Form (SFARSECT)
- Flexible Registration Purchase Order to Third Party Contracts Form (SFBACONT)
- Flexible Registration Discount Conditions Form (SFDACOND)
- Flexible Registration Coupon Rules Form (SFDRCUPN)
- Flexible Registration Discount Rules Form (SFDRRULE)
- Flexible Registration File Attachment Association Form (SFFAATCH)
- Flexible Registration Adjustment Inquiry Form (SFFIADAR)
- Flexible Registration File Attachment Review Form (SFFRATCH)
- Flexible Registration Person Extended Profile Form (SFGIEPRF)
- Flexible Registration User Relationship Form (SFPARLTN)
- Flexible Registration Global Configuration Rules Form (SFRACNFG)
- Flexible Registration Catalog Rules Form (SFRACTLG)

- Flexible Registration Email Notification Form (SFRAEMAL)
- Flexible Registration Profile Rules Form (SFRAPROF)
- Flexible Registration Advanced Search Rules Form (SFRASRCH)
- Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV)
- Flexible Registration Additional Materials Validation Form (SFTVAMAT)
- Flexible Registration File Attachment Validation Form (SFTVATCH)
- Flexible Registration Catalog Validation Form (SFTVCTLG)
- Flexible Registration Date Format Validation Form (SFTVDATE)
- Flexible Registration Discount Set Validation Form (SFTVDSET)
- Flexible Registration Profile Validation Form (SFTVPROF)
- Flexible Registration User Relationship Validation Form (SFTVRLTN)
- Flexible Registration Advanced Search Validation Form (SFTVSRCH)



Note

The following information is an overview of the Banner Flexible Registration forms. Please refer to the Banner Flexible Registration Online Help for form, block, and field detail. ■

Attachment settings

Use the following forms to create and review attachments that learners can upload during the registration process.

Flexible Registration File Attachment Validation Form (SFTVATCH)

The Flexible Registration File Attachment Validation Form (SFTVATCH) allows you to define attachments that a student can upload during the registration process.

Flexible Registration File Attachment Association Form (SFFAATCH)

Flexible Registration File Attachment Association Form (SFFAATCH) allows you to associate attachments to Courses and Sections and to create a template location and help text for the attachment.

Flexible Registration File Attachment Review Form (SFFRATCH)

Flexible Registration File Attachment Review Form (SFFRATCH) allows you to review attachments that have been uploaded by Learners.

Configuration settings

Use the following form to manage global settings in the Banner Flexible Registration application:

Flexible Registration Configuration Rules Form (SFRACNFG)

The Flexible Registration Global Configuration Rules Form (SFRACNFG) allows you to specify system-wide rules for the Flexible Registration application. Rules setup on this form will apply across all Banner Flexible Registration 'catalogs'.

Catalog settings

Use the following forms to create and manage catalogs in the Banner Flexible Registration application:

Flexible Registration Catalog Validation Form (SFTVCTLG)

The Flexible Registration Catalog Validation Form (SFTVCTLG) allows you to define the catalog codes, description, start and end dates, discount set, and the password for the catalog if it is private.

Flexible Registration Catalog Rules Form (SFRACTLG)

The Flexible Registration Catalog Rules Form (SFRACTLG) allows you to enter and maintain catalog rules and associate sections or terms to the catalog.

Discount sets and coupons

Use the following forms to create and manage discount sets and coupons in the Banner Flexible Registration application:

Flexible Registration Discount Set Validation Form (SFTVDSET)

The Flexible Registration Discount Set Validation Form (SFTVDSET) allows you to create a discount code set.

Flexible Registration Discount Conditions Form (SFDACOND)

The Flexible Registration Discount Conditions Form (SFDACOND) allows you to create conditions that will be added to the discount set and used in the discount calculation process.

Flexible Registration Discount Rules Form (SFDRRULE)

The Flexible Registration Discount Rules Form (SFDRRULE) allows you to assign discount rules to a discount set using conditions that are created on the Flexible Registration Discount Conditions Form (SFDACOND).

Flexible Registration Coupon Rules Form (SFDRCUPN)

The Flexible Registration Coupon Rules Form (SFDRCUPN) allows you to create rules for coupon discounts.

Profile settings

Use the following forms to specify extended profile information that can be used in Banner Flexible Registration:

Flexible Registration Date Format Validation Form (SFTVDATE)

The Flexible Registration Date Format Validation Form (SFTVDATE) allows you to manage a list of birth date formats for use with birth date validation in the Personal Information screen. The birth date validation occurs when a student enters or updates his/her personal information.

Flexible Registration Profile Validation Form (SFTVPROF)

The Flexible Registration Profile Validation Form (SFTVPROF) allows you to add biographic and demographic data to be displayed in the Flexible Registration application.

Flexible Registration Profile Rules Form (SFRAPROF)

The Flexible Registration Profile Rules Form (SFRAPROF) allows you to identify how biographic and demographic data is displayed in the Flexible Registration application.

Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV)

The Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV) allows you to define acceptable values for extended profile information.

Flexible Registration Person Extended Profile Form (SFGIEPRF)

The Flexible Registration Person Extended Profile Form (SFGIEPRF) allows you to view data on students who have added personal information through Flexible Registration.

Email notification

Use the following form to create and manage email notification information that is associated with registration in the Banner Flexible Registration application:

Flexible Registration Email Notification Form (SFRAEMAL)

The Flexible Registration Email Notification Form (SFRAEMAL) is used to set up and maintain email notification information such as notification code, mime type, and email message.

Additional materials

Use the following forms to create and manage optional additional materials that are associated with courses in the Banner Flexible Registration application:

Flexible Registration Additional Materials Validation Form (SFTVAMAT)

The Flexible Registration Additional Materials Validation Form (SFTVAMAT) allows you to manage a list of additional materials for courses in a catalog.

Flexible Registration Additional Materials Rules Form (SFARAMAT)

The Flexible Registration Additional Materials Rules Form (SFARAMAT) allows you to specify the default price, detail code, and additional information for optional additional materials. This form also allows you to specify which courses or sections to associate the additional material.

Flexible Registration Additional Materials Course Association Form (SFARCRSE)

The Flexible Registration Additional Materials Course Association Form (SFARCRSE) allows you to override the default details on additional materials for a course.

Flexible Registration Additional Materials Section Association Form (SFARSECT)

The Flexible Registration Additional Materials Section Association Form (SFARSECT) allows you to view and associate additional materials to a section. It displays all of the additional materials associated with a section in an easy to read display. This form also allows you to override the default details on additional materials at the section level.

Flexible Registration Additional Materials Inquiry Form (SFAIAMAT)

The Flexible Registration Additional Materials Inquiry Form (SFAIAMAT) allows you to query and display information on the additional materials for a course.

Purchase order settings

Purchase orders use the Banner 3rd Party Contracts to associate credits for a student, based on rules setup on that contract. Use the following form to manage how purchase orders are used in the Banner Flexible Registration application:

Flexible Registration Purchase Order to Third Party Contract Form (SFBACONT)

The Flexible Registration Purchase Order to Third Party Contract Form (SFBACONT) allows you to create a purchase order ID and associate it to third party contract(s).

Cart information

Use the following form to view information on Banner Flexible Registration carts:

Flexible Registration Cart Inquiry Form (SFAICART)

The Flexible Registration Cart Inquiry Form (SFAICART) allows you to review Banner Flexible Registration carts.

Advanced search settings

Use the following forms to control the advanced search capabilities:

Flexible Registration Advanced Search Validation Form (SFTVSRCH)

The Flexible Registration Advanced Search Validation Form (SFTVSRCH) allows you to create the advanced search code and a corresponding description. The rules for these codes will be created in the SFRASRCH form.

Flexible Registration Advanced Search Rules Form (SFRASRCH)

The Flexible Registration Advanced Search Rules Form (SFRASRCH) allows you to customize the advanced search. You can select the Advanced Search Template Code from the **Advanced Search Code** field and specify the order that the search attributes are displayed on the advanced search window. The SFRASRCH form allows you to choose what Advanced Search attributes will default to on the Advanced Search modal window. It also allows you to choose what other attributes will be available to the student to select from the attribute list.

Relationship settings

Use the following forms to view and create relationships.

Flexible Registration User Relationship Validation Form (SFTVRLTN)

The Flexible Registration User Relationship Validation Form (SFTVRLTN) allows you to define the relationship codes and set rules for each relationship code.

Flexible Registration User Relationship Form (SFPARLTN)

The Flexible Registration User Relationship Form (SFPARLTN) is where the relationship is created between the users using the relationship codes. This form allows you to create a relationship, edit a relationship code, and disable a relationship.

Group Registration

Flexible Registration Adjustments Inquiry Form (SFFIADAR)

The Flexible Registration Adjustment Inquiry Form (SFFIADAR) allows you to review the fee adjustment information and Job Submission information. This form shows detailed

information about the adjustment. Various details about the adjustment can be found such as the AR Tran Number, the Shopping Cart, Cross Reference and IDs.

Banner Flexible Registration web application overview

The Banner Flexible Registration web application contains the following features:

Home page

The Banner Flexible Registration Home page allows a student to search for classes, view important information, sign in to the application, and view their registration eligibility status for the catalog they are accessing.

The Home page contains a fixed area for searching and an area that can display customizable HTML content. The content that is displayed on the home page may be customized on a catalog by catalog basis.

In Banner Flexible Registration, a valid user account is required to register for classes. When a student signs in to the application, their registration time ticket, any holds on their account that may prevent them from registering for classes, and their current status as a student are displayed. The display of the Time Ticket and the Status sections can be controlled by a catalog setting on the SFRACTLG form. An institution can choose to not display one or both of these sections.

Search functionality

Banner Flexible Registration contains both a basic search and an advanced search.

Basic search

Searching for a class in Banner Flexible Registration returns all matches found in the following Banner fields:

- Subject Code
- CRN
- Title
- Course number
- Description

Advanced search

The advanced search in Banner Flexible Registration allows a student to search on the following information:

- Campus
- Class start date
- Class times
- CRN
- Course attribute
- Course level
- Course number
- Credit hours
- Description
- Instructional method
- Meeting days
- Only show sections with seats available?
- Instructor (Last Name)
- Schedule type
- Subject
- Term
- Title

Search results

Search results in Banner Flexible Registration can be viewed in either a compact view or a detailed view.

Compact view

The compact search results view displays the following information for all classes that satisfy the executed search:

- Subject Code
- Course number
- Section number
- Title

- Meeting days
- Meeting Start Date
- Meeting End Date
- Class times
- Campus
- Seats available
- Waitlist
- Building Description
- Building Room
- Base Fees

Detailed view

The detailed search results view displays the following information for all classes that satisfy the executed search:

- Subject Code
- Course Number
- Section Number
- Title

Note

For the section long title to be displayed in the search results, the Display Long Section Title option in Schedule Search Controls must be checked under the Web Processing Controls page on the Term Control Form (SOATERM). The section long title is entered on the Section Syllabus form (SSASYLB). If this option is unchecked, the section title on Schedule form (SSASECT) will be displayed. ■

Note

The course long title (from the Course Syllabus form (SCASYLB)) will not be displayed even if the Display Long Course Title option in the Catalog Search Controls section of the Term Control Form (SOATERM) is checked. ■

- CRN
- Term
- Meeting Days
- Meeting Start Date
- Meeting End Date

- Campus
- Instructor
- Seats Available
- Waitlist
- Base Fees
- Building Description
- Building Room

Browse by Subject

Selecting Browse by subject displays all the subjects in the Browse Subjects listing and the search results for the selected subject are displayed under Browse Results.

Class details

The Class Details window contains class overview information, class meeting times, class restrictions, information on additional materials, and required materials for the class.

Overview

The Overview portion of Class Details contains high level information about the selected class including term, instructor, and credit hours, as well as the course description.

If the course has a section comment, the course description is not displayed. All the text from the course has to be duplicated at the section level if it is different from the course text. Additionally, copying the text has to be done line-by-line since copy and paste of the whole description is not available.

If both SCBDESC_TEXT_NARRATIVE and SSRTEXT_TEXT are displayed, the course description does not have to be manually duplicated. A new catalog rule can be added to specify whether to add the course SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT) to the end of the **Class Description field**.

If the course description has to be displayed always, the settings can be made on the Flexible Registration Catalog Rules Form (SFRACTLG). If the course description is not to be displayed, select **No (N)** for the **Always Display Course Description on Course Details window?** field on the SFRACTLG form. Select **Yes (Y)** to always display the course description.

If the new catalog setting is set to *N*, the hierarchy for populating the course description is as follows:

1. When the **Display Long Section Description** and the **Display Long Course Description** indicators on SOATERM are checked (*Y*), display one of the following:
 - SSBDESC_TEXT_NARRATIVE, if NULL then display
 - SSRTEXT_TEXT, if NULL then display
 - SCBDESC_TEXT_NARRATIVE, if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING

2. When the **Display Long Section Description** indicator on SOATERM is checked (*Y*), and the **Display Long Course Description** indicator is unchecked (*N*), display one of the following:
 - SSBDESC_TEXT_NARRATIVE, if NULL then display
 - SSRTEXT_TEXT, if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING

3. When the **Display Long Section Description** indicator on SOATERM is unchecked (*N*), and the **Display Long Course Description** indicator is checked (*Y*), display one of the following:
 - SSRTEXT_TEXT, if NULL then display
 - SCBDESC_TEXT_NARRATIVE, if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING

4. When the **Display Long Section Description** indicator and the **Display Long Course Description** indicators on SOATERM are unchecked (*N*), display one of the following:
 - SSRTEXT_TEXT, if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING

If the new catalog setting is set to *Y*, the hierarchy for populating the course description is as follows:

1. When the **Display Long Section Description** and the **Display Long Course Description** indicators on SOATERM are checked (*Y*), display the following:
 - SSBDESC_TEXT_NARRATIVE and SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if SSBDESC_TEXT_NARRATIVE NULL then display
 - SSRTEXT_TEXT and SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if SSRTEXT_TEXT NULL then display
 - SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING
2. When the **Display Long Section Description** indicator on SOATERM is checked (*Y*), and the **Display Long Course Description** indicator is unchecked (*N*), display the following:
 - SSBDESC_TEXT_NARRATIVE and SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if SSBDESC_TEXT_NARRATIVE NULL then display
 - SSRTEXT_TEXT and SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if SSRTEXT_TEXT NULL then display
 - SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING
3. When the **Display Long Section Description** indicator on SOATERM is unchecked (*N*), and the **Display Long Course Description** indicator is checked (*Y*), display the following:
 - SSRTEXT_TEXT and SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if SSRTEXT_TEXT NULL then display
 - SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING

4. When the **Display Long Section Description** and the **Display Long Course Description** indicators on SOATERM are unchecked (*N*), display the following:
- SSRTEXT_TEXT and SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if SSRTEXT_TEXT NULL then display
 - SCBDESC_TEXT_NARRATIVE (if SCBDESC_TEXT_NARRATIVE is NULL then display SCRTEXT_TEXT), if NULL then display
 - SCRTEXT_TEXT, if NULL then display
 - NOTHING

 **Note**

When the new catalog setting is set to Y, it ignores the **Display Long Course Description** indicator on SOATERM. ■

Meeting times

The Meeting times portion of Class Details contains information on the dates, times, and location that a class meets, and the building description and building room.

Restrictions

The Restrictions portion of Class Details provides a link to Self-Service Banner to view any restrictions that may exist for the selected class to allow the student to determine if they are eligible to register for the class.

Additional materials

The Additional Materials portion of Class Details lists any additional materials that are available for the selected class and the cost of those materials to allow the student to make an educated decision when registering for the class.

Required materials

The Required Materials portion of Class Details lists the notes and materials needed for the Class. When students are going to decide which class to register for, this information will help them sign up for the class that provides the type of learning and experience they are looking for.

Schedule Builder

Banner Flexible Registration contains a Schedule Builder that allows a student to build their schedule using either a Weekly View or a List View.

Weekly View

The Schedule Builder Weekly View page displays a calendar that allows a student to add classes onto their schedule and view those classes on the calendar. This calendar view allows a student to visually plan their schedule and ensure that they have enough time between classes to make it to their next scheduled class.

List View

The Schedule Builder List View page displays a list view that allows a student to add classes onto their schedule and view those classes in a list view. This list view allows a student to view detailed information about their classes.

Registration functionality

The registration process in Banner Flexible Registration includes the following pages:

Select students

For group registrations, a page is presented to assign different students to a class or the ability to create new users.

Select additional options

On the Select additional options page, a student can select class options such as credit hours, grade mode, and course level. The student may also elect to purchase additional materials that are available for the selected classes.

Registration results

The Registration results page contains a list of the classes and associated charges for which a student is attempting to register.

In case of group registration, the Registration results page displays a list of the classes and associated charges for every student being registered by the group cart owner.

Upload Attachments

When attachments are associated to courses or sections the student is registering for, an Upload Attachments page will be displayed. This page enables a student to upload requested attachments at this point in time.

The Upload Attachments page enables the group cart owner to upload the requested attachments for the students being registered.

Payment options

On the Payment options page, a student may select from the following payment options:

- Credit card
- Purchase order
- Bill me later

Final invoice

A final invoice is generated when a student has successfully registered for classes and completed the billing process. The final invoice contains information on their payment method, class schedule, and the total amount invoiced.

Manage Registration

The Manage Registration page allows a student to review their current registrations. It allows a student to add a class from the waitlist, change their class status to audit, and drop a class. It also allows a student to specify class settings for any class in which they are allowed to customize.

Saved Plans

A plan is a set of classes that a student has saved from the Schedule Builder. A plan can be saved to allow it to be accessed at a later date. Multiple plans may be created to allow a student to create multiple potential class registration scenarios.

When registration is active, any saved plan may be opened and a student can attempt to register for those classes.

My Profile

The My Profile tab contains links that allow a student to update their personal information, manage their invoices, change their password, and reset their security questions.



2 Installation



This chapter details the steps required to perform a new install of Banner Flexible Registration or an upgrade to the current release.

Hardware and software requirements

Banner Flexible Registration requires the following hardware and software:

Hardware requirements

Banner Flexible Registration requires the following hardware:

Application server

Banner Flexible Registration has been tested on Oracle Application Server:

- OAS OC4J VM Memory Allocation: 1-2 GB
- Hard drive: minimum disk allocation of 2 GB (Oracle 11g)
- Processor
 - Intel: Dual 2.4 ghz.
 - Sparc: 1 ghz UltraSparc 6 core

Quad core recommended for large schools.

Database

Tablespace requirements will vary based upon the size of each institution. Most institutions should find 50 MB of tablespace to be sufficient for normal production use.

New installation

If you are not running a version of Banner Flexible Registration prior to 8.5.5, the 8.5.4 new install can be run.

Please refer to the 8.5.4 new installation guide for further instructions. Once 8.5.4 has been installed, the upgrade to 8.6 can be successfully run.

Upgrading from a prior version

If you are running a version of Banner Flexible Registration prior to 8.5, this upgrade cannot be used. You must upgrade from your current version to 8.5 prior to running the 8.6 upgrade.

Flexible Registration 8.6 includes all the modifications included in the following releases:

- 8.5.1
- 8.5.2
- 8.5.3
- 8.5.4
- 8.5.5

Because this upgrade is an aggregation release, the 8.6 upgrade can be run from any of the above versions.

Software requirements

Banner Flexible Registration requires the following software:

Ellucian solution dependencies

The following table indicates the software version of each Ellucian solution that you must install before installing Banner Flexible Registration.

 **Note**

Refer to *Banner UDC Release Interdependency Matrix* for the most up to date interdependencies. You can access and download it from the Download Center. ■

Ellucian Solution	Banner Flexible Registration 8.6
Banner Student	8.5.6
Banner Student Self-Service	8.5.4
Banner General	8.6
Banner Web Tailor	8.6

Oracle dependencies

The following table indicates the minimum version of Oracle components that must be installed before installing Banner Flexible Registration.

Oracle Solution	Banner Flexible Registration 8.6
Oracle Application Server (OAS)	Oracle Fusion Middleware 11g
Oracle Database	Oracle Database 11g; version 11.1.0.6.0 is the minimum required.

Java dependencies

The following table indicates the version of Java components that must be installed before installing Banner Flexible Registration.

Java Solution	Banner Flexible Registration 8.6
Java JDK	1.6.x

Download deployment files

Access the Customer Support Center (<https://connect.ellucian.com/>) and download the following deployment files:

File	Description	Version
sf80600u.trz	Flexible Registration System	8.6
sf80600d.trz	Flexible Registration Documentation	8.6

Stage deployment files

Copy the deployment files to a staging location and use the standard `unltded` procedure to extract the contents of these files. In the installation steps, below this location will be referred to as `%Staging%`.

Installation preparation

With the 11g Weblogic Oracle Application Server, the contents of the EAR file are not expanded. As such, the required configuration changes must be made before the EAR is deployed.

The instructions below may be applied for 11g environments.

Unpacking the EAR allows you to expand the contents of the EAR file and make the necessary modifications and then repack or package the EAR for deployment. It is assumed that the jar tool, included with the JAVA JDK, will be used to unpack and repack the EAR file. Other tools capable of expanding and archiving ZIP files can also be used.

Step 1 Unpack the EAR file

To unpack the EAR, perform the following steps:

1. Create a temporary folder in the %Staging% directory.
2. Enter *bfrear* for the folder name.
3. Change directory to the bfeare folder.
4. Unpack the web archive file by executing the following command:

```
jar xvf ../*.ear
```

Note

This command will unzip ANY EAR in the %STAGING% directory. If there are multiple EAR files, replace the “*.ear” with the name of the Banner Flexible Registration EAR file. ■

5. Create a new directory named “bfrwar”.

This is in preparation for unpacking the WAR file. The WAR file contains the Banner Flexible Registration web application.

6. Unpack the web archive file by executing the following command:

```
jar xvf ../*.war
```

All subsequent changes will be made against contents of the “bfrwar” directory.

Note

As with the EAR file, the WAR file can also be packed and unpacked using any tool for expanding and archiving ZIP files. ■

Step 2 Configure the Banner Flexible Registration web application

Modify the Banner Flexible Registration configuration settings are described below.

Modify the database connection string

As of Release 8.5 of Banner Flexible Registration, database connections are managed by OAS and are accessed through JNDI.

Configure symmetric key encryption

Banner Flexible Registration uses symmetric key encryption to increase the security of the SOAP requests and responses used. With symmetric key encryption, one key is used to both encrypt and decrypt messages. This key is configured in `/WEB-INF/security-applicationcontext.xml` file.

It is highly recommended that you change the value of this key to a value that is difficult to guess. To change the value of the key, perform the following steps:

1. Locate and edit the `/WEB-INF/security-applicationcontext.xml` file.
2. Find the bean with the ID of `flexRegCertificateEncryptionManager` and update the `<value>mysecretkey</value>`. The key can contain alpha-numeric characters and can be any length.

For example:

```
<bean id="flexRegCertificateEncryptionManager"
class="net.hedtech.flexreg.framework.security.FlexRegCertificateStaticKeyEncryptionManagerImpl">
  <constructor-arg>
    <value>xy8234fsdfv80sk3</value>
  </constructor-arg>
</bean>
```

3. Save the `/WEB-INF/security-applicationcontext.xml` file.

Step 3 Repackage the EAR file

Using an appropriate tool, repackage the files into the respective WAR and EAR files.

Step 4 Back up the EAR file

Make a backup copy of the EAR file to a directory of your choice for future reference.

Install Banner Flexible Registration on Oracle 11g

This section details the steps required to perform a new install of Banner Flexible Registration on Oracle 11g.

Note

If you have an existing Banner Flexible Registration version and would like to upgrade to Banner Flexible Registration 8.6, see [“Upgrade to Banner Flexible Registration on Oracle 11g” on page 2-10.](#) ■

Note

Please refer to the Banner Middle Tier Implementation Guide for recommendations on how to deploy Java web applications alongside Internet Native Banner and Banner Student Self Service. ■

Note

Depending on the configuration of the server, you may have to manage changes to the Application Server through the Change Center. If this level of control is enabled on the server, you may have to select **Lock & Edit** before making changes, and then commit those changes by clicking **Release Configuration.** ■

Step 1 Complete all steps in the Banner Flexible Registration Upgrade Guide

Access the Banner Flexible Registration 8.6 Upgrade Guide (`flexregxxxxxug.pdf`) and complete all steps in the guide.

Step 2 Configure WebLogic Server

Create a new Managed Server as described in the Banner Middle Tier Implementation Guide.

Refer to *FAQ- 1-13F9EZA* for more detailed instructions for setting up a new domain and server.

Step 3 Add a data source

This step provides details for installing the Flexible Registration application into an Oracle 11g Weblogic application server. These instructions assume that the appropriate `sfxxxxxu.trz` file has been downloaded and decompressed using Ellucian’s `unlded` tool into the `%Staging%` directory.

1. Using a web browser, connect to the Oracle WebLogic Server using the `admin` account established on the server.
2. Navigate to Domain Structures > Services > JDBC > Data Sources.

3. Select **New**.
4. Enter information under Configuration.
 - 4.1. Enter *FlexibleRegistrationDS* as the Name.
 - 4.2. Enter *jdbc/flexregdb* as the JNDI Name.
 - 4.3. Click **Next**.
 - 4.4. Select *Oracle's Driver (Thin) for instance connection* as the Database Driver.
5. Click **Next**.

Ensure the **Support Global Transactions** indicator is not checked.
6. Click **Next**.
7. Enter information under Connection Properties.
 - 7.1. Enter *Banner* as the Database Name.
 - 7.2. Enter *101.55.55.55* as the Host Name.
 - 7.3. Enter *1523* as the Port.
 - 7.4. The Oracle username defaults to *FLEXREG_USER* for the Database User Name.
 - 7.5. Enter the chosen password as the Password.
8. Click **Next**.
9. Test the database connection.
10. If the connection is good, click **Finish**.
11. Select the newly created data source.
 - 11.1. Select **Target**.
 - 11.2. Select the appropriate target server Flexible Registration will be deployed to.
 - 11.3. Click **Save**.

Step 4 Configure the email notification message queue

The email notification message queue must be configured next so that Banner Flexible Registration can use the required JMS connection factory which is specified in the message queue configuration. Deployment of the EAR file will fail if this is not done.

Refer to the [“Configure OAS 11g for email notifications”](#) section in [Chapter 3, “System Configuration”](#) for instructions on message queue configuration.

Step 5 Deploy the Flexible Registration web application archive

1. Navigate to Domain Structures > Deployments.
2. Click **Install**.
3. Enter the path to locate the EAR.
4. Select the EAR and click **Next**.
5. Click **Next**.
6. Select the appropriate target server Flexible Registration will be deployed to.
7. Click **Finish**.

Step 6 Start the Flexible Registration application

Select and start the Banner Flexible Registration application. When the application has finished deploying, the installation process is complete.

Note

Log files can be used to assist with debugging the application if errors occur. Under normal circumstances the files contain application specific logging messages. The following log files are created:

```
FlexReg_Services.log  
FlexReg_Expiration_Services.log  
FlexReg_MessageSender_Services.log
```

The location of these log files may differ based on your particular Oracle Application Server configuration, but likely locations include the following:■

- %DOMAIN_HOME%
- %DOMAIN_HOME%/%
- DOMAIN%/servers/%SERVER%/logs
- %BFR_HOME%/log

See [Chapter 3, “System Configuration”](#) for more details.

Step 7 (Optional) Configure mod_wl_ohs module

The Flexible Registration application can be accessed through the Oracle HTTP Server by defining a mod_wl_ohs URL routing configuration. This allows you to advertise a URL that is hosted via Oracle HTTP Server, which then routes requests to the application server

where Flexible Registration is deployed. The `mod_wl_ohs` module allows requests to be proxied from an Oracle HTTP Server to Oracle WebLogic Server.

The following assumes that an SSL port is available in the Oracle HTTP Server 11g. If not, please do so by following the section *Enabling SSL in WebLogic Oracle Fusion Middleware 11gR1 and 11gR2* in the latest *Banner Middle Tier Guide*.

Use the following steps to configure the `mod_wl_ohs` module using Fusion Middleware Control.

1. Log in to Oracle Enterprise Manager (OEM).
2. Navigate to the Oracle HTTP Server home page.
3. Select Administration -> `mod_wl_ohs` Configuration from the Oracle HTTP Server menu.

The `mod_wl_ohs` Configuration page is displayed.

4. In the **Locations** area, specify a location override for flexreg as follows:
 - 4.1. Click Add Row to create a new row.
 - 4.2. Enter `/flexibleregistration` in the **Location** field.

 **Note**

Note: `/flexibleregistration` is the context root value in the `application.xml` which resides in the `META-INF` directory of the EAR file. ■

- 4.3. Enter the WebLogic Server name in the **WebLogic Host** field.
- 4.4. Enter the port where the application is deployed in the **WebLogic Port** field.
- 4.5. Enter the path and file name for the log file in the **Log File** field. You must create this directory before setting this parameter.
- 4.6. Specify the type of logging performed for debugging operations in the **Debug** field.

Debugging information is written to the `/tmp/wlproxy.log` file on UNIX systems and to the `c:\TEMP\wlproxy.log` file on Windows systems.

- 4.6.1. Override this location and file name by setting the Log File parameter to a different directory and file.
- 4.6.2. Ensure that the `tmp` or `TEMP` directory has write permission assigned to the user who is logged in to the server.

5. Review the settings:
 - 5.1. If the settings are correct, click **Apply** to apply the changes.
 - 5.2. If the settings are incorrect or if you decide to not apply the changes, click **Revert** to return to the original settings.
6. Restart the Oracle HTTP Server.

The `mod_wl_ohs` module configuration is saved and is shown on the `mod_wl_ohs` Configuration page.
7. Test your Flexible Registration with the newly set up URL.

Upgrade to Banner Flexible Registration on Oracle 11g

This section details the steps required to upgrade from Banner Flexible Registration 8.5.5 to Banner Flexible Registration 8.6 on Oracle 11g.

Note

If you would like to perform a new install of Banner Flexible Registration 8.6, see [“Install Banner Flexible Registration on Oracle 11g” on page 2-6](#).

Note

Depending on the configuration of the server, you may have to manage changes to the Application Server through the Change Center. If this level of control is enabled on the server, you may have to select **Lock & Edit** before making changes, and then commit those changes by clicking **Release Configuration**.

Step 1 Complete all steps in the Banner Flexible Registration Upgrade Guide

Access the Banner Flexible Registration 8.6 Upgrade Guide (`flexregxxxxxug.pdf`) and complete all steps in the guide.

Step 2 Add a data source

Since version 8.5, Banner Flexible Registration uses JNDI to manage data connections. This step is only necessary if you did not create the JNDI data connection per the 8.5 version of the *Banner Flexible Registration Administration Guide*.

If you already have the JNDI data connection defined, skip to the next step.

1. Using a web browser, connect to the Oracle WebLogic Server using the `admin` account established on the server.
2. Navigate to Domain Structures > Services > JDBC > Data Sources.
3. Select **New**.
4. Enter information under Configuration.
 - 4.1. Enter *FlexibleRegistrationDS* as the Name.
 - 4.2. Enter *jdbc/flexregdb* as the JNDI Name.
 - 4.3. Click **Next**.
 - 4.4. Select *Oracle's Driver (Thin) for instance connection* as the Database Driver.
5. Click **Next**.
6. Ensure that the **Supports Global Transactions** indicator is not checked.
7. Click **Next**.
8. Enter information under Connection Properties.
 - 8.1. Enter *Banner* as the Database Name.
 - 8.2. Enter *101.55.55.55* as the Host Name.
 - 8.3. Enter *1523* as the Port.
 - 8.4. The Oracle username defaults to *FLEXREG_USER* for the Database User Name.
 - 8.5. Enter the chosen password as the Password.
9. Click **Next**.
10. Test the database connection.
11. If the connection is good, click **Finish**.

12. Select the newly created data source.

12.1. Select **Target**.

12.2. Select the appropriate target server Flexible Registration will be deployed to.

12.3. Click **Save**.

Step 3 Configure the email notification queue

The email notification message queue must be configured next so that Banner Flexible Registration can use the required JMS connection factory which is specified in the message queue configuration. Deployment of the EAR file will fail if this is not done.

If you already have an e-mail notification queue defined, skip to the next step.

Refer to [“Configure OAS 11g for email notifications” on page 3-11](#) for instructions on message queue configuration.

Step 4 Undeploy existing Flexible Registration Web Applications

This document assumes the use of the Oracle Web Logic Server Administration Console to undeploy the existing application. This step de-installs the existing Flexible Registration application from an Oracle 11g Server using the Oracle Web Logic Server Administration Console.

1. Navigate to Domain Structures > Deployments.
2. Select the check box next to Banner Flexible Application Enterprise Application.
3. Click **Delete** to remove the deployment.
4. Click **Yes** to confirm the removal.

Step 5 Deploy the Flexible Registration web application archive

1. Navigate to Domain Structures > Deployments.
2. Click **Install**.
3. Enter the path to locate the EAR.
4. Select the EAR and Click **Next**.
5. Click **Next**.
6. Select the appropriate target server Flexible Registration will be deployed to.
7. Click **Finish**.

Step 6 Start the Flexible Registration application

Select and start the Banner Flexible Registration application. When the application has finished deploying, the installation process is complete.

Note

Log files can be used to assist with debugging the application if errors occur. Under normal circumstances the files contain application specific logging messages. The following log files are created:

```
FlexReg_Services.log
FlexReg_Expiration_Services.log
FlexReg_MessageSender_Services.log
```

The location of these log files may differ based on your particular Oracle Application Server configuration, but likely locations include the following:

- %DOMAIN_HOME%
- %DOMAIN_HOME%/%
- DOMAIN%/servers/%SERVER%/logs
- %BFR_HOME%/log



3 System Configuration

This chapter details configuration information for the Banner Flexible Registration application.

Application Home

The application home folder allows you to define a standard location for the application configuration files outside of the application server.

Why Application Home is needed?

The Application Home folder allows you to define a standard location for the application configuration files outside of the application server.

Institutions may need to customize the application configuration files. In the previous versions of Banner Flexible Registration, customizing the application configuration files was complex. Application servers like WebLogic leave the EAR file compressed. So, modification to the configuration files would mean decompressing the EAR file, modifying the configuration file, and compressing it back to the EAR format again. This approach is both risky and complex.

The Application Home provides a means for clients to configure technical aspects of the application without the cost and risk of modifying the EAR file.

Application Home folder structure

This section describes how the resource folder is organized. Here is list of the folders in the resource folder:

Folder Name	Purpose
rootfolder	This is the top level folder (bfr_home).
config	This folder contains all the configuration files like log4j.properties.
log	This folder contains all BFR log files.

Setting up Application Home

The following steps describe how to set up Application Home.

Step 1 Set up BFR_HOME folder

Create a folder that will contain the configuration files. This folder must be accessible to the application server that is the container for Banner Flexible Registration. Additionally, the account under which the application server is running must have access to the folder and the files in the folder.

Step 2 Define the BFR_HOME environment variable

Define an operating system level environment variable called BFR_HOME and provide the path defined in Step 1 as its value. This environment variable is used by Banner Flexible Registration to locate the BFR_HOME location and read external configurations, write log files to log folder, etc.

Refer to your operating system guide for information on how to define an environment variable.

See [Appendix E, “Banner Flexible Registration Log Files”](#) for more details.

Banner Flexible Registration timeout configuration

The user can configure the user session and token timeout value for Banner Flexible Registration or use the default values. These two timeout values should not be the same value. The user session timeout value is specified in the Flexible Registration Catalog Rules Form (SFRAC TLG).

Configure user session timeout

The user session timeout dictates how much time the users can remain idle before the application logs them out. To configure the user session timeout, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRAC TLG).
2. In the **Value** field against the **Number of minutes idle before logout** parameter, specify the number of minutes before a timeout will occur.

Note

The default value for the user session timeout is 30 minutes. ■

3. In the **Value** field against the **Number of minutes before warning window** parameter, specify the number of minutes before a timeout that a warning will display.

For example, if the timeout is 30 minutes, setting the warning to 1 will display a warning to the user after 29 minutes of inactivity.

 **Note**

Partial minutes are also supported. For example, 0.5 is equal to 30 seconds. ■

Token timeout value

The token timeout value has a default value of 480 minutes.

 **Note**

The user session timeout value and the token timeout value should not be the same. The token timeout value must be substantially longer than the user session timeout value by enough time such that the user can reset the user session many times before the token timeout expires. Otherwise, the user may receive an error due to an expired token. ■

Cart expiration process

The cart expiration process is a batch processing job that runs on a customized schedule that cleans up any carts that have not been completed. As a student proceeds through the checkout process, Flexible Registration holds a seat for the student for each class in the shopping cart. The shopping cart still holds the seat even if the student does not complete the checkout process.

The cart expiration process job resolves this discrepancy. It removes the student from holding the seats if the student does not complete the payment, and the cart is marked as canceled. Payment refers to satisfying a payment method such as a credit card payment, or even electing the **Pay Later** option.

When the cart expiration process is in progress, it takes all of the carts that do not have a status of *F'inialized* or *D'eleted* and determines if they are still active. Those carts that are still active are ignored by the cart expiration process. Those carts that are no longer active (inactive/expired) are processed by the cart expiration process. Processing of the expired carts results in the individual cart being *D'eleted* or *F'inialized* by the process.

When the cart expiration process is in progress, and an open cart is being canceled or updated to *F'inialized*, if there are any open payment transactions found, it cancels those open payment transactions.

You can view the carts with the Cart Inquiry (SFAICART) form and in conjunction with the *FlexReg_Expiration_Services.log* file. You can discover and track the status of a cart through this process.

There are several rules to follow as part of the cart expiration process in determining what status to update the cart to:

- If the **Override automatic drop of expired carts** or **Pay Later** settings on the Flexible Registration Catalog Rules Form (SFRAC TLG) are set to *Y*:
 - If the cart has been confirmed (and is not already *Finalized*), the cart status is updated to *Finalized*.
 - If the cart has not yet been confirmed (and is not already finalized), the cart status is updated to *Deleted*.
 - For Group Registration, if the cart has been confirmed (and is not already *Finalized*), the expiration process calls the Bill Me payment process before finalizing the cart. The payment adjustments are made for group registration carts whose status changes to *Finalized* through the cart expiration process.
- If the **Override automatic drop of expired carts** and **Pay Later** settings on the Flexible Registration Catalog Rules Form (SFRAC TLG) are set to *N*:
 - If the cart has not yet been finalized, the cart status is updated to *Deleted*.
- If the cart was in the *Awaiting Payment* state, but through a check in Banner, payment was received, the cart will be *Finalized*.

Cart expiration automated process configuration

The Cart Expiration automated process provides Banner Flexible Registration with the ability to expire carts that were not fully processed by the application through normal user interaction. For more information please see the FAQ on the Customer Support Center.

The cart expiration process is configured in the `cartexpiration-applicationcontext.xml` file.

Default values

The default values from the configuration file are as follows:

Property Name	Purpose
defaultCartExpirationLifeMin	The minimum time between when the cart was created and the current time before the cart will be expired. (Default is 30 minutes)
defaultCartExpirationLifeAwaitingPaymentMin	The minimum time between when the cart was created and the current time before the cart will be expired. (Default is 60 minutes)
repeatInterval	The frequency of when the process will execute. (Default value is 3,600,000 milliseconds = 1 hour)

The default configuration deployed in the application is the suggested configuration that most institutions should use unless their environments warrant more frequent processing.



Warning

No other parts of this configuration file should be modified in any way. ■

Cart expiration

There is no need for your institution to modify the following values once the Banner form values on (SFRACNFG) have been configured.

```
<property name="defaultCartExpirationLifeMin" value="30" />
<property name="defaultCartExpirationLifeAwaitingPaymentMin"
value="60" />
```



Note

Once these values are configured in Banner, any manual changes to these values will be ignored. ■

Repeat interval

The following property may be modified to alter the frequency of cart expiration processing:

```
<property name="repeatInterval" value="3600000" />
```



Note

Any changes to the cart expiration configuration file will require a restart of the Banner Flexible Registration application before those changes will be used by the application. ■

Logging

The cart expiration process provides logging messages to a custom log file that can be used to research any issues. The following is the name of the log file:

```
FlexReg_Expiration_Services.log
```

File Attachments

Banner Flexible Registration provides the ability for institutions to enable students to upload file attachments during the registration process. This enables institutions to collect documents electronically and helps ensure that students are fully prepared to begin their learning activities by the time a class begins.

Banner Flexible Registration, currently, only supports a local file system for the storage of attachments. In order to fully enable this new feature, the following considerations and steps must be taken to properly configure the application.

During a group registration, the attachment is stored based on the above rules for the student ID for which the attachment is being uploaded. For example, if a cart owner is adding one attachment for each of the five students, a Banner student ID folder will be created for each of the five students and the attachment will be added for each student ID. Attachments would be added as if the students added the attachments themselves.

The following steps describe how forms are stored in the folder:

1. A folder is created (if it does not already exist) for the catalog code.
2. A folder is created (if it does not already exist) for each Banner Student ID under the term folder.
3. The attachment is stored in the folder with the name of the Student ID. The file is renamed with the following syntax:

```
[Student ID][Term][CRN][Attachment Code][Cart ID].filetype
```

Considerations

When a learner uploads a file to the institution, however, this exposes the institution to potential security risks that you should be aware of. The following section outlines various considerations you should take when configuring this feature.

Virus protection

Currently, Flexible Registration supports only storing file attachments on a file system that can be reached from your application server. Storing documents from external users always poses a risk and this risk should be given its proper attention.

Virus scans should be done regularly on the file system that stores the uploaded files.

Limit file system permissions

The file system (or directory) that will contain the uploaded attachments should only have the minimal set of permissions required to upload and view the attachment files.

The Flexible Registration application container will need permissions to write to the directory when a file is uploaded and when the file is read for download.

Limit the extensions allowed to upload

The application ships with allowing only certain file extensions to be uploaded. This will protect institutions for allowing command files or programs that can be executed from being uploaded. To enable a file extension to be uploaded it must explicitly defined. This does not provide 100% protection from an executable file from being uploaded but it does provide a layer of security.

Altering this list of extensions is not supported or recommended. Allowing learners to upload executable files will put your infrastructure at risk.

Load Balancing

If you are using the application in a load balanced environment, a common file share is recommended. This would enable you to create a single file system that each of the application servers write to. This will give you the ability to manage files in a single location.

Set attachment upload folder

To configure the location of where the attachments will be stored, access the SFRACNFG form and enter the fully qualified path to the attachment storage folder in the **Physical path for attachments** field.

If this directory is not setup properly, students will receive error messages while attempting to upload attachments.

Setup attachment download settings

In order for the attachments to be successfully downloaded from the SFFRATCH form, by administrators, the location of the Flexible Registration application must be supplied. To configure the location of the Flexible Registration deployment URL, access the SFRACNFG form and enter the URL path for the application in the Flexible Registration

Deployment URL field. This value should simply be the root url and is to not include page names (like index.jsp).

Good Value Example: <https://ellucianuniversity.edu/flexibleregistration>

Bad Value Example to not use: <https://ellucianuniversity.edu/flexibleregistration/index.jsp>

If the application is configured in a load balanced environment, simply use the URL that you publish to learners here (and not the url for a specific application server).

Fee Adjustment Process

The Fee Adjustment Process (SFKRGADJ) is a Flexible Registration process that is used to adjust fees during a group registration to ensure that the system properly accounts for fees between the payer of the classes and the students being registered. This process is integrated into the Banner Job Submission module and can be accessed using the Fee Adjustment Process (SFKRGADJ) job name. Please review [“Fee assessment” on page 4-41](#) for more details.

The Fee Adjustment Process needs to be configured to properly execute and generate the required log and report files that are normally produced with Banner Job Submission.

The Fee Adjustment Process (SFKRGADJ) will generate *.lis* and *.log* files which will show the results of the process execution. These files are then loaded into Banner for viewing from the saved output review form (GJIREVO). In order for these files to be created in the correct location and loaded for viewing, the following considerations need to be made and settings reviewed.

- A script or process file is used to interface between the Banner Job Submission and the Flexible Registration application. This script file will call back into a Flexible Registration database package to perform the process, when executed.
- This script file will generate the *.log* file which contains process execution information. The process within Flexible Registration will create a *.lis* file that contains a report or summary of the work performed during the process. The *.log* file and the *.lis* must be in the same output directory that the script expects such that the script can properly load the information into the table needed by the GJIREVO form.
- The output directory that the *.log* file will be created in is determined by the settings in your environment for Job Submission.
- The output directory that the *.lis* file will be created in is determined by the settings in the Oracle directory called *FLEXREG_JOBSUB_DIR*. This directory should have been defined and set up during installation. The physical directory path should be reviewed and updated if needed.

- If the directory path needs to change on the Oracle directory, use the `CREATE` or `REPLACE DIRECTORY` SQL command.
- Having both the paths set up such that the `.log` and `.lis` file are created in the same location ensures the Job Submission files can be viewed as expected by the person executing the process.
- If Banner Job Submission is running under a Microsoft Windows environment, the PERL version of the script will be used. The `sfkrgadj.pl` file should have been installed and placed in the proper directory for execution. Please verify the existence of this file on your Job Submission system if there is ever any issue starting the process. On Microsoft Windows, the `sfkrgadj.pl` file may be located in the `\general\misc` of your Banner code tree.
- If you are not running Microsoft Windows, the script file that will be used will be the `sfkrgadj.shl` shell script. Again, please ensure this file is located in the proper directory for execution. On some Linux environments, this could be in your `links` (env `BANNER_LINKS`) directory.

Self-Service Banner course details link configuration

Banner Flexible Registration provides a link from the Section Details dialog to the Detailed Course Information page (`bwckschd.p_disp_detail_sched`) in Banner Student Self-Service. When the Section Details dialog is open for a specific section, select the Restrictions option, and then click on the link that is displayed to access Self-Service.

Use the Flexible Registration Configuration Rules Form (SFRACNFG) to configure access to Banner Student Self-Service by setting up Self-Service URLs.

Email notification

Banner Flexible Registration provides the ability to send emails to users in response to certain events. This processing uses Message Queues to manage and coordinate the sending of emails. Banner Flexible Registration uses a queue managed on the same database server as Banner to ensure that emails will be sent if the application server crashes.

Supported events

The following event is supported by email notification processing.

Event	Release	Description
Registration Confirmation	8.5	Sends the user a registration confirmation email.
Group Registration Confirmation	8.6	Sends the group cart owner a registration confirmation email.

Create emails

Use the SFRAEMAL form to define the email type (plain text or HTML) and the email content.

Configure notification by Flexible Registration catalog

Use the SFRACLTG form to specify which emails (defined in SFRAEMAL) to send. Each Flexible Registration catalog must be configured. A given catalog may have multiple emails associated with a given event. Time boxing can be used to determine which emails to send and to send multiple emails to a client for any given event.

The **Sender Name** field on SFRACLTG is optional. If a value is entered, it will be displayed on the email message for who the email is from. The “from” address may also be displayed in brackets after the sender name. If the sender name is not specified, the value in the **From Address** field on SFRACLTG will be displayed on the email message for who the email is from. The **From Address** field is a required field and must contain a value that specifies an existing domain. For example, if the **From Address** value is *myName@myDomain.com*, “myDomain” must be a domain that actually exists. Otherwise emails will fail to be sent, and errors will result in the `FlexReg_MessageSender_Services.log` file.

Email execution interval and the JMS queue

The `email-context.xml` file specifies the configurations that are specific to email notification processing in Banner Flexible Registration. It is configured with the Quartz timer in Spring to execute like a batch job on a given time interval. So, on the time interval specified, this class will be executed and will attempt to send any emails that are in the JMS queue. The parts of this file that are important are shown below:

```
<beans>
...
<!-- Detail of how often we call the send email process bean
-->
<bean id="sendEmailTrigger"
class="org.springframework.scheduling.
quartz.SimpleTriggerBean">
```

```

        <property name="jobDetail"
ref="batchSendEmailMethodFactory
    Bean" />
        <property name="startDelay" value="10000" />

        <!-- in milli -->
        <!-- 1 hour = 3600000 -->
        <property name="repeatInterval" value="15000" />
    </bean>
    ...
</beans>

```

The time interval in which Banner Flexible Registration sends emails is specified in the `repeatInterval` property of the `sendEmailTrigger` bean. This `repeatInterval` value is specified in milliseconds. In the example above, it is set to 15000 to run every 15 seconds.

There is also a `startDelay` property also specified in milliseconds. This property determines how long to wait immediately after the application is deployed or started before executing this class for the first time. It is set in the example above to wait 10 seconds before running, after which it runs the class at the given `repeatInterval` value.

Both the `repeatInterval` and `startDelay` properties can be changed to the desired values (in milliseconds), if the values initially provided are not suitable for your needs. Any such changes to the Spring configuration files in Banner Flexible Registration should be made in a text editor (like Notepad) and not in Microsoft Word, as Microsoft Word may insert invalid characters.

 **Note**

If Oracle 11g Weblogic is used, the Banner Flexible Registration EAR file should be unpacked so that these changes can be made. The EAR file should then be repackaged and deployed to the application server. ■

Configure OAS 11g for email notifications

This section provides technical details for configuring the Oracle Application Server 11g (Weblogic) to expose the `FLEXREG_EMAIL_Q` message queue to the Banner Flexible Registration application.

The steps below must be completed before starting the Banner Flexible Registration EAR file. Otherwise, the application will fail to start.

These instructions assume you are using the Weblogic Administration Console, and that you are using an account with sufficient privileges to make modifications.

Screen shots are provided to assist with the configuration steps. These are examples and may vary from what you see at your site.

Step 1 Ensure that a data source has been added

A JDBC data source is required before configuring email notification. If one has not been added, please refer to [Step 3, “Add a data source”](#) in the [“Install Banner Flexible Registration on Oracle 11g”](#) section of [Chapter 2, “Installation”](#).

Step 2 Create JMS connector

1. Using the WebLogic Server Administration Console, locate the **Domain Structure** tree view component.
2. Select these pages in order: Services, Messaging, and JMS Modules.
3. Click **New** to create a new JMS Module.
4. Enter *FlexReg-JMS-Module* in the **Name** field.

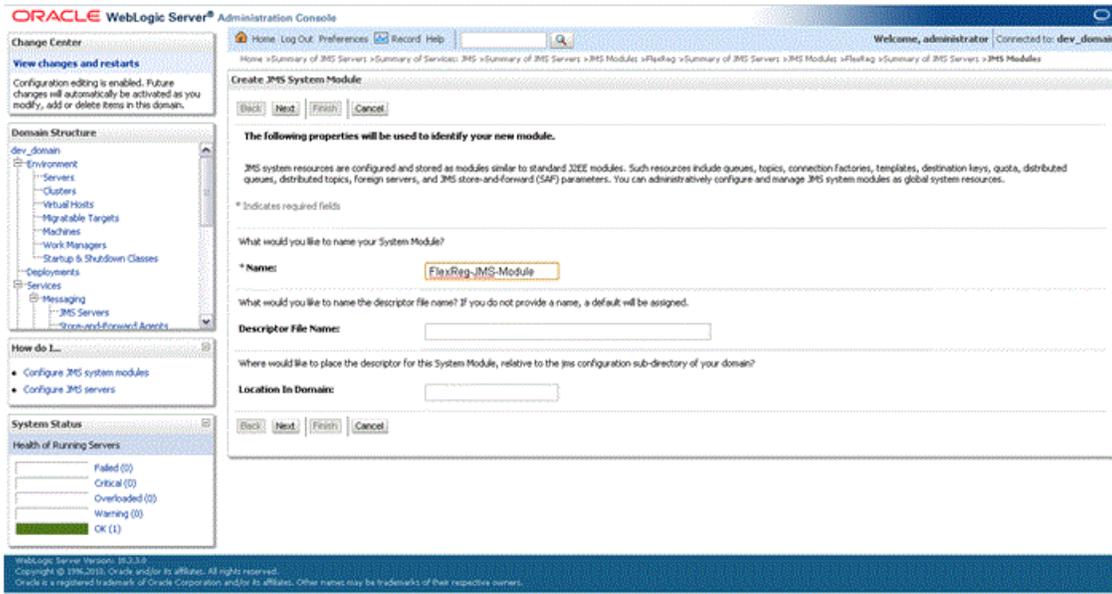
Note

The **Descriptor File Name** and **Location In Domain** fields are optional and remain empty. ■

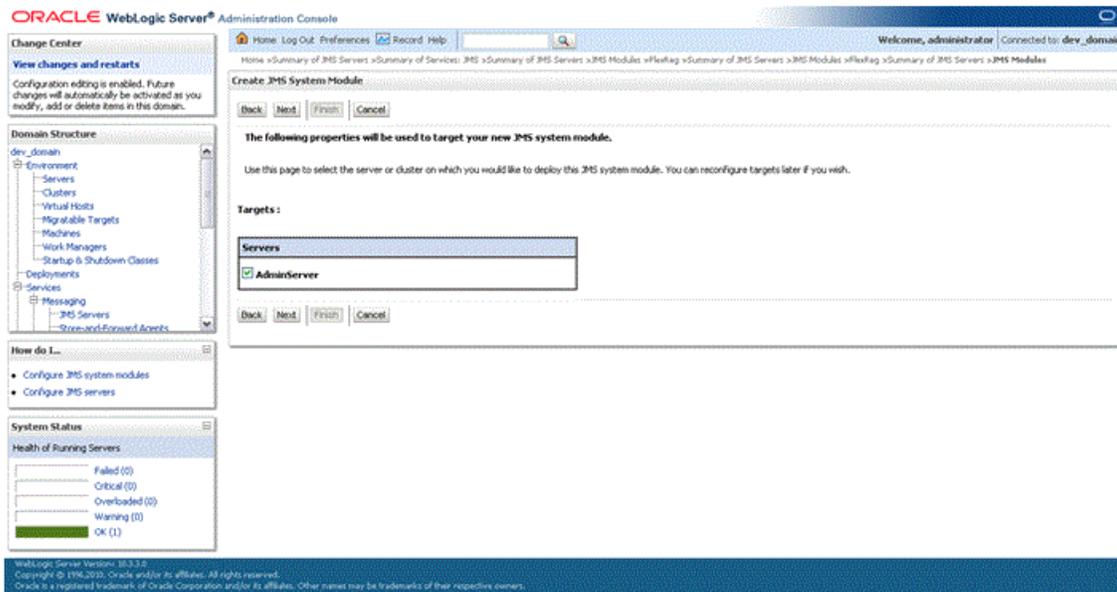
5. Click **Next**.
6. Select the appropriate target server Flexible Registration will be deployed to.
7. Click **Next**.
8. Leave the **Add Resources** check box unchecked.
9. Click **Finish** to create the module.

The following screenshots illustrate adding the JMS module and targeting the domain server.

Add a JMS module



Target the domain server for the JMS module



Step 3 Add a foreign server resource

1. Using the WebLogic Server Administration Console, locate the **Domain Structure** tree view component.
2. Select these pages in order: Services, Messaging, and JMS Modules.
3. Select the Configuration tab.
4. Under the **JMS Modules** section, click *FlexReg-JMS-Module*, which is the module you created above.
5. Under **Summary of Resources**, click **New**.
6. Under **Choose the type of resource you want to create**, select the **Foreign Server** option.
7. Click **Next**.
8. Enter *FlexRegForeignServer* in the **Name** field.
9. Click **Next**.
10. Select the appropriate target server Flexible Registration will be deployed to.
11. Click **Finish** to complete the creation of the foreign server resource.

The following screenshots illustrate adding the adding the foreign server, naming it, and targeting it to the domain server.

Select resource type of foreign server

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled "Create a New JMS System Module Resource". Below the title bar, there are navigation buttons: "Back", "Next", "Finish", and "Cancel".

The main section is "Choose the type of resource you want to create." It includes a sub-section "Use these pages to create resources in a JMS system module, such as queues, topics, templates, and connection factories." and a paragraph explaining that depending on the resource type, basic information for creation is required, and for targetable resources, target pages can be used.

There are seven radio button options:

- Connection Factory: Defines a set of connection configuration parameters that are used to create connections for JMS clients. [More Info...](#)
- Queue: Defines a point-to-point destination type, which are used for asynchronous peer communications. A message delivered to a queue is distributed to only one consumer. [More Info...](#)
- Topic: Defines a publish/subscribe destination type, which are used for asynchronous peer communications. A message delivered to a topic is distributed to all topic consumers. [More Info...](#)
- Distributed Queue: Defines a set of queues that are distributed on multiple JMS servers, but which are accessible as a single, logical queue to JMS clients. [More Info...](#)
- Distributed Topic: Defines a set of topics that are distributed on multiple JMS servers, but which are accessible as a single, logical topic to JMS clients. [More Info...](#)
- Foreign Server: Defines foreign messaging providers or remote WebLogic Server instances that are not part of the current domain. [More Info...](#)
- Quota: Controls the allotment of system resources available to destinations. [More Info...](#)
- Destination Sort Key: Defines a unique sort order that destinations can apply to arriving.

On the left side, there is a "Domain Structure" tree showing the hierarchy: dev_domain > Environment > Servers > Clusters > Virtual Hosts > Migratable Targets > Machines > Work Managers > Startup & Shutdown Classes > Deployments > Services > Messaging > JMS Servers > **Foreign Servers**.

Below the domain structure is a "How do I..." section with a list of links: Configure quotas for destinations, Configure JMS templates, Configure destination keys, Configure topics, Configure queues, Configure connection factories, Configure uniform distributed topics, Configure uniform distributed queues, Configure foreign servers, and Configure JMS SAF.

Name the foreign server

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled "Create a New JMS System Module Resource". Below the title bar, there are navigation buttons: "Back", "Next", "Finish", and "Cancel".

The main section is "Foreign Server Properties". It includes a sub-section "The following properties will be used to identify your new foreign server. The current module is FlexReg-JMS-Module." and a note "* Indicates required fields".

There is one required field:

- * Name: FlexRegForeignServer

Below the field, there are navigation buttons: "Back", "Next", "Finish", and "Cancel".

On the left side, there is a "Domain Structure" tree showing the hierarchy: dev_domain > Environment > Servers > Clusters > Virtual Hosts > Migratable Targets > Machines > Work Managers > Startup & Shutdown Classes > Deployments > Services > Messaging > JMS Servers > **Foreign Servers**.

Below the domain structure is a "How do I..." section with a list of links: Configure quotas for destinations, Configure JMS templates, and Configure destination keys.

Target the foreign server to the domain server

ORACLE WebLogic Server® Administration Console

Home > FlexReg-JMS-Module > ForeignServer-0 > placeholder > FlexRegForeignServer > Configuration > FlexRegForeignServer > Configuration > JMS Modules > FlexReg-JMS-Module > placeholder

Welcome, administrator Connected to: dev_domain

Create a New JMS System Module Resource

Back Next Finish Advanced Targeting Cancel

The following properties will be used to target your new JMS system module resource

Use this page to view and accept the default targets where this JMS resource will be targeted. The default targets are based on the parent JMS system module targets. If you do not want to accept the default targets, then click **Advanced Targeting** to use the subdeployment mechanism for targeting this resource.

The following JMS module targets will be used as the default targets for your new JMS system module resource. If the module's targets are changed, this resource will also be retargeted appropriately.

Targets:

Servers
<input checked="" type="checkbox"/> AdminServer

Back Next Finish Advanced Targeting Cancel

Domain Structure

- dev_domain
 - Environment
 - Servers
 - Clusters
 - Virtual Hosts
 - Migratable Targets
 - Machines
 - Work Managers
 - Startup & Shutdown Classes
 - Deployments
 - Services
 - Messaging
 - JMS Servers
 - Store-and-Forward Agents

How do I...?

- Configure quotas for destinations
- Configure JMS templates
- Configure destination keys
- Configure topics

Foreign server added

ORACLE WebLogic Server® Administration Console

Home > FlexReg-JMS-Module > ForeignServer-0 > placeholder > FlexRegForeignServer > Configuration > FlexRegForeignServer > Configuration > JMS Modules > FlexReg-JMS-Module > placeholder

Welcome, administrator Connected to: dev_domain

Settings for FlexReg-JMS-Module

Configuration Subdeployments Targets Security Notes

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

Name: FlexReg-JMS-Module The name of this JMS system module. [More Info...](#)

Descriptor File Name: jms\FlexReg-jms-module-jms.xml The name of the JMS module descriptor file. [More Info...](#)

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

[Customize this table](#)

Summary of Resources

Name	Type	JNDI Name	Subdeployment	Targets
<input checked="" type="checkbox"/> FlexRegForeignServer	Foreign Server	N/A	Default Targeting	AdminServer

Showing 1 to 1 of 1 Previous Next

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)

Step 4 Edit the foreign server properties

You should already be on the Settings for FlexReg-JMS-Module page. If so, continue with step 4 below. If not, do the following.

1. Use the WebLogic Server Administration Console to locate the **Domain Structure** tree view component.
2. Select these pages in order: Services, Messaging, and JMS Modules.
3. Select *FlexReg-JMS-Module* from the list of available modules.
4. Select the Configuration tab if it is not already selected.
5. Click the *FlexRegForeignServer* link to open the setting page for the foreign server.
6. Select the General tab if it is not already selected.
7. Enter *oracle.jms.AQjmsInitialContextFactory* in the **JNDI Initial Context Factory** field.
8. Enter *datasource=jdbc/flexregdb* in the **JNDI Properties** field.
9. Ensure that the **Default Targeting Enabled** indicator is checked.

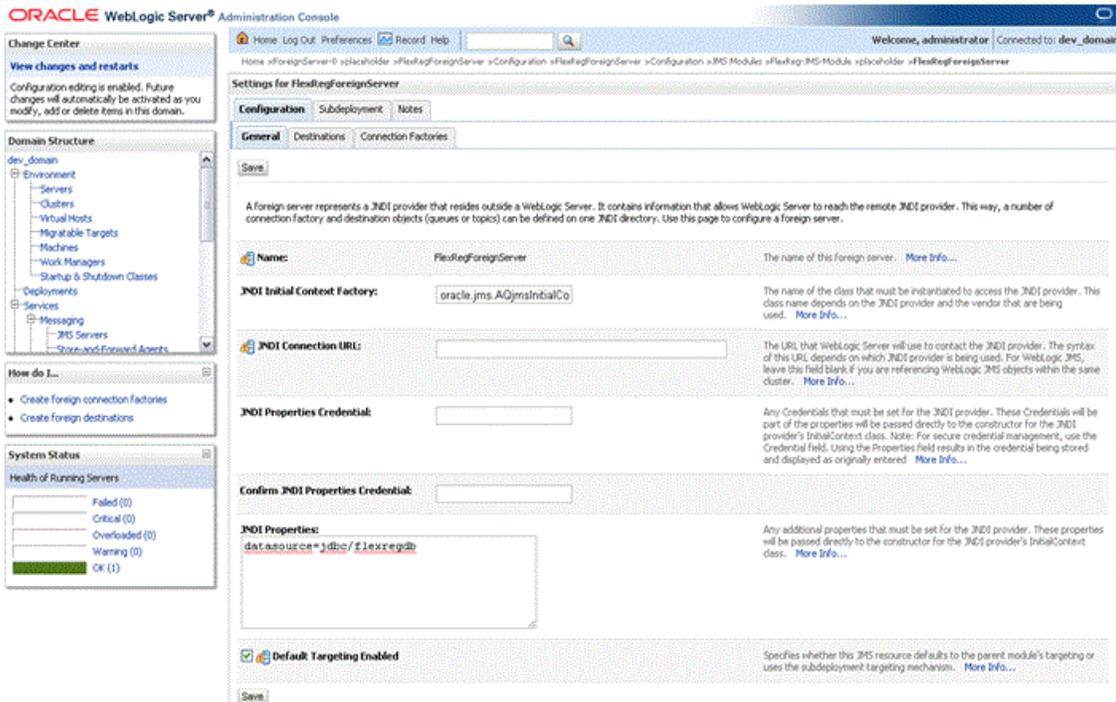
Note

The **JNDI Connection URL**, **JNDI Properties Credential**, and **Confirm JNDI Properties Credential** fields remain empty. ■

10. Click **Save**.

The following screenshot illustrates configuring the foreign server properties.

Configure Foreign Server Properties



Step 5 Specify a connection factory for the foreign server

1. Select the Connection Factories tab.
2. Click **New** to add a new foreign connection factory.
3. Enter *ForeignConnectionFactory* in the **Name** field.
4. Enter *jms/flexregQueueFactory* in the **Local JNDI Name** field.
5. Enter *QueueConnectionFactory* in the **Remote JNDI Name** field.
6. Click **OK**.

The following screenshots illustrate the foreign server connection factory configuration.

Add a Connection Factory to Foreign Server

ORACLE WebLogic Server® Administration Console

Home | Log Out | Preferences | Record | Help | Welcome, administrator | Connected to: dev_domain

Settings for FlexRegForeignServer

Configuration | Subdeployment | Notes

General | Destinations | **Connection Factories**

A foreign connection factory represents a connection factory that resides on another server, and which is accessible via JNDI. A remote connection factory can be used to refer to another instance of WebLogic Server running in a different cluster or server, or a foreign provider, as long as that provider supports JNDI.

This page summarizes the foreign connection factories that have been created for this domain.

Customize this table

Foreign Connection Factories (Filtered - More Columns Exist)

Name	Local JNDI Name	Remote JNDI Name
There are no items to display		

WebLogic Server Version: 10.3.3.0
Copyright © 1996, 2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Local and Remote JNDI Connection Factory Properties

ORACLE WebLogic Server® Administration Console

Home | Log Out | Preferences | Record | Help | Welcome, administrator | Connected to: dev_domain

Create a New Foreign JMS Connection Factory

OK | Cancel

Foreign Connection Factory Properties

The following properties will be used to identify your new foreign connection factory.

* Indicates required fields

* Name: ForeignConnectionFactory

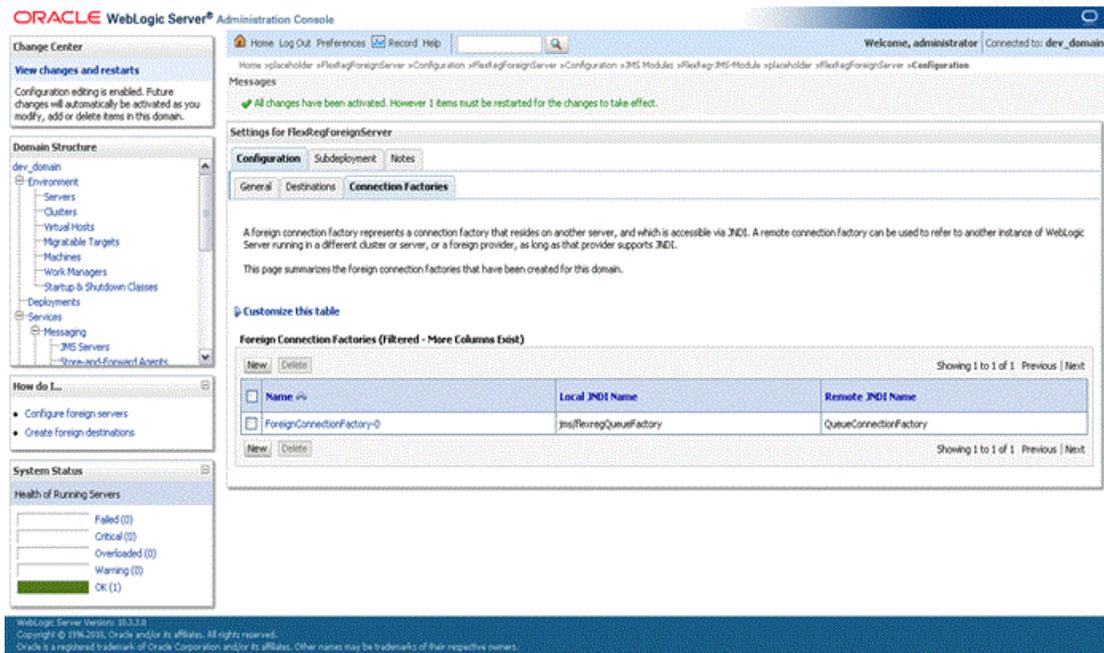
Local JNDI Name: jms/flexregQueueFactory

Remote JNDI Name: QueueConnectionFactory

OK | Cancel

WebLogic Server Version: 10.3.3.0
Copyright © 1996, 2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Foreign Server Connection Factory Added



Step 6 Specify a destination

1. Select the Destinations tab.
2. Click **New** to create a new Foreign Destination.
3. Enter *ForeignDestination-0* in the **Name** field, as the foreign destination name.
4. Enter *flexregQ* in the **Local JNDI Name** field.
5. Enter *Queues/BANINST1.FLEXREG_EMAIL_Q* in the **Remote JNDI Name** field.
6. Click **OK**.

The following screenshots illustrate the destination configuration.

Add a Destination to Foreign Server

The screenshot shows the Oracle WebLogic Server Administration Console. The main content area is titled 'Settings for FlexRegForeignServer' and has three tabs: 'Configuration', 'Subdeployment', and 'Notes'. The 'Destinations' sub-tab is selected. Below the tabs, there is explanatory text about foreign destinations and a 'Customize this table' link. A table titled 'Foreign Destinations' is shown, but it is empty. The table has columns for 'Name', 'Local JNDI Name', and 'Remote JNDI Name'. The status bar at the bottom indicates 'Showing 0 to 0 of 0' items.

Local and Remote JNDI Destination Properties

The screenshot shows the 'Create a New Foreign JMS Destination' dialog box in the Oracle WebLogic Server Administration Console. The dialog has 'OK' and 'Cancel' buttons at the top. The main section is titled 'Foreign Destination Properties' and contains the following text: 'The following properties will be used to identify your new foreign destination.' Below this, there are three input fields:

- 'Name': ForeignDestination-0
- 'Local JNDI Name': jms/flexreg
- 'Remote JNDI Name': Queues/BANNST1.FLEXREG_EMAIL_Q

 There are 'OK' and 'Cancel' buttons at the bottom of the dialog. The background shows the same console interface as the previous screenshot, but the dialog box is in the foreground.

Foreign Server Destination Added

The screenshot displays the Oracle WebLogic Server Administration Console interface. The main content area is titled "Settings for FlexRegForeignServer" and shows the "Destinations" tab. A message at the top indicates that all changes have been activated, but 1 item must be restarted. Below this, a table lists the foreign destinations:

Name	Local JNDI Name	Remote JNDI Name
ForeignDestination-0	jmsFlexreg	Queues[BANNERSTL.FLEXREG_EMAIL_Q]

The left sidebar shows the "Domain Structure" tree with "dev_domain" selected. The "System Status" section at the bottom left shows the health of running servers, with 1 OK and 0 in other states.

Debugging

Banner Flexible Registration provides the tools and information needed to diagnose and resolve many issues in the application.

For information about the log files created by Banner Flexible Registration web application, see [Appendix C. "Banner Flexible Registration EhCache Configuration"](#).

Banner Flexible Registration Configuration Validation Tool

Banner Flexible Registration includes a tool to validate your Banner Flexible Registration data and configuration. This tool only validates the database configuration elements, but does not validate that the web application has been correctly configured. For more information on configuring the web application, please see ["Install Banner Flexible Registration on Oracle 11g"](#) on page 2-6.

The output of this tool is very helpful when diagnosing configuration problems such as missing or invalid catalog settings. For more information on the Banner Flexible Registration Configuration Validation tool, please see [Appendix A, “Banner Flexible Registration Configuration Validation”](#).

Version reporting

This section discusses version reporting for Banner Flexible Registration. There are several areas of the application that comprise the entire application. They are: the user interface, the middle-tier, and the database. When Banner Flexible Registration is packaged into an EAR file, the same version is applied to the user interface and middle-tier parts of the application. The FlexReg module of the database should be the same version as that of the user interface and middle-tier, but other supporting database modules may have different version numbers.

The middle-tier contains several JAR files and one WAR file, each of which has its own version. All of these files reside in the EAR file of the application.

The database contains many modules and possibly patches (if patches have been applied).

Usage

You can access, save, and send version information from Banner Flexible Registration.

View version information

Version information for Banner Flexible Registration can be viewed by accessing the `version.jsp` page. Open a web browser and navigate to that page from the application context on the server (such as `http://server:port/flexibleregistration/version.jsp`). An XML listing of version information should then be displayed in the web browser with version information for all areas of Banner Flexible Registration.

Save version information

Version data can be saved to a file by selecting File>Save As from the File menu of your browser. The Firefox web browser defaults the file to be saved as an XML file, but the Internet Explorer browser defaults the file as a JSP page (`version.jsp`). Ensure that the file name used to save this data ends with a `.xml` extension. (Although it could be saved as a `.jsp` extension, as the contents of the file would remain the same.)

Send version information

It is also possible to send this data from the web browser as a link by selecting File>Send Link from the File menu of your browser. Depending on the browser being used, this data would then be sent in an email as a link or as an attached file.

Manage plans

Students are able to create a plan of saved courses that can be used at a later time to quickly add those courses to the Schedule Builder and attempt registration. These plans do not expire and at this time there is no interface into the maintenance of existing plans. You may decide how to manage or use saved plan data for research purposes.

The plan data is stored in the database in the following two tables in the `FLEXREG` schema:

- `SFFBPLAN` - The parent plan table that contains the plan name, plan ID, and the student PIDM associated with this plan.
- `SFFRPLAN` - The plan item table that contains the CRN and term code of courses that are part of an existing plan. This table contains the plan ID as a foreign key to identify the parent row in `SFFBPLAN`.

Before removing anything, you must review the data in these tables to determine what rows are eligible to remove based upon how old the data is and if the data is still relevant for use by the student. A database technician will be required to code the appropriate SQL statements to migrate rows out of these tables into whatever format you desire.

The following SQL Code will output all of the parent and child rows for each plan in a specific Banner Flexible Registration catalog:

```
select * from sffbplan A
      left join sffrplan B on
      A.sffbplan_pid = B.SFFRPLAN_PID
where A.sffbplan_ctlg_code = 'A0'
```

Externalizing resources

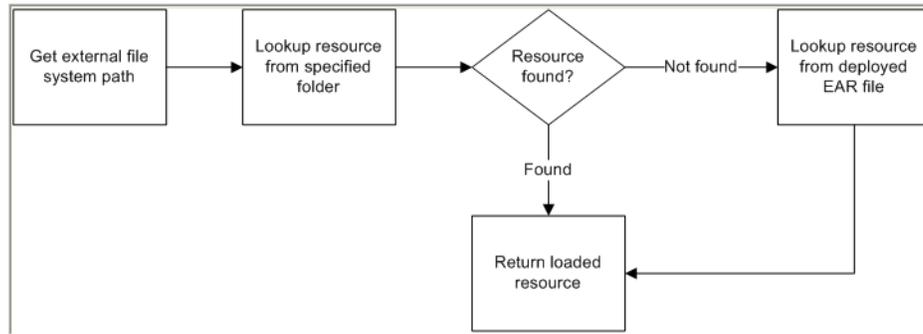
Clients may need to customize the images, messages, HTML content, or colors of the application. In previous versions of Banner Flexible Registration, customizing the look of the application was complex. Depending on the particular application server, the EAR may be decompressed to disk allowing for easy access to individual files such as Oracle Application Server 10.1.3. Other application servers may leave the EAR file compressed. In either case, if one or more resources were modified, the modification would be lost the next time the application is redeployed.

With this version, images, the message resources, HTML content, and cascading style sheet (css) files can be relocated to a separate resource folder. The server will check to see if the requested resource is in the resource folder before loading the resource from the deployed application.

This prevents user customization changes from being overwritten if an upgraded version of Banner Flexible Registration is installed.

When the application requests a resource, the server will first look in the resource folder. If the resource is found in the resource folder, the resource will be returned to the user's browser. If the resource is not found, then the request is processed by the application server as normal.

The diagram below illustrates the process that determines if a resource is retrieved from the resource folder or the EAR.



Supported resources

Not every resource or file that is deployed as part of the Banner Flexible Registration web application can be relocated to the resource. Only the folders listed below can be externalized.

Resource Folder structure

This section describes how the resource folder is organized. Here is list of the folders in the resource folder.

Folder Name	Purpose
resourceroot	This is the top level folder that is scanned for resources.
_shared	This folder contains the location of resources that are shared across all catalogs.
assets	This folder contains image assets.
assets\icons	This folder contains icon files.
assets\images	This folder contains image files.
attachments\css	This folder contains the files that control the look and feel of the file attachment window during the registration process.

Folder Name	Purpose
content	This folder contains the HTML content file(s) that displays in the home page of the application.
locale	This folder contains the <i>strings.properties</i> file for string resources.
navigation	This folder contains files for the overall navigation of the application.
navigation\css	This folder contains the style sheets for any navigation components, such as the header bar.
navigation\js	This folder contains the JavaScript files for any navigation components, such as the header bar.
styles	This folder contains the <i>styles_ascii.css</i> file for CSS styles.

The diagram below shows the complete folder structure.



Note

The structure under the `_shared` folder matches that of the deployed Banner Flexible Registration application. ■

Map folders to deployed application

The structure may seem to contain some redundancy, such as the `_shared` and `assets` folders. As an example, assume that an image exists that is accessible by the URL:

```
* http://ellucianuniversity.edu/flexibleregistration/assets/images/example.png
```

A customized version of this image (with the same name and extension) would be placed in the folder:

```
* resourceroot/_shared/assets/images/
```

Set up the resource folder

This section describes how to set up the resource folder and configure Banner Flexible Registration to look in this folder for supported resources.

Step 1 Create resource folder

Create a folder that will contain the customized resources. This folder must be accessible to the application server that contains Banner Flexible Registration. Additionally, the account under which the application server is running must have READ access to the folder and the files in the folder.

Note

If access permissions are not granted (or are set to be denied) on the resource folder, the files will be loaded from the application server deploy directory, and no error messages will be written to the `FlexReg_Services.log` file. ■

Step 2 Create shared resource folder

Create the “_shared” folder under the resource folder. This folder will contain all of the externalized resources.

Step 3 Enable resource externalization

To enable using the resource folder, access the SFRACNFG form and enter the fully qualified path to the resource folder in the **Physical path to externalized resources** field.

Note

The path must reference the resource folder and not the _shared folder. If the path specified references a directory that does not exist, the files will be loaded from the application server deploy directory, and no error messages will be written to the `FlexReg_Services.log` file. ■

Populate the resource folder

This section describes how to populate the resource folder. There are two options you can use:

- Start with an empty resource folder, and only put images that are customized into the resource folders.
- or -
- Start with a fully populated resource folder that contains all of the resource from a particular release.

Note

Please ensure that all of the resource files in the resource directory have access permissions granted. If a file exists, but the application server cannot read the file, the resource will not load and an HTTP 404 (page not found) error will be returned. For example, if the resource is an image, a broken link will display in the Flexible Registration user interface (UI) in place of the image. Regardless of the particular type of resource, an error message will be written to the `FlexReg_Services.log` file. ■

Empty resource folder

It is easiest to start with an empty resource folder. Follow the steps in the section [“Set up the resource folder” on page 3-27](#).

As resources are customized, place the customized resources into the appropriate folder using the folder structure discussed above.

Fully populated resource folder

Using a fully populated resource folder involves initializing the folder with all available resources from a given release of Banner Flexible Registration.

Download resource bundle

A file containing all of the resources that can be relocated is available with each upgrade or patch of Banner Flexible Registration. The name of the file follows the same pattern as the EAR file: `BannerFlexibleRegistration-%version%-resources.zip`, where `%version%` is the version number.

For example, `BannerFlexibleRegistration-80600-resources.zip` contains all of the resources for Banner Flexible Registration 8.6.

References

For more information about customizing the textual content, refer to the *String Resources Customization Guide*.

For more information about customizing the colors, fonts, and so on, refer to the *Style Customization Guide*.

For more information about customizing the images such as icons, refer to the *Images Customization Guide*.



4 Product Administration

This chapter details administration information for the Banner Flexible Registration Application.

Install in an SSL-enabled environment

Banner Flexible Registration can be run in an environment that uses SSL to secure communication between your browser and the application. You do not need to make any modifications to Banner Flexible Registration to run it in an existing SSL-enabled infrastructure. It is assumed that you have already configured your infrastructure to use SSL for secure communication between users and the Banner services.

Please refer to [Chapter 3, “System Configuration”](#) for more information on how to configure your system.

Set up Banner Flexible Registration

Please complete the following steps to set up your Banner Flexible Registration system:

Step 1 Configure baseline Banner Student

Banner Flexible Registration is highly dependent upon many other areas of Banner, including Accounts Receivable, Fee Assessment, Common Matching, and more. These areas must be configured properly to support Banner Flexible Registration. Additional Banner setup details are included throughout this guide as necessary and these areas must be properly setup and configured for Banner Flexible Registration to function properly.

Step 2 Create discount sets on Flexible Registration Discount Set Validation Form (SFTVDSET)

The Flexible Registration Discount Set Validation Form (SFTVDSET) allows you to create a discount set code that will be used by Flexible Registration. Once the code is created you can create discount conditions and discount rules on the Flexible Registration Discount Conditions Form (SFDACOND) and the Flexible Registration Discount Rules Form (SFDRRULE). To create a discount set, perform the following steps:

1. Access the Flexible Registration Discount Set Validation Form (SFTVDSET).
2. Enter a discount set code.

3. Enter a discount set description.

4. Click  **Save**.

The discount set can be used with any catalog that you create on the Flexible Registration Catalog Validation Form (SFTVCTLG).

Step 3 Create discount conditions on Flexible Registration Discount Conditions Form (SFDACOND)

The Flexible Registration Discount Conditions Form (SFDACOND) allows you to create conditions that will be added to the discount set and used in the discount calculation process. You can specify an order for these conditions to ensure that the conditions are processed in the appropriate order.

Note

A section or class is only eligible for one discount of each discount type. ■

To create conditions that will be added to the discount set and used in the discount calculation process, perform the following steps:

1. Access the Flexible Registration Discount Conditions Form (SFDACOND).
2. Enter the order in which you would like this condition to be processed.
3. Enter the condition code that identifies the rule you are creating.
4. Enter the description of the discount code condition.
5. Select the type of condition for the rule, from the following options:
 - Age*Calculation based on the age of the learner.
 - Early Bird*Calculation based on the registration date.
 - Course Count*Calculation based on the learner's course count.
6. Select the operator to be used in the rule calculation, from the following options:
 - >=Processed if the selected Type is greater than or equal to the specified value.
 - =Processed if the selected Type is equal to the specified value.
 - <=Processed if the selected Type is less than or equal to the specified value.

7. Enter a number to be used in the condition processing.

8. Click  **Save**.

Step 4 Create discount rules on Flexible Registration Discount Rules Form (SFDRRULE)

The Flexible Registration Discount Rules Form (SFDRRULE) allows you to assign discount rules to a discount set using conditions that are created on the Flexible Registration Discount Conditions Form (SFDACOND). Each discount condition can be assigned an effective date range and either a discount percentage or a specific discount amount.

To create rules for coupon discounts, perform the following steps:

1. Access the Flexible Registration Discount Rules Form (SFDRRULE).
2. Select a discount set.
3. Select a filter date to display all discount rules that are active for that date.
4. Enter the condition code used in the rule you are creating.
5. Enter the start date that the rule is effective from.
6. Enter the end date that the rule is effective to.
7. Enter the discount percentage or the total amount of the discount.
8. Select the type of discount from the following options:
 - Flat*Total discount amount.
 - Percent*Discount percentage.
9. Enter the detail code that will be posted to Accounts Receivable.
10. If necessary, specify which subjects or sections qualify for the discount.
11. Click  **Save**.

Step 5 Create Banner Flexible Registration catalogs on Flexible Registration Catalog Validation Form (SFTVCTLG)

Banner Flexible Registration catalogs are collections of sections that may span multiple terms. The Flexible Registration Catalog Validation Form (SFTVCTLG) allows you to create and maintain catalogs. This form allows you to create a catalog with a unique code and description as well as specify start and end dates for when the catalog may be accessed. On this form you may associate a discount set with a catalog and if desired, specify a password that must be used to access the catalog in Flexible Registration.

To create a catalog, perform the following steps:

1. Access the Flexible Registration Catalog Validation Form (SFTVCTLG).
2. Enter a code for the catalog.
3. Enter a description of the catalog.
4. Enter a start date for the catalog.
5. Enter an end date for the catalog.

Note

If no date is entered, the End Date field will be automatically populated with '31-DEC-2099'.

6. Enter an optional discount set associated with the catalog.
7. Enter an optional password required to access the catalog.
8. Click  Save.

Step 6 Create profile fields on Flexible Registration Profile Validation Form (SFTVPROF)

The Flexible Registration Profile Validation Form (SFTVPROF) allows you to create new profile fields for the Flexible Registration application. New profile fields can be specified as either Baseline or Extended and can be either a string or a list of values.

A filter exists at the bottom of SFTVPROF to display extended fields only or both extended and baseline profile fields.

1. Access the Flexible Registration Profile Validation Form (SFTVPROF).
2. Enter a numeric value to specify the order of the display of profile items on SFRAPROF.
3. Enter a code to represent the profile data.

4. Enter the description of the profile data.
5. Select a data type for the profile data:

String Allows the user to enter profile information.

String LOV Allows the user to select profile information from a list.

6. Click  **Save**.

Step 7 Specify values for any extended profile fields on Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV)

For any extended profile fields created on SFTVPROF with a type of String LOV, Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV) allows you to specify LOV items that will display in Flexible Registration.

SFRRPLOV allows you to select which LOV options will display and specify if the user may provide a user entered value.

1. Access the Flexible Registration Extended Profile List of Values Rules Form (SFRRPLOV).
2. Select an extended profile field to customize.
3. Enter possible values that the user can select from.
4. Select the Display check box to display the value to the user.
5. Select the User-Entered Value check box to allow the user to enter a custom value.
6. Click  **Save**.

Step 8 Create Profile Rules on Flexible Registration Profile Rules Form (SFRAPROF)

Biographic and demographic information can be customized for your institution in Banner Flexible Registration. Flexible Registration Profile Rules Form (SFRAPROF) allows you to specify a display order for the profile fields in Banner Flexible Registration and specify which fields will be used in the system when a student creates a new account or updates their personal information.

The following Required Types may be specified:

- Optional - indicates that the field will display in Flexible Registration, but will not be required.
- Required - indicates that the field will be required on the profile page.
- Not Used - indicates that the field will not display on the profile page.

1. Access the Flexible Registration Profile Rules Form (SFRAPROF).
2. Select a catalog to modify how biographic and demographic data is displayed.
3. Enter a numeric value to specify the order of the display of profile items.
4. Select a required value for the profile item from the following list:

*Optional*User may optionally enter data.

*Required*User is required to enter data.

*Not Used*This field is not used by the Flexible Registration application.

 **Note**

If the Required Type is System Required, the required type may not be changed. ■

5. Select the Updatable check box to allow a returning user to update this field in the Flexible Registration application.
6. Click  **Save**.

Step 9 Create additional materials on Flexible Registration Additional Materials Validation Form (SFTVAMAT)

The Flexible Registration Additional Materials Validation Form (SFTVAMAT) allows you to manage a list of additional materials for courses in a catalog.

1. Access the Flexible Registration Additional Materials Validation Form (SFTVAMAT).
2. Enter the additional materials code.
3. Enter the description of the additional materials.
4. Click  **Save**.

Step 10 Specify additional material details on Flexible Registration Additional Materials Rules Form (SFARAMAT)

The Flexible Registration Additional Materials Rules Form (SFARAMAT) allows you to specify the default price, detail code, and additional information for optional additional materials. This form also allows you to specify which courses or sections to associate the additional material.

1. Access the Flexible Registration Additional Materials Rules Form (SFARAMAT).
2. Select an additional material code.
3. Enter a default price for the additional material.
4. Enter the default detail code for the additional material.
5. Enter additional information for the additional material.
6. Select the Courses tab.
 - 6.1. Select a subject to specify the association of the additional material to a specific course within the selected subject.
 - 6.2. Select a course to specify the association of the additional material to a specific course and subject.

OR

7. Select the Sections tab.
 - 7.1. Select a term to specify the association of the additional material to a specific section within the selected term.
 - 7.2. Select a CRN to specify the association of the additional material to a specific section of the selected term.
8. Click  Save.

Step 11 Override additional materials details on Flexible Registration Additional Materials Course Association Form (SFARCRSE)

The Flexible Registration Additional Materials Course Association Form (SFARCRSE) allows you to override the default details on additional materials for a course.

1. Access the Flexible Registration Additional Materials Course Association Form (SFARCRSE).
2. Select a subject, with additional materials, to modify.

3. Select a course, with additional materials, to modify.
4. Select the additional material to modify for this course.
5. Enter a new price for the additional material for this course.
6. Select a new detail code for the additional material for this course.
7. Enter additional information for the additional material for this course.
8. Click  **Save**.

Step 12 Override additional materials details on Flexible Registration Additional Materials Section Association Form (SFARSECT)

The Flexible Registration Additional Materials Section Association Form (SFARSECT) allows you to override the default details on additional materials for a course.

1. Access the Flexible Registration Additional Materials Section Association Form (SFARSECT).
2. Select a term, with additional materials, to modify.
3. Select a CRN, with additional materials, to modify.
4. Select the additional material to modify for this section.
5. Enter a new price for the additional material for this section.
6. Select a new detail code for the additional material for this section.
7. Enter additional information for the additional material for this section.
8. Click  **Save**.

Step 13 Create coupon rules on Flexible Registration Coupon Rules Form (SFDRUPN)

The Flexible Registration Coupon Rules Form (SFDRUPN) allows you to create rules for coupon discounts.

1. Access the Flexible Registration Coupon Rules Form (SFDRUPN).
2. Select the catalog for which the coupon will be available.
3. Select a filter date to display the active coupon codes on that date.

4. Enter a coupon code.
5. Enter the start date that the rule is effective from.
6. Enter the end date that the rule is effective to.
7. Enter the discount percentage or the total amount of the discount.
8. Select the type of discount:
*Flat*Total discount amount.
*Percent*Discount percentage.
9. Enter the detail code that will be posted to Accounts Receivable.
10. Enter the usage limit for the coupon. Leave the field empty if you want to set no limit on the coupon.
11. Click  Save.

 **Note**

You may use the Courses tab or the Sections tab to limit the coupon for use with a specific course or section. ■

Step 14 Associate purchase order IDs with third party contracts on Flexible Registration Purchase Order to Third Party Contract Form (SFBACONT)

The Flexible Registration Purchase Order to Third Party Contract Form (SFBACONT) allows you to create a purchase order ID and associate it to third party contract(s).

1. Access the Flexible Registration Purchase Order to Third Party Contract Form (SFBACONT).
2. Enter a purchase order ID to assign third party contracts. The ID can be a new ID, an existing ID, or NONE.
3. Enter the 3rd party Contract ID for the vendor responsible for the purchase order.
4. Enter the contract number for the 3rd party Contract ID for this purchase order.
5. Enter the term code associated with this purchase order.
6. Enter the registration start date when the purchase order goes into effect.

7. Enter the last registration date for which the purchase order is effective.

8. Click  Save.

Step 15 Specify configuration rules on Flexible Registration Configuration Rules Form (SFRACNFG)

System-wide rules for the Flexible Registration application are specified on the Flexible Registration Configuration Rules Form (SFRACNFG). SFRACNFG allows you to specify if GOBT PAC records should be created if a new user account is created in Flexible Registration, and if your institution is using Luminis IV. This form allows you to customize the timing of cart expiration. It also covers the Self-Service Banner URLs which allow access to Self-Service Banner from Flexible Registration, allows for specification of a directory location for externalized resources (used for customized images), and configuration of SMTP server parameters for email. It allows you to specify the Flexible Registration Deployment URL which is required to open and view attachments from the Flexible Registration File Attachment Review Form (SFFRATCH) and allows you to specify the directory location of where the attachments uploaded are stored. It covers the CAS service and logout URLs that the student is redirected to, to access single-sign on across web-based applications in your environment. It allows to specify the Cashier ID used when creating Accounts Receivable adjustment records for group based registrations.

1. Access the Flexible Registration Configuration Rules Form (SFRACNFG).

2. Specify the following system wide rules for Banner Flexible Registration:

- Create GOBT PAC Records
- Using Luminis IV
- Expired Cart Minutes
- Expired Cart Awaiting Payment Minutes
- SSB Home URL
- SSB CRN Detail URL
- Physical path to externalized resources
- Physical path for attachments
- Flexible Registration Deployment URL
- CAS server URL
- CAS logout redirect URL
- Group AR Adjustment Cashier ID
- SMTP Server Name

- Enable SSL Secured Connection
- Enable Non-Secure Fallback
- SMTP user name
- SMTP user password

3. Click  Save.

Step 16 Create email message on Flexible Registration Email Notification Rules Form (SFRAEMAL)

Email message notifications and email text can be defined on the SFRAEMAL form.

1. Access the Flexible Registration Email Notification Rules Form (SFRAEMAL).
2. Select a notification email code.
3. Enter a description for the email message.
4. Select a mime type for the email message (*plain text* or *HTML*).
5. Enter the email message text.

If the mime type is *HTML*, then HTML markup will be supported in this field to render the message in HTML.

6. Click  Save.

Step 17 Create catalog rules on Flexible Registration Catalog Rules Form (SFRAC TLG)

A catalog is a collection of sections that are grouped together, to allow the ability to segregate sections to allow a learner to view targeted sections. Catalogs are made up of sections that may span multiple terms. Alternatively, a catalog can be associated with terms, then all the sections from those terms will be part of the catalog. This catalog does not replace the Banner catalog.

The Flexible Registration Catalog Rules Form (SFRAC TLG) allows you to specify rules for how the catalog functions in Flexible Registration. You may also add section and term combinations to the catalog.

To assist administrators with the maintenance and creation of the values for a new catalog, a copy feature has been created. Once a new catalog has been created, the institution can

use the Copy feature in the Rules block to quickly copy the Rules, Searchable Attributes, E-mail and User Relationships values from an existing catalog to the new catalog.

 **Note**

The parameter values from the Schedule and Student Override tabs do not get copied. ■

1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
2. Enter a catalog code.
3. On the Rules tab, the following user-defined parameter fields can be specified:

Group	Parameter	Values	Description
DISPLAY	Search Result Limit	0 to 99999	The maximum number of classes to display in the search results. If the number of matches exceeds this value, only display this number of classes. This is used to reduce the amount of scrolling the user may have, as well as to identify that the user can refine their search criteria.
DISPLAY	Self Service Link	Y or N	Specifies whether to display the Self Service Banner (SSB) link in the header bar or not. This link should take the student to the SSB homepage. <i>Y</i> The Self Service Banner link is displayed. <i>N</i> The Self Service Banner link is not displayed.
DISPLAY	URL Redirect on Sign Out	Y or N	The URL to redirect the student to, when they sign out of the application.
DISPLAY	Home Page URL	Y or N	This is the html that is displayed in the Customizable Content area of the Home Page. The catalog specific information can be created in the html and then displayed for the users to access.

Note: The URL is case sensitive.

Group	Parameter	Values	Description
DISPLAY	Forgot Password Link	Y or N	<p>On the Sign In space of the Home Page or on the Sign In modal window, display a link to assist students with recovering their password.</p> <p><i>Y</i> Enable the Forgot Password link in Flexible Registration.</p> <p><i>N</i> Disable the Forgot Password link in Flexible Registration.</p> <p>If Enable Banner Security Questions = <i>Y</i>, the user will be asked to answer their Banner security questions to reset their PIN. If Enable Banner Security Questions = <i>N</i>, a modal window will be displayed with customizable text to direct a student on how to get their password.</p>
DISPLAY	Enable Banner Security Questions	Y or N	<p>If you are using Banner security questions, this will require a new and returning user to create or answer your Banner security questions based on the Enterprise PIN Preferences (GUAPPRF) rules of the institution. This also allows a user to reset their Banner PIN.</p> <p><i>Y</i> Enable security questions in Flexible Registration.</p> <p><i>N</i> Disable security questions in Flexible Registration.</p>
DISPLAY	Override Voice and SSB Available setting?	Y or N	<p>On the Schedule form (SSASECT) the Voice Response and Self-Service Available setting controls whether a section is displayed to users or not. No will display the class in the search results of Banner Flexible Registration regardless of the Voice Response and Self-Service Available setting.</p>

Group	Parameter	Values	Description
DISPLAY	Select default Schedule Builder display	L or W	<p>Specify the Schedule Builder view to be displayed as the default.</p> <p><i>L</i> List View</p> <p><i>W</i> Weekly View</p> <p>The institution can select the default display of the Schedule Builder page.</p>
DISPLAY	Only show registered classes currently in progress	Y or N	<p>Specify whether to show in progress classes or all classes.</p> <p><i>Y</i> Do not display classes that have occurred in the past and are no longer in progress.</p> <p><i>N</i> Display all courses.</p> <p>Note: For a traditional class, once the maximum end date from SSRMEET is in the past, the class is not shown.</p> <p>For an open learning class, once the last meeting date is in the past, the class is not shown.</p>
DISPLAY	Show Student Status Section	Y or N	<p>Specify whether or not to display the student status section on the Home page.</p> <p><i>Y</i> Display student status.</p> <p><i>N</i> Do not display student status.</p>
DISPLAY	Show Registration Time Ticket Section	Y or N	<p>Specify whether or not to display the time ticket section on the Home page.</p> <p><i>Y</i> Display time ticket section.</p> <p><i>N</i> Do not display time ticket section.</p>

Group	Parameter	Values	Description
DISPLAY	Only show classes finalized through Flexible Registration	Y or N	Specify whether or not to display only those classes that were finalized through Flexible Registration. Y Display classes. N Do not display classes.
DISPLAY	Show building and room with course location	Y or N	Specify whether or not to display the building and room details. Y Display building and room. N Do not display building and room.
DISPLAY	Show building address on Meeting times tab	Y or N	Specify whether or not to display the building address on the Meeting times tab. Y Display building address. N Do not display building address. Note: The building address can be defined in two locations in Banner: under the Building Definition block on the Building Definition (SLABLDG) form and the Site Code block on the Site Code Validation (STVSITE) form. The Site Code used on the SLABLDG form is defined on the STVSITE form. When the Show building address on Meeting times tab value is turned to Y on the SFRACTLG form, the building address is pulled from the SLABLDG form. If the address is blank there, the building address is pulled from the STVSITE form.

Group	Parameter	Values	Description
DISPLAY	Display Disclosure statement to confirm registration	Y or N	<p>During the registration process, a student is required to acknowledge the institutional registration policies, to confirm the a registration. A customizable modal window is displayed with the policies or a link to the institutional policies. The acknowledgement is required for each cart, and if enabled, the disclosure is displayed on the Flexible Registration Cart Inquiry form (SFAICART).</p> <p><i>Y</i> Display disclosure statement.</p> <p><i>N</i> Do not display disclosure statement.</p>
DISPLAY	Display Change Password link under My Profile?	Y or N	<p>Specifies whether to display the Change Password link in the My Profile drop-down.</p> <p><i>Y</i> The Change Password link is displayed.</p> <p><i>N</i> The Change Password link is not displayed.</p>
DISPLAY	Number of minutes idle before logout		<p>Refers to the time in minutes the users can remain idle before the application logs them out.</p> <p>Note: The default value for the user session timeout is 30 minutes.</p>
DISPLAY	Number of minutes before warning window		<p>Refers to the time in seconds/minutes where a warning modal window is displayed to the user before the number of minutes idle before logout occurs.</p> <p>Note: The default value is 30 seconds.</p>

Group	Parameter	Values	Description
DISPLAY	Advanced Search Fields Display Code	LOV	Advanced search field rules are created on the Flexible Registration Advanced Search Rules Form (SFRASRCH). The institution can then select a rule from that form and apply to a catalog.
DISPLAY	Always display Course Description on Course Details window?	Y or N	<p>At the bottom of the Course details modal window, a course description is displayed based on the hierarchy described in the Administrative Guide.</p> <p><i>Y</i> Display course description on the course details window.</p> <p><i>N</i> Do not display course description on the course details window.</p>
DISPLAY	PCI Payment Success URL	/ pciPaymentSuccess.jsp?	The URL the student is redirected to when a PPC payment is a success.
DISPLAY	PCI Payment Failure UR	/ pciPaymentFailure.jsp?	The URL the student is redirected to when a PPC payment fails.
DISPLAY	Show base fees with course information	Y or N	<p>Specify whether or not to display the base fees with course information.</p> <p><i>Y</i> Display the base fees.</p> <p><i>N</i> Do not display the base fees.</p>
DISPLAY	Display Required Materials on Invoice?	Y or N	<p>Specify whether or not to display the Required Materials on the Invoice page.</p> <p><i>Y</i> Display the Required Materials.</p> <p><i>N</i> Do not display the Required Materials.</p>

Group	Parameter	Values	Description
DISPLAY	Display Browse By Subject view?	Y or N	<p>Specify whether or not to display the Browse By Subject link on the Homepage and the global search bar.</p> <p><i>Y</i> Display the Browse By Subject link.</p> <p><i>N</i> Do not display the Browse By Subject link.</p>
DISPLAY	Number of days to remove class from search results	0 to 9999	<p>Refers to the number of days after which a class will no longer show up in search results after the class start date.</p> <p>Note: Leave blank for the class to appear indefinitely. Use '0' and the class will not show on the first meeting day or later.</p>
DISPLAY	Allow full class to be added to Schedule	Y or N	<p>Specify whether or not to allow a user to add classes that have no available or waitlist seats to the Schedule Builder.</p> <p><i>Y</i> Allow classes with no available or waitlist seats to be added to the Schedule Builder.</p> <p><i>N</i> Do not allow classes with no available or waitlist seats to be added to the Schedule Builder.</p>
DISPLAY	Display the class description on Invoice?	Y or N	<p>Specify whether or not to display and print the class description on the invoices.</p> <p><i>Y</i> Display and print the class description on the invoices.</p> <p><i>N</i> Do not display and print the class description on the invoices.</p> <p>Note: This is the same class description that displays in the Course details.</p>
GROUP	Group Registration Payment Detail Code	TPPY	Specify the payment detail code used for group registration payments.

Group	Parameter	Values	Description
GROUP	Group Registration Charge Detail Code	TPCH	Specify the charge detail code used for group registration payments.
GROUP	Group Registration Accounts Receivable Source Code	T	Specify the Accounts Receivable source code when posting group registration transactions.
GROUP	Allow group fee adjustment from manage registration?	Y or N	<p>Specify whether or not to allow fee adjustment for group registered carts from Manage Registration page.</p> <p><i>Y</i> Allow fee adjustment for group registered carts from Manage Registration page.</p> <p><i>N</i> Do not allow fee adjustment for group registered carts from Manage Registration page.</p>
NEWUSER	Allow new user creation	A, N, E, P	<p>An institution can choose to:</p> <ul style="list-style-type: none"> • Never allow a new account to be created • Always allow a new account to be created • Not allow a new account to be created if a single Exact match is found • Not allow a new account to be created if multiple matches or a single exact match is found ('P'). <p>Note: If a match is found in any circumstance, a user will always have the ability to login with their credentials. This value will then decide if the user has the ability to create a new account.</p>

Group	Parameter	Values	Description
NEWUSER	Level Code	LOV	<p>This value is used for two different functions:</p> <ul style="list-style-type: none"> • When a General Student account is created, the process uses this value and the term to select what Quick Admit rule to use in the General Student account creation. • The Base fees that are displayed for a class is calculated using the fees on the Schedule Detail form (SSADETL), where the level is NULL or equal to this value. <p>Note: The default value is UG.</p>
NEWUSER	Matching Source	LOV	<p>When a new account is created through Banner Flexible Registration, the Banner Common Matching process is called. This value allows the institution to select the Common Matching rules to be used in the Common Matching process.</p> <p>Note: The default value is WEBADM.</p>
NEWUSER	Create a General Student record for an inactive student	Y or N	<p>Specify if a new General Student record should be created if an existing person creates a new user account.</p> <p><i>Y</i> A new General Student record will be created.</p> <p><i>N</i> A new General Student record will not be created.</p>
NEWUSER	Display New Username and PIN	Y or N	<p>Specify whether a new learner's username and PIN should be displayed on the login window.</p> <p><i>Y</i> The learner's username and PIN will be displayed.</p> <p><i>N</i> The learner's username and PIN will not be displayed.</p>

Group	Parameter	Values	Description
NEWUSER	When saving a plan, allow new user creation	Y or N	Specify if a new Banner Flexible Registration user account can be created when a user attempts to save a plan.
			<p><i>Y</i> A new user may be created.</p> <p><i>N</i> A new user may not be created.</p>
PAYOPT	Pay Later	Y or N	Specify whether a user can pay for their registration fees at a later date.
			<p><i>Y</i> The user can pay at a later date.</p> <p><i>N</i> The user is required to pay at the time of registration.</p> <p>This value is also used in the Cart Expiration process. If this value is <i>Y</i>, and the cart has been confirmed (or in Processing Payment), when the Cart Expiration process runs, finalize the cart (and do not delete the classes). In case of Group Registration, if this value is <i>Y</i>, the expiration process calls the Bill Me payment process to make payment adjustments for group carts that get finalized through the expiration process.</p>

Group	Parameter	Values	Description
PAYOPT	Pay by Purchase Order	Y, N, A	<p>Purchase Orders take advantage of Third Party Contract rules (TSACONT), and adds a student to a Contract and then runs APPLYCRED mode to assess the contract payment rules.</p> <p><i>Y</i> Valid purchase orders are allowed. The purchase order must be created first on SFBACONT and tied to a valid Third Party Contract on TSACONT.</p> <p><i>N</i> The purchase order option on the payment page will not be displayed.</p> <p>If A (Anonymous) is used, a user may enter a purchase order number that has not yet been created on SFBACONT. Instead, SFBACONT must be created with a purchase order of DEFAULT and a corresponding TSACONT record for the NONE purchase order. All purchase orders that are not specifically created in SFBACONT will then be applied to the NONE purchase order and the corresponding TSACONT rules. The institution should later move the student from the default Third Party Contract to the correct one.</p>
PAYOPT	Pay Using Payment Processor Connection	Y or N	<p>This option uses the Payment Processor Connection and a third party payment provider to make a payment (for example, Credit card payment).</p> <p><i>Y</i> The user can pay using the Payment Processor Connection.</p> <p><i>N</i> Payment via Payment Processor Connection is disabled.</p>

Group	Parameter	Values	Description
PAYOPT	Discount Source Code	LOV	<p>If a user has discounts or coupons applied to the account, this value will be used for the source code on the TBRACCD table.</p> <p>Note: The default value is Z.</p>
PAYOPT	Discount Limit (percent)	0 to 100	<p>This maximum limit is used to regulate the maximum percent of a class that can be paid through a discount.</p>
PAYOPT	Material Source Code	LOV	<p>If a user has optional materials applied to the account, this value will be used for the source code on the TBRACCD table.</p> <p>Note: The default value is T.</p>
PROFILE	Address Type	LOV	<p>This value is used for two different functions:</p> <ul style="list-style-type: none"> • When a new account is created, use this value when creating the address. • When a user is reviewing the Personal Information, only display the most active address with this type.
PROFILE	Telephone Code	LOV	<p>This value is used for two different functions:</p> <ul style="list-style-type: none"> • When a new account is created, use this value when creating the telephone. • When a user is reviewing the Personal Information, only display the most active telephone with this type. <p>Note: The default value is MA.</p>

Group	Parameter	Values	Description
PROFILE	Email Code	LOV	<p>This value is used for two different functions:</p> <ul style="list-style-type: none"> • When a new account is created, use this value when creating the e-mail. • When a user is reviewing the Personal Information, only display the most active e-mail with this type.
PROFILE	Select the Birth Date Format	LOV	<p>This value is used for two different functions:</p> <ul style="list-style-type: none"> • When a new account is created, use this value when validating the format of the birth date entered. • When a user is reviewing their Personal Information, use this value when validating the format of the birth date entered.
REGISTER	Registration Code	LOV	<p>When a user proceeds with registering for a class(es), use this value as the course registration status code. This code must be set up on the Course Registration Status Form (SFARSTS) with an End Date greater than or equal to today. Banner Error Checking will be used to ensure that the status code is valid.</p>
REGISTER	Register Cart with Errors	Y or N	<p>This setting controls how classes with no fatal error are handled when the cart has other classes with fatal errors in it. Error level checking is set on the Registration Error Checking window of the Term Control Form (SOATERM).</p> <p><i>Y</i> Registers the user in the class(es) without a fatal error.</p> <p><i>N</i> The user must correct all fatal errors before holding a seat in any class and being able to proceed with registration.</p>

Group	Parameter	Values	Description
REGISTER	Allow Term Filter	Y or N	<p>Persistent on many pages is a search bar. This bar allows a user to enter a basic or advanced search for classes, as well as selecting to filter by term.</p> <p><i>Y</i> A student may limit their search to a selected term or search all terms.</p> <p><i>N</i> A student may not limit their search to a selected term and will search all terms.</p>
REGISTER	Invoice Only Section Fees	Y or N	<p>This value is used for two different functions. When this value is <i>Y</i>:</p> <ul style="list-style-type: none"> When a user registers for a class(es), Banner Fee Assessment is run and all fees are assessed. Section Fees from SSADETL are displayed along with charges from SFARGFE where the Assess by Course = <i>Y</i> on the SFARGFE rule. Fees without the CRN will be assessed, but not displayed. An invoice will be created under the My Profile tab. <p>The Term Control form (SOATERM) Track by CRN setting must be checked. The Registration Fee Assessment Rules form (SFARGFE) setting Assess by Course must be checked.</p> <p>When this value is <i>N</i>:</p> <ul style="list-style-type: none"> When a user registers for a class(es), Banner Fee Assessment is run and all the fees are assessed. All the fees will be displayed. An invoice will not be created under the My Profile tab.

Group	Parameter	Values	Description
REGISTER	Override automatic drop of expired carts	Y or N	As part of the Cart Expiration process, carts in a non-confirmed state can be deleted based on a set of rules defined in the Administrative Guide. <i>Y</i> overrides the Cart Expiration rules and finalizes a cart, even if it would be deleted based on the Cart Expiration rules.
REGISTER	SSRMEET dates enforced for enrollment in class	Y or N	<i>Y</i> restricts a user from adding a class that has an SSRMEET Start Date of today or earlier.
REGISTER	Enforce re-admit error checking	Y or N	Specify if Banner Flexible Registration will create a new learner record if a re-admit term error occurs. <i>Y</i> If a re-admit term error occurs, the student will receive a registration error and be prevented from registering. <i>N</i> If a re-admit term error occurs, a new learner record will be created for the student and the student will be permitted to continue with registration.
REGISTER	Use Alternate PIN?	Y or N	If an alternate PIN is created for the user and the term on the Alternate PIN Identification Number form (SPAAPIN), the user cannot complete the registration without entering this number. <i>Y</i> requires a user to enter their alternate PIN to complete the registration. <i>N</i> does not allow a user to enter an alternate PIN, complete the registration without it.

Group	Parameter	Values	Description
REGISTER	Enforce Web Drop Last Class GTVSDAX setting	Y or N	If the GTVSDAX setting for WEBDROPLST = N, the student cannot drop their last registered class over the web. N does not enforce the last drop setting and always allows the user to drop their last class regardless of the setting.
REGISTER	Override Rule from SFAROVR?	LOV	Apply an override rule from the Registration Permit-Overrides Control form (SFAROVR) to all the users registering for classes in the catalog. The override rule overrides Banner registration errors, based on the rules created for that rule code.
REGISTER	Ignore time ticketing setting in GTVSDAX?	Y or N	If the GTVSDAX setting for WEBRESTTKT = Y, the student must have a valid time ticket to register over the web. Y does not enforce the registration error check for time ticket and allows the user to register regardless of their time ticket.

- On the Schedule tab, you may associate either sections or terms with a catalog.

 **Note**

When a student completes the registration, an e-mail can be generated to the student, confirming the registration. This e-mail is tracked on the Student Mail (SUAMAIL) form. ■

- On the E-mail tab, the following fields can be specified:

Email Code	Select the e-mail code from the drop-down list. The e-mail code must first be created on the Email Notification Rules Form (SFRAEMAL).
Material Code	Select the material code from the drop-down list. This is the code that is used when creating the student mail on the SUAMAIL form.
Event Code	Event on which the e-mail is sent.
Sender Name	The name of the person that will appear in the e-mail.

From Address	The e-mail from address id that will appear in the e-mail.
Subject	The subject that will appear in the e-mail.
Enabled	Select this check box to enable sending e-mails.
From Date	The first date that this e-mail should be sent. E-mails will not be sent when today's date is prior to this date.
To Date	The last date that this e-mail should be sent. E-mails will not be sent when today's date is after to this date.

- On the User Relationships tab, you can associate Relationship codes to the catalog.

 **Note**

The relationship codes are required in order to perform group registrations. ■

- On the Student Override tab, you can specify an override value or custom function to dynamically determine the desired override value. The value(s) for the override will replace the default Quick Entry Rules from SAAQKER.

- Click  **Save**.

 **Note**

The default values will still be a combination of the term code created for the registration and the level code on the SFRAC TLG form. The default value and function will override only that single value in the General Student creation. You can enter an override value or a function for each item. ■

Step 18 Access Banner Flexible Registration

To access the Banner Flexible Registration application using a catalog you have configured, use the following URL:

<http://www.bannerellucianuniversity.edu:port/flexibleregistration/index.jsp?frc=ExampleCatalog>

Please replace the *www.bannerellucianuniversity.edu* in the example URL above with the domain that you specified in Chapter 2, replace *port* with the port number to access the application, and replace *ExampleCatalog* with the name of the catalog you created in [“Create Banner Flexible Registration catalogs on Flexible Registration Catalog Validation Form \(SFTVCTLG\)” on page 4-4.](#)

File attachments

Banner Flexible Registration provides the ability for institutions to enable students to upload file attachments during the registration process. This enables institutions to collect documents electronically and helps ensure that students are fully prepared to begin their learning activities by the time a class begins.

File Attachments that are uploaded by learners are stored on a file system, where they can be accessed and viewed, by institution administrators. Please review the [“File Attachments”](#) section in [Chapter 3, “System Configuration”](#) for more details on how to set this up.

Once the file system storage has been configured, attachments can be setup. After an attachment has been defined, it is assigned to either a Course or a Section.

When students register for a Course or Section that has attachments applied to them, a new *Upload Attachments* window will appear during registration. This will prompt the student to upload the attachments.

After the attachments have been uploaded, administrators can use the Registration File Attachment Review Form (SFFRATCH) for reviewing the attachment.

The file attachment process involves the following steps:

- [“Create attachments”](#)
- [“Review attachments”](#)
- [“Manage attachments”](#)

Create attachments

Banner Flexible Registration allows your institution to configure attachments to be associated with a course/section. To define a new attachment, perform the following steps.

Step 1 Create Attachments on Flexible Registration File Attachment Validation Form (SFTVATCH)

The Flexible Registration File Attachment Validation Form (SFTVATCH) allows you to define attachments.

1. Access the Flexible Registration File Attachment Validation Form (SFTVATCH).
2. Enter a code for the attachment.

3. Enter a description of the attachment.

 **Note**

Use of a "\" or "" is not supported in this field. ■

4. Click Save.

Step 2 Specify Attachment details on Flexible Registration File Attachment Association Form (SFFAATCH)

The Flexible Registration File Attachment Association Form (SFFAATCH) allows you to specify attachment template location and help. This form also allows you to specify which courses or sections to associate the attachment.

1. Access the Flexible Registration File Attachment Association Form (SFFAATCH).
2. Select an attachment code.
3. Enter template location for the attachment. The location value must be prefixed with the appropriate protocol (http://, ftp://, etc).

 **Note**

Use of a "\" or "" is not supported in this field. ■

4. Enter the help text for the attachment.

 **Note**

Use of a "\" or "" is not supported in this field. ■

5. Select the Courses tab.
 - 5.1. Select a subject to specify the association of the additional material to a specific course within the selected subject.
 - 5.2. Select a course to specify the association of the additional material to a specific course and subject.

OR

6. Select the Sections tab.
 - 6.1. Select a term to specify the association of the additional material to a specific section within the selected term.
 - 6.2. Select a CRN to specify the association of the additional material to a specific section of the selected term.

7. Click **Save**.

Uploading of attachments

Once attachments have been created and associated to either a course or a section, learners will now have the ability to upload files during the registration process.

During registration, if a learner has selected a class that has an attachment associated to it, once the learner reviews the registration results and selects **Continue**, they will be presented a window in which they can upload files. This window will appear in between the Registration Results page and the Payment Options page.

Note

If you have enabled the Registration Disclosure feature, the attachment window will appear prior. ■

Learners will be provided a window that gives them a few of the Attachment information you setup in the previous steps. They will also have a **Browse** button in which, when they click it, they can select the file from their computer to upload.

Once the file has been selected, it is uploaded to the Flexible Registration application. The file is now stored on the institution's server, based on the settings provided.

When the student has finished uploading the file, they can click the **Continue** button to proceed through the registration process.

Review attachments

Banner Flexible Registration allows your institution to review the attachments uploaded by students. To review an attachment, perform the following steps:

1. Access the Flexible Registration File Attachment Review Form (SFFRATCH).
2. Perform a search on a specific attachment type, term, course, student ID, student name and/or upload date.
3. Click the **Open Attachment** button in the attachment row you are interested.
4. Click the **Open File Attachment** link in the newly opened browser window. This action will open/save the attachment.
5. View the attachment.

Please note that files uploaded from learners pose a risk to your infrastructure and your computer if you download them. As disclosed on the **Open File Attachment** page, you should be aware of the risk of opening a file uploaded from someone else. Please ensure you have the current Anti-Virus software installed on your computer prior to opening file attachments.

If you would like to mark the attachment as being viewed and validated, Banner Flexible Registration provides a very basic mechanism by which you can mark the attachment as being *verified*. Perform the following steps to mark the attachment that was just viewed, as being verified.

1. Access the Flexible Registration File Attachment Review Form (SFFRATCH).
2. If the form is still visible from the steps above, simply enable the window.
3. Ensure that the proper row is highlighted.
4. Select the **Verified** check box if the attachment is acceptable.
5. Click **Save**.

Manage attachments

As discussed in [Chapter 3, “System Configuration”](#) section, Anti-Virus software is recommended for use and should be run on the directories file attachments are stored. This will ensure that learners are not uploading virus infections to your infrastructure.

Over time, attachments may need to be deleted. The Network or Server Administrators can clean up the storage location by moving or deleting files. When you delete a file from the file system, it should be noted that there is now a disconnect between the file system and the record in the Registration File Attachment Review Form (SFFRATCH). If necessary and desired, you can delete the record in the form to keep the file system and the database in sync. Please note, that the SFFRATCH form stores the location of the file.

New user creation

When a student creates a new user account in the Banner Flexible Registration application, standard Banner Common Matching is used to see if a student already exists in the Banner database.

Rules on new user creation are maintained on the Flexible Registration Catalog Rules Form (SFRAC TLG). You may specify the following properties involving new user creation:

Allow new user creation	<p>Specify if new users are allowed to be created.</p> <p><i>A</i> Always - Permit the student to create a new user record.</p> <p><i>N</i> Never - Students will not be able to create a new account in Banner Flexible Registration when using this catalog.</p> <p><i>E</i> Exact - If an exact person match is found by Common Matching, prevent a new account from being created.</p> <p><i>P</i> Potential - If a potential or exact person match is found by Common Matching, prevent a new account from being created.</p>
Level Code	<p>Specify the level code used to create the General Student record.</p> <p>Note: The Quick Admit rules for this code and the term are used.</p>
Matching Source	<p>Specify the Common Matching rules to execute when creating a new student record from the Common Matching Source Code Validation Form (GTVCMS).</p>
Create a General Student record for an existing person	<p>Specify if a new General Student record should be created if a person has an account but does not have an active student record.</p> <p><i>Y</i> A new General Student record will be created.</p> <p><i>N</i> A new General Student record will not be created.</p>
Display New Username and PIN	<p>Specify whether a new learner's username and PIN should be displayed in the login window.</p> <p><i>Y</i> The learner's username and PIN will be displayed.</p> <p><i>N</i> The learner's username and PIN will not be displayed.</p> <p>Note: The alternative to displaying the username and PIN in the login window is to send a letter or an e-mail containing the information.</p>
When saving a plan, allow new user creation	<p>Specify if a new Banner Flexible Registration user account can be created when a user attempts to save a plan.</p> <p><i>Y</i> A new user can be created when saving a plan.</p> <p><i>N</i> A new user can not be created when saving a plan.</p>

General student record

When the Create a General Student record for an existing person rule is enabled on the SFRACTLG form, student data entered in the Quick Entry Rules Form (SAAQKER), for term and level combination, will be used when populating the student record. The level used is defined on the Flexible Registration Catalog Rules Form (SFRACTLG). The “From” term used to select the quick rule will be the earliest term for the sections in their cart. The “To” term will be defaulted to '999999'.

For returning learners, the following rules will apply:

- If the learner has a current student record that is active, as defined on the Student Status Code Validation Form (STVSTST), the learner record status will be updated to active for the record.
- If the learner has a person record but not a student record, a student record will be created using the Quick Entry rules.
- If the learner has a student record that is in the future, but no current student record, a new student record will be created using the Quick Entry rules.
- If the learner has a current student record that is 'inactive', as defined on the Student Status Code Validation Form (STVSTST), in a term other than the term for which the student is attempting to register, a new student record will be created using the Quick Entry rules.
- If the learner has a current student record that is 'inactive', as defined on the Student Status Code Validation Form (STVSTST) beginning in the term for which the learner is attempting to register, the status code of that record will be updated to the status code created on the Quick Entry rule form. No changes will occur to that record other than the status change.

Overriding Quick Entry Rules

To give the institution greater flexibility when creating the General Student record, institutions can also choose to override the default values from SAAQKER. A hardcoded override value or a custom function can be written to dynamically override the default values. If the override value and the custom function are left blank for a given item, the default value from the Quick Admit process will be used during the General Student creation.

Each item can have either an override value or a custom function.

Custom Override Function API

Flexible Registration provides an API to customize the Student quick admit rule. This API provides a framework that can be used to write this required database function that returns

a dynamic desired override value based on the input parameters. The following are the details of the API:

- A function needs to be created under FLEXREG schema.
- The function takes in five parameters: PIDM, term, catalog code, cart ID and banner default value for the given quick admit rule.
- The function returns the desired override value for the given quick admit rule.

The following is an example for a code snippet of a custom function.

```
CREATE OR REPLACE FUNCTION F_CUSTOM_QKER_RULE (
  p_pidm          NUMBER,
  p_term_code     VARCHAR2,
  p_catalog_code  VARCHAR2,
  p_cid           VARCHAR2,
  p_banner_default VARCHAR2
)
RETURN VARCHAR2
IS
  lv_cust_value VARCHAR2(5);
BEGIN
  --Custom logic to calculate lv_cust_value
  RETURN lv_cust_value;
END F_CUSTOM_QKER_RULE;
```

Credential Notification

When a new user account is created, the student must be notified of their USERNAME and PIN for future login. You have the ability to notify a student in a combination of 3 ways:

- Email
- Letter
- Popup Window

No matter how you elect to notify the student, a message will be displayed to the student when their profile is created. This message is customizable and can inform the student of the notification method.

Regardless of the setting of the Display New Username and PIN rule on the (SFRACTLG) form, the creation of a letter or email will follow the rules setup on the Enterprise PIN Preferences Form (GUAPPRF).

Following Baseline Banner functionality, when a student completes the profile creation process, the rules setup on the Enterprise PIN Preferences Form (GUAPPRF) under the

PIN Notification Preferences heading will be used. The Institution has the ability to create a letter, e-mail or both on this form.

 **Note**

The rules that will be followed are listed as Admin/Process changes PIN on the form. Hard-copy Letter Generation Code and E-mail Letter Generation Code are the letter codes that will be created on the Student Mail Form (SUAMAIL). ■

The rules that are used are based on the setting for When Administrator/Process changes PIN on the Enterprise PIN Preferences Form (GUAPPRF). This can be set not to display any message, send an e-mail, send a letter, or send both.

 **Note**

If the Create GOBTPAC Record is not selected on the Configuration Rules Form (SFRACNFG), no GOBTPAC record will be created, and the password on the Welcome pop-up window will display NA because there is no GOBTPAC_PIN to display. ■

Banner Flexible Registration will only display the Banner ID (or GOATPAC 3rd party ID) and PIN in the Welcome pop-up window during account creation. The following can be displayed:

- If using the Banner ID and PIN (not LDAP functionality), SPRIDEN ID and GOATPAC PIN can be displayed.
- If using Luminis/LDAP, the GOBTPAC 3rd party ID and PIN can be displayed.
- Banner Flexible Registration will not display the LDAP username or password.
- If using CAS, no username or password will be displayed.

 **Note**

Banner Flexible Registration will not create an LDAP account (including Luminis or CAS). It is the responsibility of the institution to create the LDAP account, as well as notify the students of their login credentials. If the students are using LDAP, they will not be able to log back into Banner Flexible Registration after logging out, without receiving their LDAP credentials.

When using Luminis, it is important for the WebTailor Protocol to be set to LDAP. This is used by Self Service Banner and Banner Flexible Registration. When the WebTailor Protocol is set to LDAP, the GOATPAD Third Party ID (GOBTPAC_EXTERNAL_USER) will be displayed as the user name, and the GOATPAD PIN (GOBTPAC_PIN) will be displayed as the password. ■

Acceptable usage indicator

Banner Flexible Registration uses the Captcha component to create a challenge-response test in the application to ensure that new user profiles are being created by a human being

rather than by an automated process. The Captcha component accomplishes this by generating a JPEG image which contains a random series of alpha-numeric characters which are visually distorted but still recognizable to humans. The image is displayed during the new user account creation process and requires the user to enter the same characters which are present in the image to submit the request.

The acceptable use indicator can be enabled in the Banner Flexible Registration application by setting the Display usage Page to Y on the Web Rules page in Web Tailor. If the Agree with Acceptable Use Policy indicator is displayed, a new user must accept the usage policy before their account will be created.

When the student accepts the policy, the usage indicator on GOATPAD will be selected (set to Y).

For an existing student, who has not yet accepted the policy, the student will be prompted to read and accept the policy when signing into Banner Flexible Registration for the first time. The student will not be able to sign in until agreeing to the acceptable usage policy.

 **Note**

If the Display Usage Indicator in Web Tailor is set to No, the student will not be prompted to agree with the policy when creating a new account or when signing into an existing account. ■

Relationships

Banner Flexible Registration allows each catalog to enable relationships between people. Currently, this feature is only used to enable the support of group registrations. Group registrations enable a person to create an account and sign up others for classes. Examples could include, a parent signing up children for summer camp classes or a business office manager signing up a group of employees for a job training class. For more information on how group registrations are performed, see [“Registration” on page 4-40](#).

A relationship defined in the catalog enables support for group registrations in that catalog. When a catalog has a relationship assigned to it (that allows for registration), the user will see a choice on the schedule builder, as to whether they want to register just themselves or if they want to register themselves and other people. If no relationships are enabled, they will not see that choice and registration will continue for just the logged in user.

A relationship should describe the type of relation between the parties. Examples include:

- Guardian
- Employee
- Co-Worker

There are two options that can be used to describe behavior associated to the relationship.

The **Display Credentials** option enables you to determine if the new login credentials are to be displayed when new people are created under a relationship. If this option is enabled, when a person creates a new account to register for a class for another person, the login credentials will be displayed. This will be helpful for guardians with minor children. The only way for the guardian to manage that minor child's current registration(s) is to login as that minor child.

The **Allow Registration** option enables registration for the relationship. This option should be enabled for each relationship type. If the setting is not checked, the group registration will not be enabled for the catalog. This means the selection to register just yourself or others and the ability for the user to see their relationships will not be available. Only relationship types with this option enabled will be available for others to select during the create relationship process.

 **Note**

It is important to note that a person will only have the ability to register another person. They will not have the ability to see any of the other person's classes, nor will they have the ability to drop the other person from any class once the checkout process is complete. ■

Setup

To create a relationship, perform the following steps:

1. Access the Flexible Registration User Relationship Validation Form (SFTVRLTN).
2. Enter a Relationship Code and description.
3. Select the **Display Credentials** check box if you wish to display the login credentials of the person being created to the person who is creating the student.
4. Select **Allow Registration** check box if the primary person in the relationship can register for the other person in the relationship.
5. Click  **Save**.

To associate the relationship to a catalog, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRAC TLG).
2. Select a catalog.
3. Select the User Relationships tab.

4. Select the desired relationship code to add to the catalog.

5. Click  **Save**.

Creating and Viewing

Once a relationship has been created and associated to a catalog, users can begin to build and utilize them for registration.

There are two ways a relationship can be created: a user can create a new student they intend to register or an institution administrator can associate existing students to a relationship.

When relationships that allow registrations are enabled, a user signed into Flexible Registration can create new users that they will be responsible for. This can be done in one of the two following ways.

The first is using the Relationships link. This page shows the current user who they have a relationship with. From this view, the user can create a new user by creating a new profile for them. Once the profile has been created, this new user will show up in the signed in users list. A new relationship has been created.

The second approach is creating a new user during the checkout process. Once the signed in user has added classes to their Schedule Builder and selected to register **yourself and/or others**, a new page will be shown that requires the signed in user to associate students they can register to classes in the shopping cart. If a user does not appear in the list, the signed in user can create a new user from this page.

An institution administrator can also create a relationship between two existing users.

To create a relationship between two users, perform the following steps:

1. Access the Flexible Registration User Relationship Form (SFPARLTN).
2. Select the ID for the user.
3. Enter a relationship code to use for the relationship.
4. Select the ID for the second user in the relationship.
5. Click  **Save**.

Once a relationship has been established, the link remains unless it is removed by the institution.

To disable a relationship between two users, perform the following steps:

1. Access the Flexible Registration User Relationship Form (SFPARLTN).
2. Select the ID for the user.
3. Select the **Disable** check box on the relationship you wish to disable.
4. Click  **Save**.

The relationship between the two users will not be inactive and a group registration will not be allowed between the two users.

A relationship can only be added to an existing student by an administrator through the Flexible Registration User Relationship Form (SFPARLTN).

Registration

After a student has selected classes they want to register for, they begin the checkout process within Flexible Registration. The checkout process and its step by step flow direct the student through a variety of options and choices as they walk through the registration process in the application.

Depending on the settings in the catalog or options on a class, the following steps can be presented to the student:

1. Signing up other students
2. Assigning multiple students to classes
3. Selecting class options
4. Purchase of Additional Materials
5. Review and adjustment of registration results
6. Application of coupons
7. Uploading file attachments for classes
8. Acceptance of a registration disclosure
9. Selecting a payment option
10. Making a payment

11. Reviewing the final invoice

12. Receiving e-mail acknowledgment

All of these steps encompass a complete registration flow and a shopping cart that is in its finalized or completed state.

There are two different perspectives that are supported for doing registration. You need to consider that when using each of the features of the checkout flow.

The first perspective is a user who is simply signing themselves up for a class. This user searches for classes, adds the classes to their schedule builder, checks out, makes a payment, and actually attends the class. All of the features for options, coupons, disclosure, etc. are all about the user.

The second perspective is that of a user who is not only signing themselves up for a class, but they could also be signing up others. In this situation, the person who is logged into Flexible Registration can register others for a class. This is an important distinction when understanding the various steps in the checkout flow.

In the following sections, more details are provided when considerations have to be made when the registration is one of these perspectives.

Fee assessment

During the checkout flow, once a student has reviewed and confirmed their interest in signing up for a class, Flexible Registration calls baseline Banner Fee Assessment which determines and appropriately creates charges for the student on their account at the institution.

There are two circumstances that Flexible Registration augments the baseline fee assessment process.

The first is when coupons, discounts, or material are used or purchased during the checkout. In these cases, Flexible Registration manually calculates the charges or discounts and creates entries on the users account for the correct amounts.

The second situation is when a user is not only registering themselves, but they are registering others. In order to correctly account and enforce a single payment transaction at checkout, Flexible Registration adjusts fees on each users account in the shopping cart.

During the checkout flow, before payment has been received, baseline fee assessment is run on each individual in the cart (just as if they are registering themselves). But, once the payment has been received and the shopping cart is being completed, Flexible Registration adjusts each individual account accordingly. For each registrant, a payment row is created to balance their account. For the user who is paying, a charge is created. Thus, when the payment is processed, the users who paid account will be in balance.

When Flexible Registration makes an adjustment to an account, it uses the detail codes defined in the SFRACTLG form for the current catalog. The following configurable values are used:

- Group Registration Payment Detail Code
- Group Registration Charge Detail Code

If a student drops a class through the Banner Flexible Registration Manage Registrations page, Banner Fee Assessment is called, and the reversal of the tuition and fees are processed. Banner Flexible Registration identifies if adjustments are needed to credit the original payer. If adjustments are needed, the original adjustment payment detail code and original adjustment payment detail code are used to adjust the transactions as a result of Fee Assessment. This will ensure that the student's account is appropriately adjusted, and that the original payer's account is properly credited for the drop. The **Allow group fee adjustment from manage registration?** on the Catalog Rules Form (SFRACTLG) determines if this adjustment occurs.

It should be noted that if the Invoice by Section fees is set to *N* in the Catalog Rules Form (SFRACTLG), this after payment adjustment is not performed. If the setting is *N*, then the process cannot properly determine how to adjust fees under certain complex conditions. Therefore, the student account will be updated as if they were the person that paid for the class.

To ensure adjustments are made on classes where fees changed outside of Flexible Registration, the Fee Adjustment Process (SFKRGADJ) can be used.

Considerations

- If the Invoice By Sections fees setting is *N* after payment adjustments on the shopping cart are not performed.

For example, a group registration is performed and a student is put on a waitlist. The Invoice By Section fees setting is *N*, when the student is waitlisted. When the student comes to the Manage Registration page from the waitlist, the student will have the charges posted to their account. The adjustment will not move to the original payer for the cart.

- If a complex situation results in multiple shopping carts for a student who registers for a class, the after payment adjusts are not performed. This will be under conditions such as a student registers, then drops, has someone register them and then drops, again and has a third party register them for a class.
- If a Purchase Order is used for payment, Flexible Registration fee adjustment is never done.

Fee Adjustment Process

The Fee Adjustment Process (SFKRGADJ) is included in Flexible Registration. This process is a Banner Job Submission process that is used to execute after payment

adjustments. This process reviews a given shopping cart to make sure all the accounts are in balance. If a student's account is not in balance, the fees will be adjusted to the payer of the shopping cart. This process helps ensure that any change in class fees after registration or if a student drops a class administratively (through calling the institution) or through Banner Self Service, the charges will properly be accounted for.

The Fee Adjustment Process can be run at various levels of detail. For example, it can be run for a given term, for a student, or for all users in a shopping cart. Depending on the situation and desired outcome, an institution administrator can execute the process as needed.

To run the Fee Adjustment Process (SFKRGADJ), perform the following steps:

1. Access the Fee Adjustment Process (SFKRGADJ).
2. Enter the criteria you wish use to scope the process.
3. Enter a term code.
4. Submit the process as directed on the Banner General Process Submission Control Form (GJAPCTL).

To view any adjustments that have taken place, you can use the Adjustments Inquiry Form (SFFIADAR). This form shows detailed information about the adjustment. Various details about the adjustment can be found such as the AR Tran Number, the Shopping Cart, Cross Reference and IDs.

Discounts and coupons

Banner Flexible Registration allows you to create discount sets and coupon codes that can be used by students when registering for classes.

Each class in the shopping cart can receive a maximum of one discount of each type (age based, date based and course count), as well as one coupon, for a total of four discounts.

Coupon processing is supported in both individual and group registrations. In the case of a user signing up others for a class, coupons are applied on a per registrant basis since the charges for group registration are made on a per registrant basis. This means that a coupon in essence will be applied multiple times in the shopping cart – once for each person. If a coupon has a usage limit applied to it and it is a group registration, then the coupon will only be applied to people in the cart while the limit is still valid. Therefore, there could be a situation, for example, if the coupon limit is set to three and the shopping cart has 30 students in it, the coupon will only be applied to three students.

Create discount sets

Discount sets define the conditions and rules under which a discount can be applied during registration.

1. To create a discount set, perform the following steps:
 - 1.1. Access the Flexible Registration Discount Set Validation Form (SFTVDSET).
 - 1.2. Enter a discount set code and description.
 - 1.3. Click  **Save**.
2. To create conditions that will be added to the discount set and used in the discount calculation process, perform the following steps:
 - 2.1. Access the Flexible Registration Discount Conditions Form (SFDACOND).
 - 2.2. Enter the condition code that identifies the rule you are creating.
 - 2.3. Enter the description of the discount code condition.
 - 2.4. Select the type of condition for the rule, from the following options:
 - AgeCalculation* based on the age of the learner.
 - Early BirdCalculation* based on the registration date.
 - Course CountCalculation* based on the learner's course count.
 - 2.5. Select the operator to be used in the rule calculation, from the following options:
 - >=Processed if the selected Type is greater than or equal to the specified value.
 - =Processed if the selected Type is equal to the specified value.
 - <=Processed if the selected Type is less than or equal to the specified value.
 - 2.6. Enter a number to be used in the condition processing.
 - 2.7. Click  **Save**.
3. To create rules for coupon discounts, perform the following steps:
 - 3.1. Access the Flexible Registration Discount Rules Form (SFDRRULE).
 - 3.2. Select a discount set.
 - 3.3. Select a filter date to display all discount rules that are active for that date.

- 3.4. Enter the condition code used in the rule you are creating.
 - 3.5. Enter the start date that the rule is effective from.
 - 3.6. Enter the end date that the rule is effective to.
 - 3.7. Enter the discount percentage or the total amount of the discount.
 - 3.8. Select the type of discount from the following options:
 - Flat*Total discount amount.
 - Percent*Discount percentage.
 - 3.9. Enter the detail code that will be posted to Accounts Receivable.
 - 3.10. If necessary, specify which subjects or sections qualify for the discount.
 - 3.11. Click  **Save**.
4. To specify the maximum discount limit for each subject or section, perform the following steps:
 - 4.1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
 - 4.2. Select a catalog.
 - 4.3. Specify the source code used for discounts added to Accounts Receivable.
 - 4.4. Specify the maximum discount limit for each section.
 - 4.5. Click  **Save**.

Create coupon codes

To create rules for coupon discounts, perform the following steps:

1. Access the Flexible Registration Coupon Rules Form (SFDRUCUPN).
2. Select the catalog for which the coupon will be available.
3. Select a filter date to display the active coupon codes on that date.
4. Enter a coupon code.
5. Enter the start date that the rule is effective from.
6. Enter the end date that the rule is effective to.

7. Enter the discount percentage or the total amount of the discount.
8. Select the type of discount from the following options:
 - Flat*Total discount amount.
 - Percent*Discount percentage.
9. Enter the detail code that will be posted to Accounts Receivable.
10. Enter the usage limit for the coupon. Leave empty if you want to set no limit on the coupon.
11. If necessary, specify which subjects or sections will accept the coupon.
12. Click  **Save**.

Invoicing

Banner Flexible Registration allows you to display one of two different invoice types:

- an invoice that only displays the section charges for the classes in the student's cart
- an invoice that displays the total current fee assessment charges for the student

Section charges invoice

To only display the section charges for the classes in the student's cart on the invoice, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
2. Select a catalog.
3. Set the **Invoice Only Section Fees** parameter to *Y*.

The **Invoice Only Section Fees** parameter allows an institution to select what charges are displayed to the student in the invoice and determine if an old invoice will be made available to a returning student.

- When this parameter is set to *Y*, *only* section fees are displayed. (These are the TBRACCD fees where the TBRACCD_CRN field is populated with the CRN and term for the newly registered class or classes). This requires that the **Track by**

CRN indicator on SOATERM is checked (Y) for SSADETL charges and the **Assess by Course** indicator on SFARGFE is checked (Y) for SFARGFE fees.

Previously generated invoices created through Banner Flexible Registration will be made available to returning students under the My Invoices option on the My Profile menu.

- When this parameter is set to *N*, section fees, as well as other TBRACCD transactions, are displayed on the invoice.

Previously generated invoices created through Banner Flexible Registration will *not* be made available to returning students under the My Invoices option on the My Profile menu.

4. Click  **Save**.
5. Access the Term Control Form (SOATERM) and enable the **Track by CRN** option.

 **Warning**

Be aware of the effects of setting the **Track by CRN** check box to checked (Y) for a term for which fee assessment has been previously run. When you go into production with Flexible Registration in a term that is in progress, changing the setting of **Track by CRN** from unchecked (N) to checked (Y) will cause a double assessment of previously assessed non-Flexible Registration section fees from SSASETL. This occurs because SFKFEE1 needs to insert the TBRACCD_CRN value. Since it cannot insert the value on previously assessed records that have been saved, it creates a new assessment. See FAQ 1-1PAJV5 for more information. ■

6. Click  **Save**.
7. If you are using the Registration Fee Assessment Rules Form (SFARGFE), enable the **Assess by Course** indicator.

 **Note**

If a general fee is used that is not tied to a class, that fee will be posted to Accounts Receivable, but will not display in this invoice type. ■

Total fee assessment invoice

To display the total current fee assessment charges for the student on the invoice, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRACCTLG).
2. Select a catalog.

3. Set the **Invoice Only Section Fees** parameter to *N*.

4. Click  **Save**.

All of the charges as a result of the current registration will be displayed on the Registration Results page, and not necessarily just the charges for the classes in the current cart. On the Final Invoice, all of the student's transactions will be grouped and displayed so the student can see the total charges for their account.

In case of Group Registration, when the payment is made, the Final Invoice page also displays the total amount paid for the current cart.

Additional materials

Additional materials that are available for a class can be specified in Banner Flexible Registration to allow a student to purchase additional materials for classes during the registration process.

1. To create additional materials, perform the following steps:
 - 1.1. Access the Flexible Registration Additional Materials Validation Form (SFTVAMAT).
 - 1.2. Enter the additional materials code.
 - 1.3. Enter the description of the additional materials.
 - 1.4. Click  **Save**.
2. To specify the default price, detail code, and additional information for optional additional materials, perform the following steps:
 - 2.1. Access the Flexible Registration Additional Materials Rules Form (SFARAMAT).
 - 2.2. Select an additional material code.
 - 2.3. Enter a default price for the additional material.
 - 2.4. Enter the default detail code for the additional material.
 - 2.5. Enter additional information for the additional material.

2.6. Select an optional subject to limit the requirement of the additional material to a specific subject or select an optional course to limit the requirement of the additional material to a specific course.

2.7. Click  **Save.**

3. To specify the maximum discount limit for each subject or section, perform the following steps:

3.1. Access the Flexible Registration Catalog Rules Form (SFRAC TLG).

3.2. Specify the Accounts Receivable source code for additional materials for a course in the catalog on the Material Source Code field.

3.3. Click  **Save.**

Override additional material settings for a course

To override the default details on additional materials for a specific subject or course, perform the following steps:

1. Access the Flexible Registration Additional Materials Course Association Form (SFARCRSE).

2. Select a subject or course, with additional materials, to modify.

3. Select the additional material to modify for this course.

4. Enter a new price for the additional material for this course.

5. Select a new detail code for the additional material for this course.

6. Enter additional information for the additional material for this course.

7. Click  **Save.**

Overriding additional material settings for a section

To override the default details on additional materials for a specific section, perform the following steps:

1. Access the Flexible Registration Additional Materials Section Association Form (SFARSECT).

2. Select a term and CRN, with additional materials, to modify.

3. Select the additional material to modify for this section.
4. Enter a new price for the additional material for this section.
5. Select a new detail code for the additional material for this section.
6. Enter additional information for the additional material for this section.
7. Click  **Save**.

Additional materials order fulfillment

When a student elects to purchase additional materials during the registration process, your institution will need to access the Flexible Registration Additional Materials Inquiry Form (SFAIAMAT) to query and display information on the additional materials purchased for a course through Banner Flexible Registration. This form allows you to perform a search on a specific term, course, student ID, invoice number, detail code, and/or additional material code. In addition, the additional materials information is also stored in a Banner table that will allow your institution to perform report and data mining functions to track the selection of addition materials.

Note

Banner Flexible Registration does not perform any order fulfillment functions. ■

Idle cart expiration

The amount of time that a student may keep classes in their cart before registering for those classes can be specified on the Flexible Registration Configuration Rules Form (SFRACNFG). To specify how long a session can be inactive before it is marked as idle:

1. Access the Flexible Registration Configuration Rules Form (SFRACNFG).
2. Specify how many minutes a session can be inactive before it is marked as idle in the Expired Cart Minutes field.
3. Click  **Save**.

Payment options

Banner Flexible Registration allows you to enable the following payment options:

- Purchase order

- Bill me later
- Credit card using Banner Payment Processor Connection

Payment options can be enabled or disabled for a catalog on the Flexible Registration Catalog Rules Form (SFRACTLG).

Please refer to the following payment options matrix to determine what will happen if various combinations of payment methods are enabled:

Purchase Order	Pay Later	Payment Processor Connection	When the user checks out...
N	N	N	No payment options are available. The student is immediately registered for classes and can be billed at a later date.
N	Y	N	The student is immediately registered for classes and can be billed at a later date.
Y	N	N	The student is required to pay with a valid purchase order to complete registration.
A	N	N	The student is required to pay with a valid or anonymous purchase order to complete registration.
Y	Y	N	The student may choose to pay with a valid purchase order or pay later to complete registration.
A	Y	N	The student may choose to pay with a valid or anonymous purchase order, or pay later to complete registration.
N	N	Y	The student is required to pay with a credit card using the Payment Processor Connection to complete registration.
N	Y	Y	The student may choose to pay with a credit card using the Payment Processor Connection or pay later to complete registration.
Y	N	Y	The student may choose to pay with a credit card using the Payment Processor Connection or a valid purchase order to complete registration.

Purchase Order	Pay Later	Payment Processor Connection	When the user checks out...
A	N	Y	The student may choose to pay with a credit card using the Payment Processor Connection or a valid or anonymous purchase order to complete registration.
Y	Y	Y	The student may choose to pay with a credit card using the Payment Processor Connection or a valid purchase order. The student may also elect to pay later.
A	Y	Y	The student may choose to pay with a credit card using the Payment Processor Connection or a valid or anonymous purchase order. The student may also elect to pay later.

External payment

The external payment option allows a student to register and pay for classes using an external payment provider specified by your institution. An invoice will be generated for the student after the student has registered and paid.

Note

In Banner Flexible Registration 8.6 release, this functionality has been removed. ■

Purchase order

The purchase order option allows a student to register and pay for classes using a purchase order. An invoice will be generated for the student after the student has registered and paid. To enable the purchase order option, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRAC TLG).
 - 1.1. Set the Pay by Purchase Order parameter to Y to allow the student to pay using a valid purchase order.
 - 1.2. Click  Save.

Bill me later

The Bill me later payment option allows a student to successfully register for classes without submitting any payment information. Fees will be assessed after the student has registered, and the student can be billed at a later date. To enable the Bill me later payment option, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
2. Set the Pay Later parameter to Y to allow the student to pay at a later date.
3. Click  **Save**.

Set up credit card types

To set up credit card types, perform the following steps:

1. Access the Credit Card Merchant ID Form (GOAMERC).
2. Enter *FLEXREGCCREGFEES* in the **Process** field in the Key block.
3. Enter a value in the **Credit Card** field.
4. Enter a value in the **System** field.
5. Use Next Block to access the Credit Card Merchant ID block.
6. Define the credit card types for the rule.
7. Click  **Save**.

Credit card using the Payment Processor Connection

The Payment Processor Connection connects Banner Flexible Registration with third-party vendors that process payment card transactions.

The Payment Process Connection is installed and supported through Banner. Flexible Registration simply utilizes the existing capabilities and services.

Payment amounts are entered or acknowledged in Banner. The Payment Processor Connection uses URL redirects and Web services to connect Banner to the vendor applications where payment card information is entered, processed, and confirmed.

For more information on the Payment Processor Connection, please refer to *Banner General Payment Processor Connection Handbook*.

To enable the credit card payment option, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
2. Set the Pay Using Payment Processor Connection parameter to Y to allow the student to pay with a credit card.
3. Click  **Save**.
4. Configure the Payment Processor Connection URL by following the steps in [“Payment Processor Connection URL” on page 4-54](#).

Payment Processor Connection URL

As mentioned in the Payment Processor Connection Handbook, there are two URLs that can be configured to support the end of the payment process: the URL to redirect a student to when payment succeeds or if payment fails.

Note

Normally, these parameters should not change when Flexible Registration is deployed. The only time these parameters should change is potentially in a load balanced environment. ■

If the parameters must be changed, the URL for the Payment Processor Connection is specified in the Flexible Registration Catalog Rules Form (SFRACTLG).

The parameter values can be found in the `DISPLAY` group and are named `PCI Payment Success URL` and `PCI Payment Failure URL`.

If the URL values begin with a `."` or `"/`, the URL is considered to be a relative URL and Banner Flexible Registration will build the complete URL for processing at runtime.

If the `successURL` or `failureURL` values are modified and the new values do not begin with a `."` or `"/`, the URL is considered to be an absolute URL and Banner Flexible Registration will use those exact URL values.

Note

The success and failure URLs must end with a `“?”` so that when the final redirect occurs, the appropriate URL request parameters can be appended at runtime to the URL as needed. ■

Make a payment using Payment Processor Connection

When a user attempts to make a payment, a new browser window will open and the Payment Processor Connection site will display. The student must enter all of the required information to make a payment.

At this point, Flexible Registration will poll the Banner database to determine if payment has been made. When the payment provider accepts the payment, it will send a payment received notification to Banner. Flexible Registration will continually poll the Banner database until it receives a response that this payment has been received by Banner.

Upon confirmation of the payment, Flexible Registration will take the user to the Invoice page where the user can view their invoice for the courses they have just paid for.

Cancel the payment using Payment Processor Connection

When a user attempts to cancel the payment, there are two possible outcomes depending on the following criteria:

- If the user clicks **Cancel Payment** on the Awaiting payment window followed by **Yes** on the Cancel Payment confirmation dialog box, the payment details screen will remain displayed. The **Submit payment** button will remain enabled so the user can continue to pay for their courses in the event the user cancelled their payment by mistake.
- If the user clicks **No** on the Cancel payment dialog box, the Awaiting payment window will remain open.
- If the user clicks **Yes** to confirm cancellation, Flexible Registration will stop polling the Banner database for payment confirmation, and the payment details screen will remain displayed. The **Submit payment** button will remain enabled in the event the user cancelled their payment by mistake.

Note

If the user closes the payment provider browser window prior to completing the payment process but does not click **Cancel**, the user will be registered for the course but registration will not be finalized because no payment was received. In this instance, the next time the user logs in to the Flexible Registration application, an Outstanding Payment message will popup and the user will be given the option to pay for their courses or cancel. ■

- If the user does not login again, the cart expiration process will run and determine how to handle the users shopping cart. For more information, see [“Cart expiration process” on page 3-3.](#)

Additionally, a database job can be configured to run on a regular basis that will clean up any shopping carts that have not completed payment. If this job has run, the user will not receive notification that their cart has been emptied by the process.

Home page configuration

Banner Flexible Registration allows each catalog to display specialized home page content created in HTML. There are three ways to customize home page content, and Banner Flexible Registration will look for this content in the following manner:

1. Based on the catalog selected, if a catalog specific URL has been specified on the SFRAC_TLG form, this content will be displayed on the home page. For more information, please see [“Specify a catalog specific URL” on page 4-56](#).
2. If no URL has been specified for the catalog or the URL is not valid, Banner Flexible Registration will look in the \content folder for a file named `homePage_ID.html` where *ID* is the catalog ID, and display the content in this file. For more information, please see [“Create a catalog specific HTML file” on page 4-57](#).
3. If no URL has been specified for the catalog or the URL is not valid and a \content\homePage_ID.html file does not exist, Banner Flexible Registration will display the `homepage_default.html` file that is found in the \content folder. The content in the `homepage_default.html` file will display for all catalogs, and may be customized for your institution. For more information, please see [“Modify the default home page content” on page 4-57](#).

Specify a catalog specific URL

To specify a URL that contains the catalog specific content you would like to display on the Banner Flexible Registration Home page, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRAC_TLG).
2. Enter a catalog code.
3. On the Rules tab, enter the URL you would like to display on the Home page in the Home Page URL parameter field.

Note

The URL is case sensitive. ■

For more information, please refer to [“Externalize HTML content” on page 4-58](#).

4. Click  Save.

Create a catalog specific HTML file

To specify an HTML file that contains the catalog specific content you would like to display on the Banner Flexible Registration Home page, perform the following steps:

1. Create an HTML file for a catalog that contains the content you would like to display on the Banner Flexible Registration Home page.
2. Save the HTML file in the `\content` folder and name it `homePage_ID.html` where `ID` is the catalog ID. For example:
`homePage_SPRING2010.html`

Note

The URL is case sensitive and the catalog ID must use uppercase. ■

3. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
4. Enter the catalog code.
5. On the rules tab, enter `./content/homePage_ID.html` where `ID` is the catalog ID in the Home Page URL parameter field.
6. Click  Save.

Modify the default home page content

If no URL has been specified for the catalog or the URL is not valid and a `\content\homePage_ID.html` file does not exist, Banner Flexible Registration will display the `homepage_default.html` file that is found in the `\content` folder. The content in the `homepage_default.html` file will display for all catalogs, and may be customized for your institution.

To customize the `homepage_default.html` file, perform the following steps:

1. Customize the `homepage_default.html` file with the content you would like to display on the Banner Flexible Registration Home page.
2. Save the HTML file as `homepage_default.html`.

The `homepage_default.html` file will display for any catalog that does not have a URL or HTML file specified.

Externalize HTML content

Catalog specific and default HTML content pages can also be externalized. Externalizing these files prevents customization changes made to them from being lost upon a redeploy of the Banner Flexible Registration EAR file. For more information, please refer to [“Externalizing resources” on page 3-24](#) in Chapter 3.

HTML fragment support

In Banner Flexible Registration, HTML does not have to conform to the standard HTML format. Tags useful to a browser, such as <title>, <head>, <body>, and others are ignored. The absence of such tags allows for an institution to produce HTML fragments.

Supported HTML tags

The AS3 HTML Parser used in Banner Flexible Registration supports a subset of the standard HTML tags. Of those supported tags, only a subset of the available attributes are available.

The following table lists the HTML tags supported by the Flexible Registration HTML parsing engine:

HTML Tag	Supported Attributes	Notes
a	href target	
abbr		textDecoration is always set to underline
acronym		textDecoration is always set to underline
address		fontstyle is always set to italic
b		fontWeight is always set to bold
basefont		
big		fontSize is always set to 16 or 2 more than the parent element if the parent element exists
blockquote	paddingLeft paddingTop paddingBottom	
br		
button		

HTML Tag	Supported Attributes	Notes
cite		fontstyle is always set to italic
code		textFamily is always set to courier
del		textDecoration is always set to underline
dfn		fontstyle is always set to italic
div	align	
em		fontstyle is always set to italic
h1		paddingTop is always set to 20 fontSize is always set to 30 fontWeight is always set to bold paddingBottom is always set to 20
h2		paddingTop is always set to 20 fontSize is always set to 22 fontWeight is always set to bold paddingBottom is always set to 20
h3		paddingTop is always set to 20 fontSize is always set to 18 fontWeight is always set to bold paddingBottom is always set to 20
h4		paddingTop is always set to 20 fontSize is always set to 16 fontWeight is always set to bold paddingBottom is always set to 20
h5		paddingTop is always set to 20 fontSize is always set to 14 fontWeight is always set to bold paddingBottom is always set to 20
h6		paddingTop is always set to 20 fontSize is always set to 12 fontWeight is always set to bold paddingBottom is always set to 20
i		fontstyle is always set to italic
img	src	
ins		textDecoration is always set to underline

HTML Tag	Supported Attributes	Notes
kbd		textFamily is always set to courier
p	align	
q		
samp		textFamily is always set to courier
small		fontSize is always set to 16 or 3 less than the parent element if the parent element exists
span		
strike		textDecoration is always set to underline
strong		fontWeight is always set to bold
table	align bgcolor cellspacing	
TBODY		
td	rowspan colspan align valing	
text		
TFOOT		
th		fontWeight is always set to bold horizontal align is always set to center
THEAD		
tr	align valign	
tt		textFamily is always set to courier
var		fontstyle is always set to italic

Supported HTML attributes

The following lists the HTML attributes supported by the Flexible Registration HTML parsing engine:

- background-color
- color
- font-family
- font-size
- font-style
- font-weight
- height
- letter-spacing
- margin
- margin-bottom
- margin-left
- margin-right
- margin-top
- overflow
- padding
- padding-bottom
- padding-left
- padding-right
- padding-top
- text-align
- text-decoration
- text-indent
- vertical-align
- visibility (none)
- width

Image support

The following image formats are supported by Adobe Flash:

- JPEG
- Progressive JPEG

- PNG
- GIF

For more information about the supported image types, refer to the Adobe site http://livedocs.adobe.com/flex/3/htmlhelp.html?content=controls_16.html.

JavaScript support

The HTML Parser tool does not support JavaScript.

CSS support

The HTML Parser tool does not support embedded or linked CSS.

Support for Flexible Registration Style

To use the default Banner Flexible Registration style in the HTML content produced by your institution, simply ensure that all of the content you produce is wrapped in a `<homepage>` tag. For example:

```
<homepage class="homePageGlobal">  
HTML  
</homepage>
```

Homepage URL displayed on sign out

You may specify, for each catalog, the page that is displayed when a user signs out of the Banner Flexible Registration on the Flexible Registration Catalog Rules Form (SFRAC TLG). To set the log out page, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRAC TLG).
2. On the rules tab, enter the URL you would like to display when the user signs out of the application in the `URL Redirect on Sign Out` parameter field.

Note

The URL is case sensitive and must be a valid URL. ■

3. Click  **Save**.

Permalinks

A link can be created to allow students to access specific search results in Banner Flexible Registration. This link can allow your institution to push a course or group of courses to a targeted student population.

The following search criteria can be included in a link:

Search Text	Search Criteria	Valid Values	Example
crn	CRN	Any valid CRN value	crn:10138
term	Term Name	Text search	term:201009
subj	Subject Code	Any valid Subject Code from STVSUBJ	subj:MATH
camp	Campus Code	Any valid Campus Code from STVCAMP	camp:M
schd	Schedule Type	Any valid Schedule Type from STVSCHD	schd:L
im	Instructional Method	Any valid Instructional Method from GTVINSM	im:TR
mt	Meeting Days	Single characters representing each day of the week: <ul style="list-style-type: none"> • <i>U</i> - Sunday • <i>M</i> - Monday • <i>T</i> - Tuesday • <i>W</i> - Wednesday • <i>R</i> - Thursday • <i>F</i> - Friday • <i>S</i> - Saturday 	mt:MWF
timesa	Start After Time	Class starts after Text search Format: HHMM	timesa:1230 Note: This search will return all classes that start at 12:30 or later.

Search Text	Search Criteria	Valid Values	Example
timesb	Start Before Time	Class starts before Text search Format: HHMM	timesb:1300 Note: This search will return all classes that start at 13:00 or before.
timeeb	End Before Time	Class ends before Text search Format: HHMM	timeeb:1600 Note: This search will return all classes that end at 16:00 or before.
datesa	Start After Date	First date a course can start Text search Format: YYYYMMDD	datesa:20100901
datesb	Start Before Date	Last date a course can start Text search Format: YYYYMMDD	datesb:20101230
inst	Instructor	Text search	inst:Smith
numb	Course Number	Text search	numb:101
title	Course Title	Text search	title:Poetry
hrs	Credit Hours	Numeric value Note: A wildcard may not be used with a Credit Hour search.	hrs:3
desc	Course Description	Text search	desc:Algebra
os	Open Seats	Boolean <ul style="list-style-type: none"> • <i>Y</i> - will display only sections with seats available • <i>N</i> - will display both sections with seats available and sections without seats available 	os:Y

Search Text	Search Criteria	Valid Values	Example
levl	Course Level Code	Text search	levl:GR
attr	Course Attribute	Any valid attribute code from SFSRATTR	attr:HUMA

Create a permalink using the basic search

To create a permalink using the basic search criteria of Subject Code, CRN, Course Title, Course Number, Course Description:

1. Start with the base URL for the Banner Flexible Registration application for the selected catalog. For example:

```
http://www.ellucianuniversity.edu/index.jsp?frc=ExampleCatalog
```

2. Append your desired search criteria to the end of the URL. For example, to limit the search to all courses that contain the words *Business Management* in the Subject Code, CRN, Course Title, Course Number, and Course Description:

```
http://www.ellucianuniversity.edu/index.jsp?frc=ExampleCatalog&q=Business%20 Management
```

The following results may be returned:

Subject Code	CRN	Course Title	Course Number	Course Description
BUS	845048	Business Management Practices	422	Examines successful CEOs and the strategies that they employ in the workplace.
ACCT	922345	Small Business Accounting	241	Management and accounting techniques used in a small operation.

3. Save and publish the URL to the targeted student population.

Create a permalink using the advanced search

Permalinks support the ability to execute a search using the same search criteria available through advanced search.

To create a permalink using the advanced search, perform the following steps:

1. Start with the base URL for the Banner Flexible Registration application for the selected catalog. For example:

```
http://www.ellucianuniversity.edu/index.jsp?frc=ExampleCatalog
```

2. Append your desired search criteria to the end of the URL. For example, to limit the search to the subject of MATH:

```
http://www.ellucianuniversity.edu/  
index.jsp?frc=ExampleCatalog&q=subj:MATH
```

3. Append any additional search criteria to the end of the URL. For example, to limit the search to all MATH classes with open seats:

```
http://www.ellucianuniversity.edu/  
index.jsp?frc=ExampleCatalog&q=subj:MATH+os:Y
```

4. Save and publish the URL to the targeted student population.

Restrictions

The following restrictions must be considered when creating a permalink.

URL Encoding

The URL must not use any reserved characters in the search criteria. When using the following reserved characters as part of a text string, please make the following substitutions:

Character	Hex Substitution
Space (" ")	%20
Dollar ("\$")	%24
Ampersand ("&")	%26
Plus ("+")	%2B
Comma (",")	%2C
Forward slash/Virgule ("/")	%2F
Colon (":")	%3A
Semi-colon (";")	%3B

Character	Hex Substitution
Equals ("=")	%3D
Question mark ("?")	%3F
'At' symbol ("@")	%40

For example, to search for the following course title:

Great English Writers I: Shakespeare, Chaucer, and Dickens

you must enter the following to include the colon “:” and the commas “,” in your search:

```
&q=title:Great English Writers I%3A Shakespeare%2C Chaucer%2C
and Dickens
```

Note

Banner Flexible Registration will automatically encode any spaces between words in a text string, so there is no need to use the %20 hex encoding value for any spaces. ■

Difference between “+” and “%20”

Use “+” to distinguish between different fields such as "subj:SUBJECT+crn:CRN". Use “Spaces (or %20)” to distinguish terms in a field. For example, "subj:SUBJECT+crn:CRN1%20CRN2".

Some fields like **Subject** treat the value as a single term. Other fields such as **CRN** support a list.

Note

Do not use spaces in URLs. The HTTP specification does not allow for spaces in URL. Many, but not all, browsers will allow the user to include spaces as the spaces improve the readability of the address. ■

Date Range Search

To search for a course that starts between a range of dates, you must include both the *datesa* and *datesb* search text. For example, to search for course that begin between September 15, 2010 and September 30, 2010, include the following in the permalink:

```
&q=datesa:20100915+datesb:20100930
```

Wildcards

Wildcards may be included in any text string search as long as two characters are included along with the wildcard.

```
&q=inst:Sm*
```

Create a permalink on the Landing Page

Permalinks can be used on the Flexible Registration landing page. To prevent the application from reloading, the institution must set up the Permalink with special syntax in the HTML code. This syntax only applies to permalinks on the landing page. This syntax should not be used if you are creating the permalink on a non-Flexible Registration web page.

When creating the anchor tag on the landing page use the following syntax, inside the "href" property (instead of the URL):

```
javascript:ContentManager.doPermaLinkBasedSearch('[SEARCH TEXT]')
```

The syntax should change the anchor tag to call a Javascript function instead of including the URL to the index.jsp.

The following examples demonstrate how this should be done:

This example creates a link to classes that have Math in the title:

```
<ahref="javascript:ContentManager.doPermaLinkBasedSearch('math');">Math</a>
```

The above example is demonstrating a basic search with the permalink.

The next example creates a link to classes that occur in a given month:

```
<ahref="javascript:ContentManager.doPermaLinkBasedSearch('datesa:20120915+datesb:20120930');">Classes for October 2010</a>
```

The above example is demonstrating how to include advanced search syntax in the permalink.

Searching

Banner Flexible Registration contains both a basic search, that searches a limited set of search attributes, and an advanced search that allows a student to fully customize a search for classes.

Basic search

The Basic Search that the learner initiates is a single text box that will search:

- Subject Code
- CRN

- Title
- Course number
- Description

When more than one search word is used, Banner Flexible Registration will return classes that contain all search words among all of the searchable fields.

For example:

If you search on *Business Management*, the following results will be returned:

Subject Code	CRN	Course Title	Course Number	Course Description
BUS	84504	Business Management Practices	422	Examines successful CEOs and the strategies that they employ in the workplace.
ACCT	92234	Small Business Accounting	241	Management and accounting techniques used in a small operation.

 **Note**

An implicit wildcard character is not appended to a user's search. Therefore, a search on *bu* may not return any search results, but a search on *bu** using a wildcard will return any course that includes the word *Business*. ■

CRN search in basic search

You can search for multiple CRNs by entering the CRNs, separated by spaces, in the search box. The search is performed for each CRN and the CRN search results are displayed at the top of the basic search results.

Basic search information

The basic search in Banner Flexible Registration processes search requests using the following settings:

- Non-alphanumeric characters will be removed from the search string, including single and double quotes.
- SQL key words will be removed from the search string. (SELECT, DROP, DELETE, UPDATE, INSERT, and so on).

- Wildcard characters ('*' character) will be translated into Oracle Full Text index wildcards ('%' character). For more information on searching with wildcards, please see [“Wildcard” on page 4-71](#).
- Whitespace will be ignored.
- A CRN search is performed on each search string.

Sort order for search results

The search results are displayed in the following sort order, by default:

- The search results are grouped into CRN search results and basic search results, with the CRN search results being displayed first.
- Within the CRN search results and the basic search results, the sort order is as follows:
 - Term
 - Subject Code
 - Course Number
 - Section Number

Advanced search

The advanced search in Banner Flexible Registration allows you to limit your search to return all matches found using the following information:

- Campus
- Class start date
- Class times
- CRN
- Course attribute
- Course level
- Course number
- Credit hours
- Description
- Instructional method
- Meeting days
- Only show sections with seats available?
- Instructor (Last Name)

- Schedule type
- Subject Code
- Term
- Title

Operators

The following boolean operators can be used in Banner Flexible Registration when using multiple search words:

- and
- or

and

The and boolean operator is the default operator when multiple search words are used. For example:

Searching for *Business and Accounting* will return the same results as searching for *Business Accounting*. Both of these searches will return all classes that contain both the words Business and Accounting.

or

The or boolean operator can be used to search for one search word or another. For example:

Searching for *Business or Accounting* will return all classes that contain the word Business and all classes that contain the word Accounting. This search will also return all classes that contain both the words Business and Accounting.

Wildcard

An asterisk (*) can be used as a wildcard to substitute for one or more characters in a search word. When searching using a wildcard, Banner Flexible Registration requires that at least two non-wildcard characters be used along with a wildcard.

For example:

Searching for C* will not return any results because Banner Flexible Registration requires at least two non-wildcard characters in a search.

Searching for Ch* will return the following results:

Chemical Engineering

Chemistry

Organic Chemistry

CHEM 101

Searching for Ch*y will return the following results:

Chemistry

Organic Chemistry

Term searching

The term dropdown list displays all the terms that have classes in this catalog. The term field can be used to specify the term that will be used by the basic search.

For example, if a student selects the Fall 2009 term from the dropdown and then enters Accounting in the basic search field, all accounting classes in the Fall 2009 term will be returned in the results. If the term is not selected first, then all Accounting classes for all terms in the catalog will be returned.

Note

This search term pull down is only displayed if the Catalog rule (SFRACTLG) for Allow Term Filter = Y. ■

Maximum search results

You may specify, for each catalog, the maximum number of search results returned for any search on the Flexible Registration Catalog Rules Form (SFRACTLG). For performance benefits, this should be set to a reasonable number. By default, it is set to 50 results. To set a maximum search results limit, perform the following steps:

1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
2. Enter a value for Search Results Limit.
3. Click  **Save**.

Customizing advanced search fields

Banner Flexible Registration allows your institution to control the advanced search capabilities of the application. To customize advanced search, perform the following steps:

1. Access the Flexible Registration Advanced Search Rules Form (SFRASRCH).
2. Select the DEFAULT value from the **Advanced Search Code** field.
3. Click **Sort** to specify the order that the search attributes are displayed on the advanced search window.

The following are the search attributes:

- Subject
 - Schedule type
 - Campus
 - Class start date
 - Class times
 - Course number
 - Course level
 - Credit hours
 - CRN
 - Description
 - Instructional method
 - Instructor (Last Name)
 - Meeting days
 - Only show sections with seats available?
 - Course attribute
 - Term
 - Title
4. Check the **Visible** check box to specify which fields are available to the user on the Advance Search window.

5. Check the **Default Field** check box to display the attribute as a default selection on the Advance Search window.
6. Click  **Save**.

Search processing

Search is processed in the following manner:

1. Capture the search string.
2. Remove non-alphanumeric characters (single/double quotes, dashes, dots, etc.). The following regular expression should suffice: `[^a-zA-Z0-9*]`
3. Remove SQL reserved words (SELECT, DROP, DELETE, UPDATE, INSERT, etc.).
4. Parse the remaining search string for supported operators and wildcards.
5. Split the search string by spaces into string tokens.
6. Perform a CRN search for each string token.
7. Parse the resulting search string components injecting the default AND operator between search strings.
8. Build any required SQL code with the resulting search string criteria.
9. Perform the Basic Search.
10. Display search results, with CRN Search results at the top of the list.

For example:

Example 1:

Search String: Drop Small Business Accounting

1. Drop Small Business Accounting --> search string captured.
2. Drop Small Business Accounting --> no non-alphanumeric characters found, so none removed.
3. Small Business Accounting --> removed SQL reserved word “Drop”.
4. Small Business Accounting --> no supported operators or wildcards found, so none replaced.

5. Small Business Accounting --> CRN Search performed, no CRNs, so no CRN search results.
6. Small AND Business AND Accounting --> broke apart the remaining string tokens and inserted the default AND operator between them.
7. Small AND Business AND Accounting --> Basic Search performed.

Example 2:

Search String: 123456

1. 123456 --> search string captured.
2. 123456 --> no non-alphanumeric characters found, so none removed.
3. 123456 --> no SQL reserved words found, so none removed.
4. 123456 --> no supported operators or wildcards found, so none replaced
5. 123456 --> CRN search performed for 123456. CRN search results returned for 123456.
6. 123456--> basic search results returned for 123456.

Browse by Subject search

The Browse By Subject option enables students to search for classes based on the subject name of the class.

Perform the following steps to do a Browse By Subject search:

1. In the home page, click the **Browse By Subject** option.

All the subjects are listed under Browse Subjects on the left pane.

2. Select a subject from the list.

The classes for the selected subject are displayed under Browse Results on the right pane.

3. Click **Details** if you want to view the details of a course.
4. Click **Add** if you want to add a course to the Schedule Builder.

Authentication

Banner Flexible Registration uses the same model and options as Banner Self-Service for authentication. Authentication in Self-Service Banner is accomplished either through a proprietary ID/PIN mechanism, or through an LDAP bind. These options are system-wide, and only one may be used. Banner Flexible Registration will use the same options that are specified for Banner Self-Service.

- To configure authentication using the ID/PIN, please refer to the *Banner General User Guide*.
- To configure authentication using LDAP, please see the “Implementing Single Sign-On for Self-Service Banner” chapter in the *Banner General Middle Tier Implementation Guide*.

If a Central Authentication Service (CAS) service is deployed in your infrastructure, CAS can be used for authentication. If CAS is enabled, it will remove the authentication function from Flexible Registration and will utilize the sign-in interface deployed by CAS. CAS becomes an alternative to doing authentication within Flexible Registration. For end-users, this means they will not see a User Name and Password form in the Flexible Registration user interface. It will be replaced with instructions on logging into CAS. Please see the section called Single Sign-On for more information.

Single Sign-On (SSO)

Banner Flexible Registration supports Single Sign-On (SSO) with other applications in your environment. Using SSO enables a user to automatically sign in to an application without having to enter their username and password again. Banner Flexible Registration supports SSO in two ways: using CAS or application specific connections. Both are described in the following sections.

For information on using CAS see the CAS section.

For information on SSO from Banner Self Service, see the *Single Sign-On* section under *Banner Self Service Integration*.

For information on SSO from Luminis, see the *Luminis Single Sign-On* section.

Central Authentication Service (CAS)

CAS is directly supported by Banner Flexible Registration. CAS is a popular application that enables single sign-in across web based application in your environment. For more information on CAS, please refer to the CAS open source project at <http://www.jasig.org>.

Banner Flexible Registration uses CAS for authentication for both signing in while within the application and navigating to Banner Flexible Registration from another CAS supported application.

When CAS is enabled, the user is not be presented with a Banner Flexible Registration sign-in page, but will be redirected to the CAS login web page. When the sign-in has been completed, the user will be redirected back to Banner Flexible Registration and be automatically signed in.

When CAS is enabled and the user is navigating to Banner Flexible Registration from a CAS supported web site, they should be directed to a secure URL. Banner Flexible Registration will then work with CAS to determine if the user is already signed in to CAS and will handle the authentication into the Banner Flexible Registration application.

Banner Flexible Registration also supports the single sign-out capability of CAS as well. When CAS is enabled and the user chooses to sign out of Banner Flexible Registration, they will be presented the option of not only signing out of Banner Flexible Registration, but also signing out of CAS.

Prerequisites

You must install and configure CAS before configuring Banner Flexible Registration for CAS. Please refer to the chapter or appendix in the BEIS Handbook that details CAS installation and configuration.

Banner Flexible Registration is following the design described in the BEIS Handbook, using the *samlValidation function* of CAS. The *samlValidation* function will return back to Banner Flexible Registration a *UDC_IDENTIFIER* attribute used to lookup a user's Banner ID. Therefore, CAS must be setup and the *UDC_IDENTIFIER* should be published through the SAML validation routine prior to trying CAS with Banner Flexible Registration.

The CAS support in Flexible Registration has been tested with the Luminis V portal and CAS configuration as well as with an independent CAS 3.5.0 server. Other configurations may work, but some may require additional configuration. Please note that the Flexible Registration components require the *UDC_IDENTIFIER* attribute, as described, and requires the *samlValidate* service on the CAS server.

CAS Setup

CAS Server URL

To enable CAS in Banner Flexible Registration, use the Flexible Registration Configuration Rules form (SFRACNFG), and populate the **CAS server URL** field with the full url to your CAS service. The **CAS server URL** field should be the full URL, including https, the port number, the machine name, and the CAS application root. For example, <https://ellucianuniversity.com:443/cas>. Please note that SSL must be enabled on the CAS server and the Banner Flexible Registration server so that the authentication process can be secure.

The **Flexible Registration Deployment URL** field should also be populated with the full URL to your Banner Flexible Registration application. This URL should include the

protocol, port number, machine name, and the application root only. For example, `https://ellucianuniversity.com:443/flexibleregistration`. This url should not contain `index.jsp` because it does not represent a landing page URL, but simply the application root URL. If you are in a load balanced environment, this should be the main published url, not an individual machine url.

Logout Redirect URL

As mentioned, Banner Flexible Registration also supports the single-sign out capability of CAS by presenting the user a link to the logout function on the CAS server. CAS optionally supports a redirect URL that you may want to redirect all users to after they sign out. This URL is not a Banner Flexible Registration URL but some other URL you may have. The Logout Redirect URL setting in the SFRACNFG form is the URL you wish to redirect a user to after the sign out process has completed in CAS. This URL should be the fully qualified URL to a page. This should include the protocol, port number, machine name, and application root, including a page if required.

For example, setting to `http://ellucianuniversity.com/home.html` will redirect all users to that page when doing a CAS sign out from Banner Flexible Registration.

CAS Secure URL

In order to access Banner Flexible Registration through SSO, you need to provide users a CAS protected URL. This URL is a special URL that will enforce that the user is signed in through CAS. The following is the structure and example of a CAS protected url:

```
https://ellucianuniversity.com:443/flexibleregistration/cas/  
index.jsp?frc=A0
```

```
Structure [PROTOCOL]://[MACHINE]:[PORT]/[APPLICATION]/cas/  
index.jsp?frc=[CATALOG CODE]
```

If you want to publish a URL to users who may not have accounts at your institution, you should still publish the normal Banner Flexible Registration URL. For example:

```
https://ellucianuniversity.com:443/flexibleregistration/  
index.jsp?frc=A0
```

The only difference between the CAS Secure URL and the normal URL is the `/cas` before the `index.jsp` link.

If you are using the Multi-Entity Processing (MEP) support features of Banner Flexible Registration, the CAS Secure URL should look like the following example:

```
https://ellucianuniversity.com:443/flexibleregistration/cas/  
institution/CAMP1/catalog/A0/
```

Considerations

If you choose to use CAS for Single Sign-In, the following items need to be taken into consideration.

Application login is disabled

If CAS is setup and in use, it invalidates the need for an individual application to have its own login process. This is, in fact, one of the benefits of CAS - removing the need for each individual application within an organization to handle username and password functionality.

Therefore, when CAS is enabled, Banner Flexible Registration will no longer display a form on the landing page or in the Sign-In pop up dialog for signing in. Users will now be given instructions and a URL link to click on that will redirect them to CAS for sign-in. Once the user is signed-in, they will be directed back to Banner Flexible Registration.

This does not impact new user creation.

Disable Change Password Link

The Change Password link should be disabled in the SFRAC TLG form. Since CAS will be handling authentication requests, Banner Flexible Registration cannot determine the method for changing a password. Therefore, the link should be hidden from logged in users.

Kiosks or Public Computers

If Flexible Registration is accessed from a kiosk or a public computer, you may want to automatically have CAS sign out when the student signs out of Flexible Registration. Normally, when a student signs out of the application, they are signed out of Flexible Registration, but not signed out of CAS. Therefore, they can continue to single sign-in to other applications in the same browser instance. This may not be feasible in some cases. Therefore, you may want to force the student to be signed out of all applications and CAS when they sign out of Flexible Registration. To do this, you can use the SFRAG CNF form and place the logout URL to CAS in the **URL Redirect on Sign Out** for the catalog.

This will now force the Flexible Registration application to call the CAS sign out process, whenever the student signs out of Flexible Registration.

Define CAS managed service

Banner Flexible Registration must be defined as a CAS managed service in order to be protected by CAS.

Use the following steps to define the CAS managed service.

1. Open a browser and go to the CAS server management page, which is at a URL similar to the following:

```
http(s)://<CAS server>:<port>/cas/services/manage.html
```

2. Enter a valid administrator user name and password (obtained from the CAS administrator). The Services Management page is displayed.
3. Click the Add New Service tab.
4. Add a new service by entering the following values

Name	<i>Flexible Registration</i>
Service URL	<i>https://<server>:<port>/registration application name>/**</i>
Description	<i>Protecting Flexible Registration via CAS</i>
Attributes	<i>Select UDC_IDENTIFIER</i>

5. Click **Save Changes**. The CAS server is now protecting the Flexible Registration URL.

LDAP configuration on Oracle 11g Application Server

If you are using Oracle Database 11g Release 1 (11.1) or greater, you will need to grant access to several Flexible Registration database IDs to the network APIs for LDAP. In order to do this, you will need to add these accounts to your access control list in Oracle. This is a documented change in the way Oracle manages access.

For Flexible Registration, you will need to run the following command for the FLEXREG, FLEXREG_USER, and FLEXREG_USER_ROLE accounts, respectively:

- `dbms_network_acl_admin.add_privilege('YOUR_ACL_XML', 'FLEXREG', TRUE, 'connect');`
- `dbms_network_acl_admin.add_privilege('YOUR_ACL_XML', 'FLEXREG_USER', TRUE, 'connect');`
- `dbms_network_acl_admin.add_privilege('YOUR_ACL_XML', 'FLEXREG_USER_ROLE', TRUE, 'connect');`

The 'YOUR_ACL_XML' should be the XML filename you have used to setup the ACL.

Banner Self-Service integration

Banner Flexible Registration is fully integrated with Banner Self-Service.

Single Sign-on

Banner Flexible Registration supports Single Sign-on from Banner Self-Service. Spring security filters are configured to read and validate the session cookie created during the Self-Service session.

The login package `twbkwbis` was modified to use the domain defined in the Web Tailor `CPCOOKIEDOMAIN` parameter. To enable single sign-on, modify this parameter to use your specific domain. For example:

Parameter Name: `CPCOOKIEDOMAIN`

Parameter Value: `yourdomain.edu`

Self-Service menu links

During the Flexible Registration installation process, a new menu for Flexible Registration can be added to the Student Main menu. For more information, please see the *Banner Flexible Registration Upgrade Guide*.

The Flexible Registration menu contains links that can be customized to point to the URLs you have created for your Flexible Registration catalogs. Students will be able to use these links to access Flexible Registration from Student Self-Service using single sign-on.

Note

It is not required that you update the Student Self-Service menus to include links to Flexible Registration catalogs. As an alternative or in addition to providing links through Self-Service, you may publish the catalog links through other means, such as making them available on your Institution's home page. Then students would be able to access Flexible Registration directly, without needing to first sign in to Student Self-Service. ■

Manual Web Tailor updates

To update Self-Service Menu links for your catalogs in Web Tailor Administration:

1. As the Web Tailor administrator, log in and select the Web Tailor Administration tab to access the Web Tailor main menu.
2. Select Web Tailor Parameters.

3. Click on the **Add a new Web Tailor Parameter** button at the bottom of the Customize a Web Tailor Parameter page.
4. Add a name for each catalog in the **Parameter Name** field.

For example, *FLEXREG_CAT_A0_URL*. Note the parameter used in this step is also referenced in Step 18.
5. Add a web address for the respective catalog in the **Parameter Value** field.

The Flexible Registration catalog URL is `http://xxxxxxx/index.jsp?frc=A0`
6. Repeat this step for each catalog.
7. Return to the Web Tailor main menu.
8. Select Web Menus and Procedures.
9. Click on the `bmenu.P_StuMainMnu` procedure name in the list on the Web Menus and Procedures page.
10. Click on the **Customize Menu Items** button at the bottom of the Customize a Web Menu or Procedure page.
11. Click on the **Copy Baseline to Local** button on the Reorder or Customize Menu Items page.

If the *Copy Baseline* option is not a valid choice, click the **Add a New Menu Item** button.
12. Add the `bmenu.P_FlexRegMnu` menu item.

The Link Text property should be given. For example, *Catalog A0*.
13. Ensure that the **Database Procedure** and **Submenu Indicator** options are checked.
14. Navigate back to the Web Menus and Procedures page.
15. Click on the `bmenu.P_FlexRegMnu` procedure name in the list.
16. Click on the **Customize Menu Items** button.
17. Click on the **Copy Baseline to Local** button.
18. Add a new menu item that points to the new procedure name created for each catalog.

The following is the syntax for this menu item.

```
http://[your.server.name:port]/  
twbkwbis.P_launchapp?url_param=[web tailor parameter  
name]&logout_ind=Y
```

The server name above should be the fully qualified URL to your server running Web Tailor.

The Web Tailor Parameter in the above link should be the Parameter Name from Step 4. For example:

```
http://[your.server.name:port]/  
twbkwbis.P_launchapp?url_param=FLEXREG_CAT_A0_URL&logout_ind=Y
```

 **Note**

Make sure you do not omit the `logout_ind=Y` parameter. Omitting the `logout_ind=Y` parameter will allow any user to enter Self-Service Banner without requiring the user to authenticate. ■

The Link Text property should also be supplied.

19. Ensure that the **Database Procedure** and **Submenu Indicator** options are unchecked.
20. Select Web Tailor Administration.
21. Click Web Menus and Procedures.
22. Click the **CREATE** button.

PAGE NAME = twbkwbis.P_launchapp

DESCRIPTION = twbkwbis.P_launchapp

ENABLED INDICATOR is CHECKED.

Prohibit student registration through Student Self-Service (Optional)

To update menus through Web Tailor Administration to prohibit the use of Student Self-Service pages related to registration, perform the following steps:

1. Disable the Get Course page, by performing the following steps:
 - 1.1. From Web Tailor menu select Web Menus and Procedures.
 - 1.2. Search for and select `bwckgens.P_RegsGetCrse`.
 - 1.3. Copy baseline to local.

1.4. Uncheck the student role.

1.5. Submit changes.

 **Note**

The `bwckgens.P_RegsGetCrse` package has both a student and a faculty role. Ensure that only the student role is disabled so faculty will still be permitted to access the page. ■

2. Disable the Alternate Pin Verification page, for add/drop, by performing the following steps:
 - 2.1. From Web Tailor menu select Web Menus and Procedures.
 - 2.2. Search for and select `bwskfreg.P_AltPin`.
 - 2.3. Copy baseline to local.
 - 2.4. Uncheck the enabled indicator on this package.
 - 2.5. Submit changes.
3. Disable the Add or Drop Classes page by performing the following steps:
 - 3.1. From Web Tailor menu select Web Menus and Procedures.
 - 3.2. Search for and select `bwskfreg.P_AddDropCrse`.
 - 3.3. Copy baseline to local.
 - 3.4. Uncheck the enabled indicator on this package.
4. Disable the Look Up Classes page, by performing the following steps:
 - 4.1. From Web Tailor menu select Web Menus and Procedures.
 - 4.2. Search for and select `bwskfcls.P_GetCrse`.
 - 4.3. Copy baseline to local.
 - 4.4. Uncheck the enabled indicator on this package.
5. Disable the Look Up Classes item on the Registration menu by performing the following steps:
 - 5.1. From Web Tailor menu select Web Menus and Procedures.
 - 5.2. Search for and select `bmenu.P_RegMnu`.

5.3. Copy baseline to local.

5.4. Disable sequence 3 - `bwskfcls.p_sel_crse_search`.

Prohibit students from adding/dropping classes via Registration Tools channel (Optional)

To prevent students from adding or dropping classes via the Registration Tools channel, perform the following steps:

1. Log into Self-Service Banner as a Web Tailor Administrator and Banner Channel Administrator.
2. From the Web Tailor menu select Channel Preferences.
3. Select `SN_REGTOOLS`.
4. Select the row where `Key = ADDDROP`.
5. Update the display link to *No*.
6. Click **Submit**.

Registration locks

Banner Flexible Registration automatically releases a Banner registration lock when the Final Invoice is displayed or if a registration is canceled. Under all other circumstances, when the application requires a lock, it will be released when the timeout expires.

Banner Flexible Registration does not automatically release registration locks when a user logs out of the system. Like Banner Self-Service, the Registration Access Lock for Flexible Registration will be released when the `REGACCTIME GTVSDAX` time limit rule expires.

The `GTVSDAX` entry for Registration Access Time Out must contain a numeric value that represents the number of minutes a session may remain idle before the session is considered to be timed out. The delivered value is *UPDATEME*, which must be modified at your institution. Change this value to a numeric value, for example, 15 for 15 minutes.

Note

If no value is entered, the default time of 60 minutes will be used. ■

When the session times out, it will not be terminated. Upon time out, another user will be permitted to initiate a registration session for the same ID and term. In this instance, the second user will be in control of the student's registration records. The following message will display to the user:

Another registration session intervened for this ID and TERM while your session was idle. Please try again later.

When the original user resumes work on the student's registration records a Rollback will take place.

 **Note**

The registration access lock performs a group cart validation in case of group registration. If a group cart owner is signing up other students, they will not be able to register the students who have a registration lock. Also, if the group cart owner is signing up a student for a class, the student will not be able to register by himself at the same time as the student will have a registration lock. ■

CAS Enabled Banner Self Service

If Banner Self-Service is CAS enabled, Flexible Registration can participate in this SSO exchange through CAS. Please refer to the Single Sign-In section with CAS for information. A URL link to Flexible Registration can be made using the secure URL format.

Luminis integration

Banner Flexible Registration provides support for user authentication using Luminis to generate ID and passwords, as well as using Single Sign-on from Luminis.

Luminis authentication

Banner Flexible Registration allows learners to log in through the Banner Flexible Registration web interface using IDs and passwords generated by Luminis.

Using Luminis IV

Luminis IV support must be enabled in both Banner Flexible Registration and Web Tailor to provide a successful login through Banner Flexible Registration when using Luminis IV.

To enable Luminis IV support, perform the following steps:

1. In Web Tailor, set the *LDAPL4FLAG* parameter to *Y*.
2. In Banner Flexible Registration, check the **Using Luminis IV** indicator on SFRACNFG.

Luminis single sign-on

Banner Flexible Registration supports single sign-on (SSO) from Luminis IV and Luminis V to Banner Flexible Registration.

Luminis IV

If CAS is not enabled or you are using Luminis IV:

 **Note**

Single sign-on from Banner Flexible Registration to Luminis is not provided or supported. ■

For more information, please see the *FAQ #1-8I70AY Banner Flexible Registration Implementation Options on the Customer Support Center* for instructions and configuration files used to enable SSO from Luminis to Banner Flexible Registration.

Luminis V

Luminis V uses CAS for authentication and single sign-in. Please refer to the Single Sign-In section with CAS for information.

Once CAS is enabled for Flexible Registration, SSO can be achieved from Luminis to Flexible Registration through the secure URL format. You can simply create a URL link to Flexible Registration in Luminis to the secure URL in Flexible Registration. Once the user clicks that link, the SSO process will be initiated.



5 Processes

This section contains information about processes generated by the Banner Flexible Registration System.

List of Processes

A list of all Flexible Registration processes. Included for each process are its code, name, and description.

Processes

Information about each report or process followed by an output sample. For each report or process, the following information is provided:

- A brief summary of the contents or purpose of the process
- The description, values, must-enter status, default value, and validation source of each of the process parameters
- The order in which data is organized and presented on the output
- The forms that collect the data appearing in the output
- When or how often the process is run, or the circumstances under which it is most often requested
- The physical form in which the data is delivered to you
- A detailed description of each field on the output

List of Processes

The Banner Flexible Registration System contains the process shown in the chart below.

Report/ Process	Name	Description
SFKRGADJ	Flexible Registration Fee Adjustment Process	Executes the group registration fee adjustments. Results are posted to the GJIREVO form. Additional details can be found by using the SFFIADAR form.



Flexible Registration Fee Adjustment Process

Description Executes the group registration fee adjustments. Results are posted to the GJIREVO form. Additional details can be found by using the Flexible Registration Adjustment Inquiry Form (SFFIADAR).

Parameters	Name	Description/Values	Required?	Default	Validation
01	ID Number	The ID of the registrant or cart owner to process.	No	None	SOAIDEN
02	Term Code	The Term Code to process adjustments on.	Yes	None	STVTERM
03	Shopping Cart ID Part 1	The first part of the Shopping Cart ID to look for.	No	None	
04	Shopping Cart ID Part 2	The second part of the Shopping Cart ID to look for.	No	None	
05	Selection Identifier	Code that identifies the sub-population to work with.	No	None	
06	Application Code	General area for which selection id was defined.	No	None	
07	Creator ID	The user id of the person creating the sub-population rules.	No	None	None
08	User	The user id of the person who ran the Population Selection.	No	None	

Sort Order

Not applicable

Frequency

On demand.

Report Format

Standard, horizontal.

Fields

Not applicable.

Report Sample—Flexible Registration Fee Adjustment Process (SFKRGADJ)

```
REPORT CONTROL INFORMATION * * *                                     PAGE      1
Process.....: SFKRGADJ
Release.....: 8.6
One Up Number.....: 84202
Activity Date.....: 26-MAR-2013 11:50:33
ID.....:
Term.....: 201341
Cart ID.....:
Selection Id.....:
Application Code.....:
Creator.....:
User.....:
There were 0 adjustment rows created.
Please refer to SFFIADAR for more information.
```



6 Third Party Software Licenses

Banner Flexible Registration uses the following third party software licenses:

Sun Microsystems, Inc.

The Binary Code License Agreement may be found at the following URL:

<http://www.java.com/en/download/license.jsp>

Component	Version	URL
activation-1.1.jar	1.1	http://java.sun.com/products/archive/javabeans/jaf11.html
jms-1.1.jar	1.1	http://java.sun.com/products/jms
jmxri-1.2.1.jar	1.2.1	http://java.sun.com/javase/technologies/core/mntr-mgmt/javamanagement/
jmxtools-1.2.1.jar	1.2.1	http://java.sun.com/javase/technologies/core/mntr-mgmt/javamanagement/
jta-1.0.1B.jar	1.0.1B	http://java.sun.com/javaee/technologies/jta/index.jsp
mail-1.4.jar	1.4	http://java.sun.com/products/javamail/
persistence-api-1.0.jar	1	http://java.sun.com/javaee/technologies/persistence.jsp

Apache License version 2.0

The Apache license may be found at the following URL:

<http://www.java.com/en/download/license.jsp>

Component	Version	URL
axiom-1.2.5.jar	1.2.5	http://ws.apache.org/axis/
axis-saaj-1.3.jar	1.3	http://ws.apache.org/axis/
cglib-2.1_3.jar	2.1.3	http://cglib.sourceforge.net/
commons-beanutils-core-1.7.0.jar	1.7.0	http://commons.apache.org/beanutils/
commons-codec-1.3.jar	1.3	http://commons.apache.org/codecs/
commons-collections-3.1.jar	3.1	http://commons.apache.org/collections/
commons-dbcp-1.2.2.jar	1.2.2	http://commons.apache.org/dbcp/
commons-discovery-0.2.jar	0.2	http://commons.apache.org/discovery/
commons-fileupload-1.2.jar	1.2	http://commons.apache.org/fileupload/
commons-httpclient-3.1.jar	3.1	http://hc.apache.org/httpclient-3.x/
commons-io-2.0.jar	2.0	http://commons.apache.org/io/
commons-lang-2.4.jar	2.4	http://commons.apache.org/lang/
commons-logging-1.1.1.jar	1.1.1	http://commons.apache.org/logging/
commons-pool-1.3.jar	1.3	http://commons.apache.org/pool/
ehcache-1.2.3.jar	1.2.3	http://ehcache.org/
kaptcha-2.3.0.jar	2.3.0	http://code.google.com/p/kaptcha/
log4j-1.2.15.jar	1.2.15	http://logging.apache.org/log4j/1.2/index.html

Component	Version	URL
opensaml1-1.1.jar	1.1	http://www.apache.org/licenses/LICENSE-2.0.txt
spring-aop-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-asm-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-beans-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-context-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-context-support-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-core-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-expression-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-jdbc-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-jms-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-orm-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-oxm-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-security-config-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-security-core-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-security-core-tiger-2.0.4.jar	2.0.4	http://www.springframework.org
spring-security-web-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org

Component	Version	URL
spring-test-2.5.5.jar	2.5.5	http://www.springframework.org
spring-tx-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-web-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-webmvc-3.0.5.RELEASE.jar	3.0.5	http://www.springframework.org
spring-ws-core-2.0.2.RELEASE.jar	2.0.2	http://www.springframework.org
stax-api-1.0.1.jar	1.0.1	http://stax.codehaus.org/Home
wstx-asl-3.2.6.jar	3.2.6	http://woodstox.codehaus.org/
xmlsec-1.4.3.jar	1.4.3	http://santuario.apache.org/java143releasenotes.html
mouseWheelEnabler.as	1.0.0	http://code.google.com/p/flex-wmode-mousewheel-handler-example/

Berkeley Software Distribution (BSD)

The Berkeley Software Distribution license may be found at the following URL:

<http://www.opensource.org/licenses/bsd-license.php>

Component	Version	URL
antlr-2.7.6.jar	2.7.6	http://www.antlr2.org/download.html
asm-1.5.3.jar	1.5.3	http://asm.ow2.org/
asm-attrs-1.5.3.jar	1.5.3	http://asm.ow2.org/
dom4j-1.6.1.jar	1.6.1	http://www.dom4j.org/
As3corelib-0.92.swc	0.92	http://code.google.com/p/as3corelib/
HTMLParser.as	1.0	http://sourceforge.net/projects/as3htmlparser

Adobe Cairngorm license

The Adobe Cairngorm license may be found at the following URL:

<http://opensource.adobe.com/wiki/display/cairngorm/License>

Component	Version	URL
Cairngorm-1.0.0.swc	1.0.0	http://opensource.adobe.com/wiki/display/cairngorm/Cairngorm

Common Public license

Version 1.0 of the Common Public license may be found at the following URL:

<http://www.opensource.org/licenses/cpl1.0.php>

Component	Version	URL
junit-4.4.jar	4.4	http://www.junit.org/
wsdl4j-1.5.2.jar	1.5.2	http://sf.net/projects/wsdl4j

Creative Commons Attribution license

Version 3.0 of the Creative Commons Attribution license may be found at the following URL:

<http://creativecommons.org/licenses/by/3.0/>

Component	Version	URL
Puremvc-2.0.4.swc	2.04	http://puremvc.org

Eclipse Public license

Version 1.0 of the Eclipse Public license may be found at the following URL:

<http://www.eclipse.org/legal/epl-v10.html>

Component	Version	URL
Oval-1.31.jar	1.31	http://oval.sourceforge.net

Lesser General Public license

The Lesser General Public license may be found at the following URL:

<http://www.gnu.org/copyleft/lesser.html>

Component	Version	URL
hibernate-3.2.7.ga.jar	3.2.7	https://www.hibernate.org/
hibernate-annotations-3.2.1.ga.jar	3.2.1	https://www.hibernate.org/

MIT license

The MIT license may be found at the following URL:

<http://www.opensource.org/licenses/mit-license.php>

Component	Version	URL
As3ds-1.0.0.swc	1.0.0	http://code.google.com/p/as3ds
Jquery.form.js	3.0.9	http://jquery.malsup.com/form/
mustache.js		https://github.com/janl/mustache.js

Public Domain

The following public domain component was used:

Component	Version	URL
Aopalliance-1.0.jar	1.0.0	http://aopalliance.sourceforge.net

JA-SIG license

The JA-SIG license may be found at the following URL:

<http://www.jasig.org/cas/license>

Component	Version	URL
cas-client-core-3.2.1.jar	3.2.1	http://www.jasig.org/cas/license



A Banner Flexible Registration Configuration Validation



The Banner Flexible Registration Configuration Validation process allows you to validate your Banner Flexible Registration data when configuration changes are made.

What configuration settings are checked?

Each configuration check script is targeted at a general domain or area of the application configuration. The scripts that are executed are as follows:

Order	Test Script Name	Description
1	<code>sf_globl_chk.sql</code>	Validates global configuration settings.
2	<code>sf_frc_chk.sql</code>	Validates catalog based configurations.
3	<code>sf_sdax_chk.sql</code>	Validates GTVSDAX rules.
4	<code>sf_term_chk.sql</code>	Validates the configuration of term records for each catalog.
5	<code>sf_profile_chk.sql</code>	Validates the profile configuration rules.
6	<code>sf_local_chk.sql</code>	Stub script for validating local configuration checks.
7	<code>sf_system_privs_chk.sql</code>	Validates required system privileges.
8	<code>sf_ldap_chk.sql</code>	Validates grants required when LDAP is in use.
9	<code>sf_pci_chk.sql</code>	Validates grants required when the PCI patch is applied and displays Baseline Credit Card Configuration values for review.

Execution of the Configuration Validation Process

When executing the Banner Flexible Registration Configuration Validation Process, ensure that the Oracle database is running. You may execute the script from the command line or from within a PL/SQL tool configured to execute SQL Plus from a command window.

Command line execution

To execute the script from the command line, perform the following steps:

1. Open a command window in the `/plus/` folder.
2. At the prompt, execute the following command:

```
sqlplus /nolog @sf_chk_config_driver.sql
```
3. When prompted, enter the password for the BANINST1 user and press enter. The driver script will execute the version check script specified within `sf_chk_config_driver.sql`.
4. After the script completes, enter the password for the FLEXREG user and press enter. The driver script will execute all of the remaining validation scripts specified within `sf_chk_config_driver.sql`.
5. Output from the script will display in the command window. The script will also spool the output to `sf_chk_config.LST` in the `/plus/` folder.
6. Review the output and make any necessary configuration changes.

Make enhancements to the Configuration Validation Process

When additional validation checks need to be added, those checks should be added to the `sf_local_chk.sql` script.

 **Note**

Mimic the existing patterns and message format used in the existing scripts. ■

 **Note**

Refer to [“Test script patterns” on page A-13](#) for a description of the patterns used to create the SQL test scripts. ■

To add additional validation checks, edit the `sf_local_chk.sql` script file (located in the `/plus/` folder) and save your changes.

Validation Process output

The `sf_chk_config_driver.LST` file will contain a section for each of the SQL scripts processed. Messages from each of the scripts will appear indented below the *Checking XXXX Configurations* header. The section will end with a *done* label. The driver script prints *DONE* when all scripts have been executed.

Messages are identified as either **WARNING** or **ERROR**.

If the existing set up might result in failure during deployment of the Banner Flexible Registration application, the message will be marked as an **ERROR**. If the data should be reviewed for completeness, it is marked as **WARNING**.

Note

All **ERROR** messages must be resolved before implementing Banner Flexible Registration.

Validation check errors

This section provides details about each of the testing scripts and an explanation for each of the warnings or errors that may be reported.

Global Configuration Error Messages (`sf_globl_chk.sql`)

The following errors may be detected by the `sf_globl_chk.sql` script:

Message	Severity	Description
More than one Global Configuration record exists.	ERROR	The SFRACNFG form supports only a single Global Configuration record. Multiples in the SFRRGLBL table are an error and this must be corrected before implementing Banner Flexible Registration.

Message	Severity	Description
Global Configuration Rules are missing.	ERROR	A Global Configuration record is required on the SFRACNFG form. If there is no record in the SFRRGLBL table, it must be corrected before implementing Banner Flexible Registration.
API generated errors.	ERROR	A variety of API generated errors may occur. These should only occur if the data was entered via SQL without the use of the API. These must be corrected before implementation.

Catalog Configuration Error Messages (sf_frc_chk.sql)

The following errors may be detected by the sf_frc_chk.sql script:

Message	Severity	Description
PARAMETER_CODE - still contains [*CHANGE*] default value.	WARNING	Alert that some rules have not been defined. Note: The Banner Flexible Registration application can be implemented when some of the optional parameters are left as *CHANGE*. Please see “Create Banner Flexible Registration catalogs on Flexible Registration Catalog Validation Form (SFTVCTLG)” on page 4-4 for more information.
Rules do not exist for this catalog.	WARNING	All catalogs set up on the SFRACCTLG form are checked by this script. Some may not be catalogs which will be implemented or catalogs with an End Date that has passed. If the catalog will be made available for Banner Flexible Registration, this should be corrected on the SFRACCTLG form.
API generated errors.	ERROR	A variety of API generated errors may occur. These should only occur if the data was entered via SQL without the use of the API. These must be corrected before implementation.

Message	Severity	Description
Payment Processor must be N when PPC patch NOT applied.	ERROR	If the Banner General 8.2.1.1 release or the Banner General 8.3 release have not been applied, the Payment Processor Connection payment type is not supported. This check searches for the posting in the GURPOST table of p1-639eqc_gen80200 or GURVERS of '8.3' and displays this error if the GURPOST record does not exist and the PPC_PAYMENT rule is set to Y for any catalog.
Internal Payment is no longer supported.	WARNING	As of Banner Flexible Registration 8.3.2, support for the Java Payment Client has been removed and the INTERNAL_PAYMENT rule has been deleted. If you have the rule set to Y, this warning will be displayed if the credit card package (SFOKCCRD) exists and is Valid on the database.
Internal Payment is no longer supported. Remove INTERNAL_PAYMENT rule.	ERROR	As of Banner Flexible Registration 8.3.2, support for the Java Payment Client has been removed and the INTERNAL_PAYMENT rule has been deleted. If you have the rule set to Y, this warning will be displayed if the credit card package (SFOKCCRD) exists but is NOT Valid on the database.
Internal Payment is no longer supported.	WARNING	As of Banner Flexible Registration 8.3.2, support for the Java Payment Client has been removed and the INTERNAL_PAYMENT rule has been deleted. If you have the rule set to N, this warning will be displayed regardless of the state of the credit card package (SFOKCCRD) on the database.

Message	Severity	Description
XXXXX rule has not been configured for catalog.	WARNING	When a catalog contains at least one rule on SFRACCTLG, all other parameters in SFTVCPRM will be checked. If the parameters do not exist in the SFRRCTLG table for this catalog, and are not System Required, a warning message will be displayed. This should be reviewed before the catalog is used in Banner Flexible Registration.
Unable to determine if all SFTVCPRM rules configured.	ERROR	A SQL failure in executing the check against SFTVCPRM and the sf_ctlg_configuration API. Database packages should be checked for validity.

GTVSDAX rules (sf_sdax_chk.sql)

The following errors may be detected by the sf_sdax_chk.sql script:

Message	Severity	Description
WEBDROPLST Web Drop Last Class valid values Y or N.	WARNING	The Web Drop Last Class value in GTVSDAX is not set to Y or N. This should be corrected.
WEBMANCONT Web Use Management Controls valid values Y or N	WARNING	The Web Use Management Controls value in GTVSDAX is not set to Y or N. This should be corrected.
WEBRESTTKT WebVR Restrict Reg to Time Tkt valid values Y or N.	WARNING	The Web Voice Response Restrict Registration to Time Ticket value in GTVSDAX is not set to Y or N. This should be corrected.
AUTODROP Drop Connected Courses valid values C, Y, or N.	WARNING	The Drop Connected Courses value in GTVSDAX is not set to C, Y, or N. This should be corrected.
ADMINDROP Drop Courses in Admin Error valid values ' or N.	WARNING	The Drop Courses in Admin Error value in GTVSDAX is not set to Y or N. This should be corrected.
REGACCTIME Reg Access Time Out must be a number.	WARNING	The Registration Access Time Out value in GTVSDAX must be set to a number. This should be corrected.

Message	Severity	Description
WEBRSTSDRP default drop code not found in GTVSDAX.	ERROR	The default drop code in GTVSDAX was not found. This should be corrected.
WEBRSTSDRP default drop code not valid. Default drop code not found in STVRSTS.	ERROR	The default drop code in GTVSDAX is not valid. This should be corrected.

Term Rules (sf_term_chk.sql)

The following errors may be detected by the `sf_term_chk.sql` script:

Message	Severity	Description
SOATERM record does not exist.	ERROR	SOATERM must be set up for all terms identified on SFRACTLG for each catalog.
SFAESTS record does not exist.	ERROR	SFAESTS must be set up for all terms identified on SFRACTLG for each catalog.
SFARSTS record does not exist.	ERROR	SFARSTS must be set up for all terms identified on SFRACTLG for each catalog.
SOAORUL record does not exist.	ERROR	SOAORUL must be set up for all terms identified on SFRACTLG for each catalog. Note: This only needs to be configured when open learning is in use.
SORRSTS record does not exist.	ERROR	SORRSTS must be set up for all terms identified on SFRACTLG for each catalog. Note: This only needs to be configured when open learning is in use.
SFARSTS record does not contain RW.	ERROR	The SFRRSTS table does not contain the registration code identified for the catalog and term on SFRACTLG. In this case, it is RW.

Message	Severity	Description
SORRSTS record does not contain RW.	ERROR	The SORRSTS table does not contain the registration code identified for the catalog and term on SFRACCTLG. In this case, it is RW. Note: This only needs to be configured when open learning is in use.
Catalog Configuration rule for Registration Code missing.	ERROR	When there is no value for the Registration Code on SFRACCTLG, the term check will display this error since it cannot validate the registration code with the SFRRSTS and SORRSTS tables.

Profile Rules (sf_profile_chk.sql)

The following errors may be detected by the sf_profile_chk.sql script:

Message	Severity	Description
SFRAPROF Profile record does not exist.	ERROR	SFRAPROF must be set up for each catalog.
GUAPPRF Pin creation rules do not exist.	ERROR	GUAPPRF Pin creation rules must be set up.

Local Check (sf_local_chk.sql)

The following errors may be detected by the sf_local_chk.sql script:

Message	Severity	Description
Unhandled Exception raised by local configuration check. SQLCODE:SQLERRM.	ERROR	Error executing logic. The SQL script should be reviewed to ensure it is correct and appropriate error trapping occurs. The SQLCODE and SQLERRM will be appended to the end of the message.

System Privileges Check (sf_system_privs_chk.sql)

The following errors may be detected by the sf_system_privs_chk.sql script:

Message	Severity	Description
Missing SYSTEM privilege for FLEXREG on CREATE ANY SYNONYM.	ERROR	<p>The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following:</p> <pre>GRANT CREATE ANY SYNONYM TO FLEXREG;</pre>
Missing SYSTEM privilege for FLEXREG on CREATE JOB.	ERROR	<p>The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following:</p> <pre>GRANT CREATE JOB TO FLEXREG;</pre> <p>Note: As of Banner Flexible Registration 3.0, this is no longer used and can be ignored.</p>
Missing SYSTEM privilege for FLEXREG on CREATE MATERIALIZED VIEW.	ERROR	<p>The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following:</p> <pre>GRANT CREATE MATERIALIZED VIEW TO FLEXREG;</pre> <p>Note: As of Banner Flexible Registration 3.0, this is no longer used and can be ignored.</p>
Missing SYSTEM privilege for FLEXREG on CREATE PROCEDURE.	ERROR	<p>The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following:</p> <pre>GRANT CREATE PROCEDURE TO FLEXREG;</pre>
Missing SYSTEM privilege for FLEXREG on CREATE PUBLIC SYNONYM.	ERROR	<p>The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following:</p> <pre>GRANT CREATE PUBLIC SYNONYM TO FLEXREG;</pre>

Message	Severity	Description
Missing SYSTEM privilege for FLEXREG on CREATE SEQUENCE.	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT CREATE SEQUENCE TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on CREATE SESSION	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT CREATE SESSION TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on CREATE SYNONYM	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT CREATE SYNONYM TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on CREATE TABLE	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT CREATE TABLE TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on CREATE TRIGGER	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT CREATE TRIGGER TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on CREATE VIEW	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT CREATE VIEW TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on DEBUG CONNECT SESSION	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT DEBUG CONNECT SESSION TO FLEXREG;

Message	Severity	Description
Missing SYSTEM privilege for FLEXREG on DROP PUBLIC SYNONYM	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT DROP PUBLIC SYNONYM TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on EXEMPT ACCESS POLICY	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT EXEMPT ACCESS POLICY TO FLEXREG;
Missing SYSTEM privilege for FLEXREG on UNLIMITED TABLESPACE	ERROR	The FLEXREG user must have this system privilege but does not. To correct this error, via SQL*PLUS as the SYSTEM user perform the following: GRANT UNLIMITED TABLESPACE TO FLEXREG;

LDAP Check (sf_ldap_chk.sql)

The following errors may be detected by the sf_ldap_chk.sql script:

Message	Severity	Description
Grant to KEY_DIR does not exist.	ERROR	LDAP is configured according to twbklibs.twgbldap_rec.twgbldap_prot ocol but the required KEY_DIR grant is missing.

PCI Check (sf_pci_chk.sql)

The following errors may be detected by the sf_pci_chk.sql script:

Message	Severity	Description
Missing grant for FLEXREG on GOKPURL.	ERROR	When the Baseline PPC patch has been applied to the database, grants must be set up for FLEXREG and FLEXREG_USER to access the new Baseline dbprocs.
Unable to check for grant for FLEXREG on GOKPURL.	ERROR	Unable to determine if the grants for PPC patch have been created for Banner Flexible Registration to work with Credit Card payments.
Unable to determine if Payment Processor Connection Patch applied.	ERROR	There was a failure in selecting from GURPOST to determine if the Banner General 8.2.1.1 patch or Banner General 8.3 was installed for supporting Payment Processor Connection.
External Merchant ID is not defined on GTVCCRD for VISA.	REVIEW	Baseline definition of Credit Cards must be established with an external merchant ID as provided by merchant.
Confirm that External Merchant ID defined as Visa for VISA is correct.	REVIEW	Review baseline definition of Credit Card to ensure that the external merchant ID is correct.
Missing GTVPROC code for Flexible Registration - FLEXREGCCREGFEES.	ERROR	This code should have been inserted during the Banner Flexible Registration installation. If missing, please add it.
Credit Cards have not been entered on GOAMERC for FLEXREGCCREGFEES.	ERROR	Credit Cards which will be permitted through Banner Flexible Registration must be defined on the GOAMERC form.
Confirm that VISA credit card set up for 0 Merchant ID is correct.	REVIEW	Review the Credit Card crosswalk on GOAMERC to make sure that all credit cards which will be accepted through Banner Flexible Registration have been defined. The Merchant ID is associated with the GTVSDAX settings. Check WEBSTUCCID for default value.
Confirm that Default Merchant ID defined as 0 on GTVSDAX is correct.	REVIEW	Review the GTVSDAX settings for WEBSTUCCID. Refer to Baseline documentation on required set up.

Message	Severity	Description
Confirm that Default Term defined as 202020 on GTVSDAX is correct.	REVIEW	Review the GTVSDAX settings for WEBCCDEFTERM. Refer to Baseline documentation on required set up.
PAYVEND_URL must be defined on TWGBPARM with the Vendor URL.	ERROR	PPC payments will only work if the Vendor URL is correctly defined in Web Tailor. Refer to Baseline set up requirements for PPC.
Confirm that http://... is the correct Vendor URL.	REVIEW	PPC payments will only work if the Vendor URL is correctly defined in Web Tailor. Refer to Baseline set up requirements for PPC.

Test script patterns

This section recommends specific patterns to follow for the testing scripts.

Wrap all tests with exception handling

Each test must be wrapped in a BEGIN-END-EXCEPTION block. This method ensures that all validations are executed even if a particular test results in an unexpected SQL exception. For example:

```
--Check the Reg Access Time Out. Valid value must be a number.
--Wrap this in a block to catch exception on the TO_NUMBER.
DECLARE
lv_regacctime NUMBER;
BEGIN
SELECT TO_NUMBER(f_get_sdax('REGACCTIME','REGISTRATION'))
INTO lv_regacctime
FROM DUAL;
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line(' 'REGACCTIME' Reg Access Time Out must
be a number.'||' *WARNING*');
END;
```

Leverage Banner Validation API calls

Tests that validate a particular form should leverage the Banner Validation APIs. For example:

```

--Wrap the API validation in a block to catch exceptions
DECLARE
  lv_errmsg VARCHAR2(10000);
  lv_errline VARCHAR2(10000);
  lv_pos_start NUMBER;
  lv_pos_end NUMBER;

BEGIN
  sfb_ctlg_configuration_rules.p_validate(
    p_ctlg_code => lv_ctlg_rec.r_ctlg_code,
    p_cprm_code => lv_ctlg_rec.r_cprm_code,
    p_value => lv_ctlg_rec.r_value);

EXCEPTION WHEN OTHERS THEN
  --The API error messages come back concatenated together with
  ::
  --Break this up so each appears on a separate line.
  lv_errmsg := SQLERRM;
  lv_errline := '';
  lv_pos_start := 0;
  lv_pos_end := 0;
  --loop through the SQLERRM and break off messages based on ::
  WHILE (LENGTH(lv_errmsg) > 0 ) LOOP
    lv_pos_start := NVL(INSTR(lv_errmsg,'::'),1);
    --if not found, just take first position in string
    lv_errmsg := SUBSTR(lv_errmsg,lv_pos_start + 2,
LENGTH(lv_errmsg));
    --break off the Oracle error # and ::
    lv_pos_end := NVL(INSTR(lv_errmsg,'::') - 1
,LENGTH(lv_errmsg));
    --if not found, just take entire string
    lv_errline := SUBSTR(lv_errmsg,1,lv_pos_end);
    lv_errmsg := LTRIM(RTRIM(SUBSTR(lv_errmsg,lv_pos_end + 3,
LENGTH(lv_errmsg)), ' '), ' ');
    IF (lv_errline IS NOT NULL) THEN
      dbms_output.put_line(' ' ||lv_errline||' *ERROR*');
    END IF;
  END LOOP;

END;

```

B Banner Flexible Registration Web Application Services Configuration

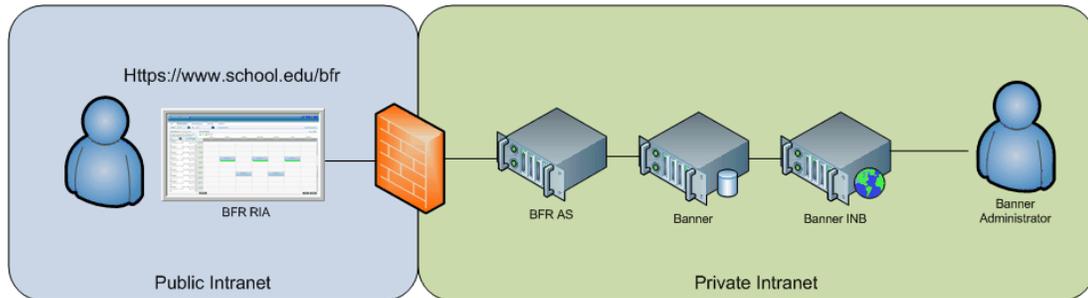


This appendix provides information on diagnosing and resolving connectivity issues for Banner Flexible Registration. It describes the interaction between the application user interface and the web services, as well as the dependencies that exist. It discusses how to validate a deployment, configure the application, and modify the configuration. It also provides technical details regarding the interaction between the user interface and the middle tier components. This information is intended for technical users who are responsible for the deployment and configuration of Banner Flexible Registration.

Deploy application



Here is an example of a simple deployment of the application into an institution's environment.



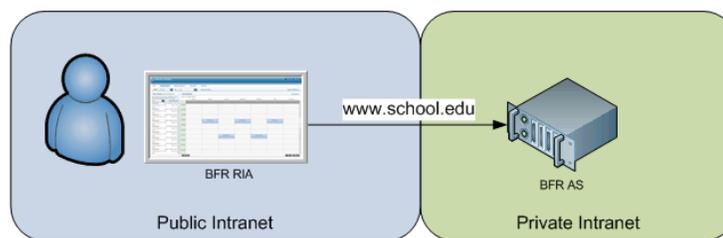
The key components are as follows:

Component	Description
 <p data-bbox="451 428 526 447">BFR RIA</p>	<p data-bbox="657 333 1333 430">This is the Banner Flexible Registration Rich Internet Application (BFR RIA). It has a robust user interface that allows users to interact with a catalog of class sections.</p>
 <p data-bbox="451 646 519 667">BFR AS</p>	<p data-bbox="657 543 1373 674">This is the Application Server on which Banner Flexible Registration is installed (BFR AS). It provides the resources to support the BFR RIA, and it contains the services which are consumed by the BFR RIA.</p> <p data-bbox="657 701 1365 798">The services use the Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) and are specific to BFR AS.</p>
 <p data-bbox="451 970 519 991">Banner</p>	<p data-bbox="657 867 1373 997">This is the existing Banner System. Banner Flexible Registration is tightly integrated with Banner and uses much of the baseline functionality including, but not limited to, new student admissions, section registration, and payment.</p>
 <p data-bbox="451 1188 519 1209">Banner INB</p>	<p data-bbox="657 1085 1373 1182">This is the Internet-native Banner (INB) interface. This allows the Banner administrators to manage Banner at their institutions.</p>
 <p data-bbox="451 1421 519 1442">Banner Administrator</p>	<p data-bbox="657 1297 1373 1428">This is the Banner administrator. This represents the individuals responsible for managing Banner Flexible Registration through the BFR extensions to the Internet-native Banner (INB) interface.</p>

 **Note**

The Banner System, Internet-native Banner interface, and Banner administrator elements are listed as components to provide a more complete picture of the elements included in a typical implementation of Banner Flexible Registration. ■

Here is an example of the interaction between the BFR RIA and the BFR AS. In this example, the user must access Banner Flexible Registration installed on the BFR AS machine at <https://www.ellucianuniversity.edu/bfr>.

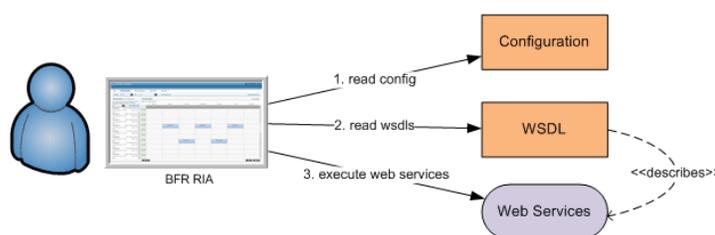


Access web services

The BFR RIA is the front end that interacts with the BFR AS, the application server that hosts the Banner Flexible Registration web services. The Banner Flexible Registration application is run within the browser. This is different from traditional web sites where a new page is loaded in response to a user request. The BFR RIA generates requests to activate services that are provided by the BFR AS. Services perform tasks such as authenticating user credentials, searching for sections that meet user criteria, registering users for sections, and reviewing existing registration records.

Banner Flexible Registration uses SOAP over HTTP functionality to communicate with the BFR AS. The BFR RIA must know how to locate and access the services and then download the appropriate Web Service Definition Language (WSDL) files. The WSDL files are .xml files that contain information about the web services, including the Uniform Resource Identifier (URI) used to access the services. Once the application has located the URI for the WSDL files, the files can be downloaded.

The diagram below illustrates these processing steps in order.



Read WSDL files

The WSDL files describe the contract between a web services provider and web services customers. This includes a detailed description of the contents of a request and a response, as well as where the services are located. Banner Flexible Registration must download these WSDL files when the application is launched.

Typically, web services are available through the same URI that exposes the RIA, such as `http://www.ellucianuniversity.edu/bfr`. (An example of a service that could be exposed is a security service such as the login process.) The application loads the WSDL by checking the defined service element value and requesting the `ws-security.wsdl` resource. To continue with the example from the diagram, the URI used to access the `ws-security.wsdl` resource would be:

```
http://www.ellucianuniversity.edu/bfr/ws-security.wsdl
```

Here are the results of the query.

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<definitions
targetNamespace="urn:hedtech:flexreg:ws:security:v1.0"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="urn:hedtech:flexreg:ws:security:v1.0"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:schema="urn:hedtech:flexreg:ws:authentication:v1.0">
<types>
  <xs:schema
xmlns="urn:hedtech:flexreg:ws:authentication:v1.0"
elementFormDefault="qualified"
targetNamespace="urn:hedtech:flexreg:ws:authentication:v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="LoginRequest">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="username" type="xs:string"/>
          <xs:element name="password" type="xs:string"/>
          <xs:element name="token" type="TokenType"/>
          <xs:element name="catalogId" minOccurs="1"
maxOccurs="1" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="LoginResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="resultCode" type="xs:string"/>
          <xs:element name="token" type="TokenType"/>
          <xs:element name="userFirstName"
type="xs:string"/>
          <xs:element name="userLastName"
type="xs:string"/>
          <xs:element name="username" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</types>
</definitions>
```

```

        <xs:element name="authenticated"
type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="TokenType">
    <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="CaptchaType">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="id"
type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="code"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="LoginRequest">
    <part name="LoginRequest" element="schema:LoginRequest"/>
</message>
<message name="LoginResponse">
    <part name="LoginResponse" element="schema:LoginResponse"/>
</message>
<portType name="security">
    <operation name="Login">
        <input name="LoginRequest" message="tns:LoginRequest"/>
        <output name="LoginResponse" message="tns:LoginResponse"/
>
    </operation>
</portType>
<binding name="securityBinding" type="tns:security">
    <soap:binding style="document" transport="http://
schemas.xmlsoap.org/soap/http"/>
    <operation name="Login">
        <soap:operation soapAction=""/>
        <input name="LoginRequest">
            <soap:body use="literal"/>
        </input>
        <output name="LoginResponse">
            <soap:body use="literal"/>
        </output>
    </operation>

```

```

</binding>
<service name="securityService">
  <port name="securityPort" binding="tns:securityBinding">
    <soap:address location="ws-security/" />
  </port>
</service>
</definitions>

```

 **Note**

The location (URI) of the service is included at the end of the WSDL file.

```

<service name="securityService">
  <port name="securityPort" binding="tns:securityBinding">
    <soap:address location="ws-security/" />
  </port>
</service>

```

Locate and request URI

The location value is generated at runtime and is based on the requested URI. This means that a specific application service can be addressed using more than one name.

For example, a sample application server is accessible from the Internet using the URI `http://www.ellucianuniversity.edu/` and could be accessed from the institution's intranet using the URI `http://as1.ellucianuniversity.edu:7777/bfr` and `http://123.45.67.89:7777/bfr`. In both cases, the value of the location attribute in a WSDL file reflects the URI of the request.

Here are some examples.

Request URI	soap:address\location Value
<code>http://www.ellucianuniversity.edu/bfr/ws-security</code>	<code>http://www.ellucianuniversity.edu/bfr/ws-security</code>
<code>http://as1.ellucianuniversity.edu:7777/bfr/ws-security</code>	<code>http://as1.ellucianuniversity.edu:7777/bfr/ws-security</code>
<code>http://123.45.67.89:7777/bfr/ws-security</code>	<code>http://123.45.67.89:7777/bfr/ws-security</code>

Modify generated addresses

When an institution does not want Banner Flexible Registration to create a URI based on the request, it can instead choose to restrict the WSDL files to a specific URI.

Oracle OAS	Description
------------	-------------

Oracle 11g	This process requires the modification of deployed files. With Oracle 11g, you can:
------------	---

- deploy the .ear file by mounting the Enterprise Archive (EAR) directly, or
- deploy the .ear file by first exploding the EAR into a directory structure and then mounting that directory structure.

If your institution is mounting the EAR directly, you must make the modifications and include those changes in a new .ear file, and then deploy the EAR.

If your institution explodes the EAR first, you can modify the files directly.

An institution could choose to restrict the WSDL files to a particular URI for the following reasons:

- to limit web service access to a single URI
- to not use an incorrect URI returned in the WSDL file
- to use a single URI to service all web service requests

Use these steps to restrict the WSDL files to a selected URI. Refer to the details that follow these summary steps for more information.

1. Specify the web server location of the URIs.
2. Restart the application server.

Step 1 Specify web server location of URIs

The URI for each web service is defined in the `WEB-INF\ spring-ws-servlet.xml` file. Five services must be modified to specify the URI web server location.

- `ws-search`
- `ws-domain`
- `ws-profile`
- `ws-payment`
- `ws-security`

Use the following steps to specify the web server location of the URIs.

1. Open the `spring-ws-servlet.xml` file.
2. Edit the file using a text editor, such as Notepad, Notepad++, TextPad, or UltraEdit.

Do not use Microsoft Word, Open Office Writer, or any other word processing software that adds formatting characters to the file.

3. Search for `id="ws-search"` to locate the definition of the `ws-search` service.
4. Modify the value attribute of the `locationUri` property to include the complete URI.

Here is an example of the required change.

```
<bean id="ws-search" class="org.springframework.ws.wsdl.
wsdl11.DynamicWsdl11Definition">
  <property name="builder">
    <bean class="org.springframework.ws.wsdl.wsdl11.
builder.XsdBasedSoap11Wsdl4jDefinitionBuilder">
      <property name="schema" value="/search.xsd" />
      <property name="portTypeName" value="search" />
      <property name="locationUri" value="ws-search/" />
      <property name="locationUri" value="http://www.school.
edu/bfr/ws-search/" />
      <property name="targetNamespace" value="urn:hedtech:
flexreg:ws:search:v1.0" />
    </bean>
  </property>
</bean>
```

5. Repeat the process for the remaining four services.
 - `ws-domain`
 - `ws-profile`
 - `ws-payment`
 - `ws-security`

6. Save the changes.
7. Close the file.

Step 2 Restart application server

Once the changes have been made to disable the automatic generation of the location URIs and specify the web server location of the URIs, the Banner Flexible Registration application server must be restarted. If the application server is not restarted, unexpected and erratic processing behavior could result.

Validate the configuration

It is important to validate that the application and associated web services are configured correctly. This applies to both basic installations and installations where the configuration files have been modified.

The configuration can be validated each time the application is deployed or modified by connecting to the application and using all of the expected addresses. Access the application on the application server using the public-facing URI and any internal URIs (i.e., directly against the machine's name).

Validate web service location URI

Use these steps to validate the web service location URI.

1. Request each web service WSDL file directly.
2. Check that the value of the `soap:address/location` attribute is correct.

Here is an example of the services, associated WSDL files, and the URIs used to access the WSDL files.

Web Service	WSDL File	Example URI
ws-search	/ws-search.wsdl	ws-search.wsdl
ws-domain	/ws-domain.wsdl	ws-domain.wsdl
ws-profile	/ws-profile.wsdl	ws-profile.wsdl
ws-payment	/ws-payment.wsdl	ws-ws-payment.wsdl
ws-security	/ws-security.wsdl	ws-security.wsdl

Refer to the “Read WSDL files” topic for a sample file and the position of the location URI value in the file. If the `spring-ws-context.xml` file has been modified per the instructions found in the “Modify generated addresses” topic, you will see the URIs that you provided instead of the values listed in the Example URI column above.

Troubleshooting

Listed below are some frequently asked questions and the associated answers and tips.

Can I use a load-balanced environment?

You can install Banner Flexible Registration in a load-balanced environment with minimal configuration. However, clients using the Payment Provider Connector (PPC) to process payments must ensure that the PPC success and failure URLs are configured through the Flexible Registration Catalog Rules Form (SFRACTLG) with the full URL to the front end web server or load balancing server.

Can I access the application through the internet and the intranet?

Assuming there are no network devices interfering with or manipulating the HTTP requests and responses, you will be able to access Banner Flexible Registration from the internet and the intranet without modifying the configuration files. However, it is possible that a device, such as a firewall or load balancer, may modify the HTTP requests.

C Banner Flexible Registration EhCache Configuration

This appendix provides information on EhCache configuration.

EhCache configuration

EhCache provides a middle-tier caching of specific database objects to enable high-performance retrieval of common data that is infrequently modified in the database. Once the data is cached on the application server, subsequent requests for that data are provided from the cache instead of actually retrieving the data from the database. The caches for each object periodically expire and are refreshed based upon configuration elements that can be customized to suit the needs of your institution.

The deployed EhCache configuration file provides for very fast cache expirations so that changes to cached data will be frequently refreshed from the database. This helps with testing of the deployed application so that you can make changes to data that is cached and have it become quickly available to the application without having to wait long periods of time for the cached data to expire.

EhCache is deployed in the following location:

```
%ORA_HOME%\j2ee\home\applications\%DEPLOYED_NAME%\BannerFlexibleRegistrationxxxxx\WEB-INF\classes\ehcache.xml
```

ehcache.xml

The cache configuration file is organized into three sections of settings for various caches.

Section 1

This section contains mandatory default EhCache specific settings and does not need to be modified.

Section 2

This section contains Flexible Registration specific caches related to basic domain objects and domain collection objects that are infrequently modified and used to drive drop down list boxes in the user interface. These cache settings have been configured specifically to

help test the application and will not provide good caching performance for a deployed production application. Once the application has been deployed and tested, your institution should modify the settings in this section to provide good caching performance balanced with the need for appropriate cache expiration values.

All of the caches in this section have been configured with a `TIMETOLIVESECONDS="20"` setting. This setting defines the maximum amount of time between creation time and when an element expires. Once the system has been tested and deployed, the expiration setting for these caches should be updated to at least a five minute expiration value of `TIMETOLIVESECONDS="300"`.

 **Note**

If you don't expect the values that comprise these domain objects to change very often, setting a longer `TIMETOLIVESECONDS` value should be considered to improve caching performance. ■

Section 3

This section contains Flexible Registration specific caches for the Captcha validation process and registration cart processing.

 **Warning**

These caches should not be modified. ■

 **Note**

Any changes to the `ehcache.xml` file will require a restart of the Banner Flexible Registration application before those changes will be used by the application. ■

D Banner Flexible Registration Captcha Configuration

This appendix provides information on captcha configuration.

Captcha configuration

Captcha stands for Completely Automated Public Turing Test To Tell Computers and Humans Apart. The Captcha component provides a challenge-response test in the Banner Flexible Registration application to ensure that new user profiles are being created by a human being rather than by an automated process. The Captcha component accomplishes this by generating a JPEG image which contains a random series of alpha-numeric characters which are visually distorted but still recognizable to humans. The image is displayed during the profile creation process and requires the user to enter the same characters which are present in the image to submit the request.

Captcha is deployed to the following configuration file:

```
captcha-applicationcontext.xml
```

Troubleshooting

Captcha display issues on Linux

In some cases the Captcha component is unable to render an image when running under a Linux operating system that does not have an installed Sun Java JDK. To prevent this issue from occurring, perform the following step:

Execute the following command from the console:

```
java -version
```

If the version mentions GNU, then the Captcha component will not produce an image.

To resolve this problem, simply download and install a Sun JDK and make sure that its bin directory comes before the GNU java in the path.

Note

This is likely to occur with Ubuntu and OpenSUSE, and may occur with other distributions.



E Banner Flexible Registration Log Files

This appendix provides information on the logs created by Banner Flexible Registration web application.

The Banner Flexible Registration web application logs events that occur during the processing of requests from the Banner Flexible Registration web application.

This log contains invaluable information that can aid with resolving issues.

Default logging

By default, Banner Flexible Registration writes to application server's standard output.

The location of server output may differ based on your particular Oracle Application Server configuration.

In WebLogic 11g, the likely locations include the following:

- `<<weblogic_domain>>/servers/<<server_name>>/logs/<<server_name.out>>`
- `<<weblogic_domain>>/servers/<<server_name>>/logs/<<server_name.log>>`
- No log file created. All events are being written directly to the console.

Modifying the location of the log files

Banner Flexible Registration provides an option to modify the default logging behavior and write to log files at custom locations. The following details describe the steps to customize logging behavior.

1. Download the BannerFlexibleRegistration-`<<version>>`-resources.zip contains all of the resources for Banner Flexible Registration.
2. Copy the customized log configuration file from `config/log4j.properties` into `BFR_HOME% config`.
3. Your institution can modify where the log files are created by modifying the copied `log4j.properties` file. If unmodified, setting up this external `log4j.properties` in `BFR_HOME` allows log files being written to `%BFR_HOME%/log`.

By default, the names of the log files are as follows:

- FlexReg_Services.log (Services log file) - The Services log file is the main log file and contains information about the interaction between the Banner Flexible Registration web application and the Application Server.
- FlexReg_Expiration_Services.log (Expiration log file.) - The Expiration log file contains detailed information about the cart expiration process. (For more information on the cart expiration process, please see [“Cart expiration process” on page 3-3](#).)
- FlexReg_MessageSender_Services.log (E-mail log file) - The E-mail log file contains detailed information about the e-mail process that sends e-mails as part of the registration check out flow.

Identifying issues using the log files

The type of issues most likely to occur are database related issues when the catalog has not been correctly configured. All log messages are displayed in the following format:

```
[ %sequence% -- %userid% -- %pidm% -- %ipaddr%] %date% %time%
%level% %message%.]
```

%sequence%

This element of the log line is a unique sequence number that can be used to group all log entries created during the processing of a single request.

Note

Multiple requests from the same user will have different %sequence% values. ■

%userid%

This field will either be blank, contain the word ANONYMOUS or will contain the Banner ID of the student.

If the value is blank, it means that the log line is for a system process. When the application first starts, there is the potential for many log entries to be created that are related to the logging application start up. These lines will not have a value in the %userid% field.

If the entry was created while processing a request from a user, then the entry will be ANONYMOUS or the Banner ID (example: jdoe1234).

%pidm%

If the %userid% contains a Banner ID, then %pidm% will contain the PIDM for that student.

%ipaddr%

`%ipaddr%` contains the IP address of the user that initiated the request.

%date% and %time%

These fields capture the system date and time at the time the log entry was created. The date and time are based on the local system time for the application server.

%level%

`%level%` captures the level of the log. Valid values include the following:

- FATAL
- ERROR
- WARN
- INFO
- DEBUG

These values correspond to the log levels defined by the Log4J logging package.

%message%

`%message%` contains the details of the log entry. If the log file has INFO and DEBUG logging enabled, then the message will show detail about all of the components involved in the processing of a request. ERROR and FATAL will often contain a lot of detailed information about the nature of the problem.

For example, the message below is an example of the type of error that one might see in the log. This error was created by removing the EXECUTE grant to the package FLEXREG:SFSKSRCH to the FLEXREG_USER_ROLE, to which the FLEXREG_USER belongs.

```
[54470867605 -- A00013292 -- 37882 -- 149.24.111.98] 2009-10-23
12:05:24,671
ERROR [net.hedtech.flexreg.ws.SearchEndPoint] - Exception
occurred during basic search. Exception message:
CallableStatementCallback; bad SQL grammar [{call
SFSKSRCH.P_SEARCH_SECTIONS(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?)}]; nested exception is
java.sql.SQLException: ORA-06550: line 1, column 7:
PLS-00201: identifier 'SFSKSRCH' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
```

From this example, we can see that a database error (ORA-06550) occurred when a registered and logged in user (A00013292, pidm=37882) executed a search (SearchEndPoint).

The message of the log message includes details created by the Oracle Database server. While the message does not explicitly say that the root cause is a missing grant, it does provide sufficient information to begin searching for a solution.

Logging Levels and Personally Identifiable Information

This section provides details about the supported logging levels and about how personally identifiable information is handled.

Logging levels

The `log4j.properties` file is broken down by package structure. A given package is assigned its own logging level. These packages can have a parent/child relationship with one another. In such a case, the child package logging level will override the parent package logging level. The following are examples of packages in the `log4j.properties` file that illustrate this:

- `log4j.logger.net.hedtech=WARN, mainAppLogAppender`
- `log4j.logger.net.hedtech.framework=DEBUG, mainAppLogAppender`

The logging entries listed show that the parent package of `net.hedtech` is set to logging level `WARN`. It has a child package, `net.hedtech.framework` which is set to `DEBUG`. This means that any logging done in any classes of the child package will be done at the `DEBUG` level or higher. Any logging done in any classes within the parent package will only be done at the `WARN` level or higher (excluding `DEBUG`, which is a lower logging level than `WARN`). The logging levels work such that information for the given package is logged at the specified level plus any higher levels as well. Therefore, for package "X" with a logging level of `DEBUG`, information will be logged in that package at the `DEBUG` level and higher, including `WARN`, `ERROR`, and `FATAL`. For package "Y" with a logging level of `WARN`, information will be logged at the `WARN` level and higher, including `ERROR` and `FATAL`.

The logging levels govern what is logged to the log file. The logging levels are as follows:

- `FATAL`
- `ERROR`
- `WARN`
- `INFO`
- `DEBUG`

FATAL is the most restrictive logging level. At this level, only FATAL messages are logged. DEBUG is the least restrictive. At this level, DEBUG, INFO, WARN, ERROR, and FATAL messages (DEBUG and all higher-level messages) are logged.

Other logging levels may be specified. Please refer to the log4j documentation for more information.

Personally Identifiable Information

The logging levels in the `log4j.properties` file in Banner Flexible Registration are set in such a way to avoid Personally Identifiable Information (PII) from being logged in the log files. However, these logging levels can be changed to produce more data in the log files about the application. This may be necessary for debugging purposes.



Warning

If the logging levels for packages in the `log4j.properties` file are changed, personally identifiable information may be logged to the log files. ■

Reporting issues to Ellucian

When reporting a bug, providing the results of the DB Configuration Check Tool and log files will help analyze the issue. For more information, please see [Appendix A, “Banner Flexible Registration Configuration Validation”](#).

The logs files may contain personally identifiable information. If you provide the log file, you must take appropriate measures to secure the content. Using a compression tool that includes support for strong encryption of the content is highly recommended. For improved security, do not provide the password with the encrypted file. If sending over e-mail, you may send the file and password in separate e-mails. Another option is to include the encrypted log files with your service request and request ActionLine to contact an individual at the institution for the password.