ellucian™

Banner
Payment Processor Connection Handbook

December 2011

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

Revision History

Publication Date	Summary
December 2011	New version that supports Payment Processor Connection for Banner software.

Banner
Payment Processor Connection Handbook

Contents

Chapter 1 Overview 1-1

PCI compliance 1-1

Supported vendors 1-2

Processing flow 1-2

Data exchange 1-4

Banner dependencies 1-5

Contents of this handbook 1-5

Chapter 2 PaymentTransaction Web Service 2-1

What is a Web service? 2-1

What is the Banner Payment Transaction Service Adapter? 2-1

Requirements 2-2

Oracle application server and Java 2-2

Oracle database 2-3

Banner Translation Service 2-3

Installation on Oracle Application Server 10.1.3.4/5 2-3

Step 1 Create an OC4J instance 2-4

Step 2 Install the adapter 2-7

Step 3 Define the adapter data source 2-10

Step 4 Define the Translation Service data source 2-15

Step 5 Install the adapter 2-17

Step 6 Configure the security role and user 2-21

Step 7 Enable schema validation (optional) 2-26

Step 8 Configure logging 2-27
Step 9 Verify the deployment 2-28

Installation on Oracle WebLogic Server 11g 2-29

Recommended configuration 2-30
Installation steps 2-30
Step 1 Configure logging (optional) 2-30
Step 2 Define the adapter data source 2-32
Step 3 Define the Banner Translation Service data source 2-39
Step 4 Install the adapter 2-41
Step 5 Configure the security group and user 2-48
Step 6 Enable schema validation (optional) 2-53
Step 7 Verify the deployment 2-54

Web service operations 2-56

AddAccountTransaction operation 2-57
ReportTransactionError operation 2-58

SOAP fault messages 2-59

Message mapping to Banner 2-59

AddAccountTransaction 2-59
ConfirmAddAccountTransaction 2-60
ReportTransactionError 2-60
ConfirmReportTransactionError 2-60

Setup requirements 2-61

Payment card types 2-61
Accounting information 2-61

Chapter 3 Common Implementation 3-1

Implementation steps 3-1

Step 1 Define processes that use payment card processing 3-2
Step 2 Define source code 3-3
Step 3 Define printer code 3-3
Step 4 Build address hierarchy 3-3
Step 5 Define default merchant IDs 3-4
Step 6 Enable multiple merchant ID option 3-5

Step 7 Build multiple merchant ID hierarchy 3-6
Step 8 Define payment card types 3-9
Step 9 Define accounting information 3-10
Step 10 Define Banner Web Tailor parameters 3-12
Step 11 Define payment transaction descriptions 3-12
Step 12 Provide success and failure URLs to payment processing vendor 3-13

Implementation examples 3-14

Objects used with payment card processing 3-15
Vendor Payment Transaction Audit Form (GOICCAU) 3-15
Object Library (GOQOLIB) 3-17
Tables 3-18
Packages 3-18
API 3-21

Chapter 4 Banner Student Implementation 4-1

Form setup for admissions applications 4-1
Application Fee Waiver Reason Validation Form (STVWAIV) 4-1
Web Application Customized Lists Form (SAAWADP) 4-2
Admissions Application Form (SAAADMS) 4-3
Electronic Application Submitted Form (SAAETBL) 4-5
Electronic Applicant Web Default Rules Form (SAAWADF) 4-6

Form setup for enrollment verification requests 4-7
Enrollment Verification Type Code Validation Form (STVEPRT) 4-7
Web Self Service Options Validation Form (STVWSSO) 4-8
Web Payment Options Validation Form (STVWPYO) 4-9
Enrollment Verification Request Rules Form (SFAEPRT) 4-10

Form setup for registration and student accounts 4-13

Form setup for transcript requests 4-13
Transcript Type Code Validation Form (STVTPRT) 4-13
Web Self Service Options Validation Form (STVWSSO) 4-14
Web Payment Options Validation Form (STVWPYO) 4-14

Transcript Type Rules Form (SHATPRT) 4-14
Web Transcript Request Rules Form (SHAWTRR) 4-16

Form setup for graduation applications 4-17
Web Payment Options Validation Form (STVWPYO) 4-18
Graduation Application Display Rule Code Validation Form (STVGADR) 4-18
Self-Service Graduation Application Display Rules Form (SHAGADR) 4-18

Chapter 5 Banner Student Self-Service Implementation 5-1

Processing features 5-1

Common implementation for Banner Student Self-Service 5-3
Step 1 Verify common implementation 5-3
Step 2 Define AR holds indicator 5-3
Step 3 Define default term codes 5-4

Implementation for admissions applications 5-5
Step 1 Verify form setup 5-5
Step 2 Customize procedure definitions in Banner Web Tailor 5-5
Step 3 Customize information text in Banner Web Tailor 5-7

Implementation for enrollment verification requests 5-8
Step 1 Verify form setup 5-8
Step 2 Customize procedure definition in Banner Web Tailor 5-8

Implementation for registration and student accounts 5-9
Step 1 Customize procedure definition in Banner Web Tailor 5-9
Step 2 Establish link to payment card payments 5-10

Implementation for transcript requests 5-11
Step 1 Verify form setup 5-11
Step 2 Customize procedure definitions in Banner Web Tailor 5-11

Implementation for graduation applications 5-12
Step 1 Verify form setup 5-12
Step 2 Customize procedure definition in Banner Web Tailor 5-13
Step 3 Customize information text in Banner Web Tailor 5-14

Web pages 5-14
Select a Waiver (bwskapmt.P_SelectWaiver) 5-14
Process a Waiver (bwskapmt.P_ProcessWaiver) 5-14
Application Fee Payment (bwskalog.P_ProcIndex) 5-14
Enrollment Verification Request Summary (bwskrqst.P_Disp_Confirm) 5-15
Credit Card Payment (bwckcpmt.P_CCPaymentTermSelected) 5-15
Transcript Request Summary (bwskwtrr.P_Disp_Confirm) 5-15
Graduation Application Summary (bwskgrad.P_Disp_Payment) 5-15

Chapter 6 Banner Flexible Registration Implementation 6-1

Processing features 6-1

Implementation 6-2
Step 1 Verify common implementation 6-2
Step 2 Verify AR holds indicator 6-2
Step 3 Verify default term code 6-2
Step 4 Verify process code 6-2
Step 5 Enable payment card processing 6-2
Step 6 Configure success and failure URLs 6-3

Chapter 7 Banner Advancement Implementation 7-1

Campaign Detail Form (AFACAMP) 7-1

Designations Form (ADADESG) 7-1

Advancement Prospect Information Form (AMAINFO) 7-2

Fiscal Year Validation Form (ATVFISC) 7-2

Chapter 8 Banner Advancement Self-Service Implementation 8-1

Processing features 8-1

Implementation for Banner Advancement Self-Service 8-1
Step 1 Verify common implementation 8-2
Step 2 Verify form setup 8-2
Step 3 Indicate display of gift number on receipts 8-2

Step 4 Indicate display of donor ID on receipts 8-2
Step 5 Customize procedure definitions in Banner Web Tailor 8-3
Step 6 Customize information text in Banner Web Tailor 8-6
Step 7 Customize rules for gifts made with Banner Advancement Self-Service. . 8-7

Web pages 8-9

Make a Donation (bwakgift.P_Make_A_Donation and bwakgift.P_Pledge_Payment) 8-9
Credit Card Payment (bwakgift.P_Make_A_Donation and bwakgift.P_Pledge_Payment) 8-9
Online Receipt (bwakgift.P_Donation_Receipt) 8-10
Make a Donation (bwakngft.P_Make_A_Donation) 8-10
Online Receipt (bwakngft.P_Donation_Receipt) 8-10

Troubleshooting T-1

Payment not applied to account T-1

Missing payment processing Web page T-1

1 Overview

The Payment Processor Connection connects Banner® applications with third-party vendors that process payment card transactions. Payment amounts are entered or acknowledged in Banner. The Payment Processor Connection uses URL redirects and Web services to connect Banner to vendor applications where payment card information is entered, processed, and confirmed.

Payment card processing can be implemented for the following types of payments made in Banner:

Application	Payment
Banner Student Self-Service	Admissions applications Enrollment verification requests Registration and student accounts Transcript requests Graduation applications
Banner Advancement Self-Service	Gifts and pledge payments
Banner Flexible Registration	Registration (non-credit and traditional)

PCI compliance

The Payment Card Industry (PCI) Security Standards Council is a forum that develops, enhances, stores, disseminates, and implements security standards for payment card transactions on a global basis. PCI security standards affect the storage, transmission, and processing of cardholder data as part of the authorization or settlement of payment card transactions. PCI standards apply to organizations that store, transmit, and process cardholder data. PCI standards also apply to software vendors who develop payment applications that store, transmit, and process cardholder data.

Ellucian applications do *not* store, transmit, and process cardholder data. Rather, cardholder data is stored, transmitted, and processed by a PCI-compliant application or service that is provided by your payment processing vendor. The Payment Processor Connection connects your Ellucian applications with this payment processing vendor.

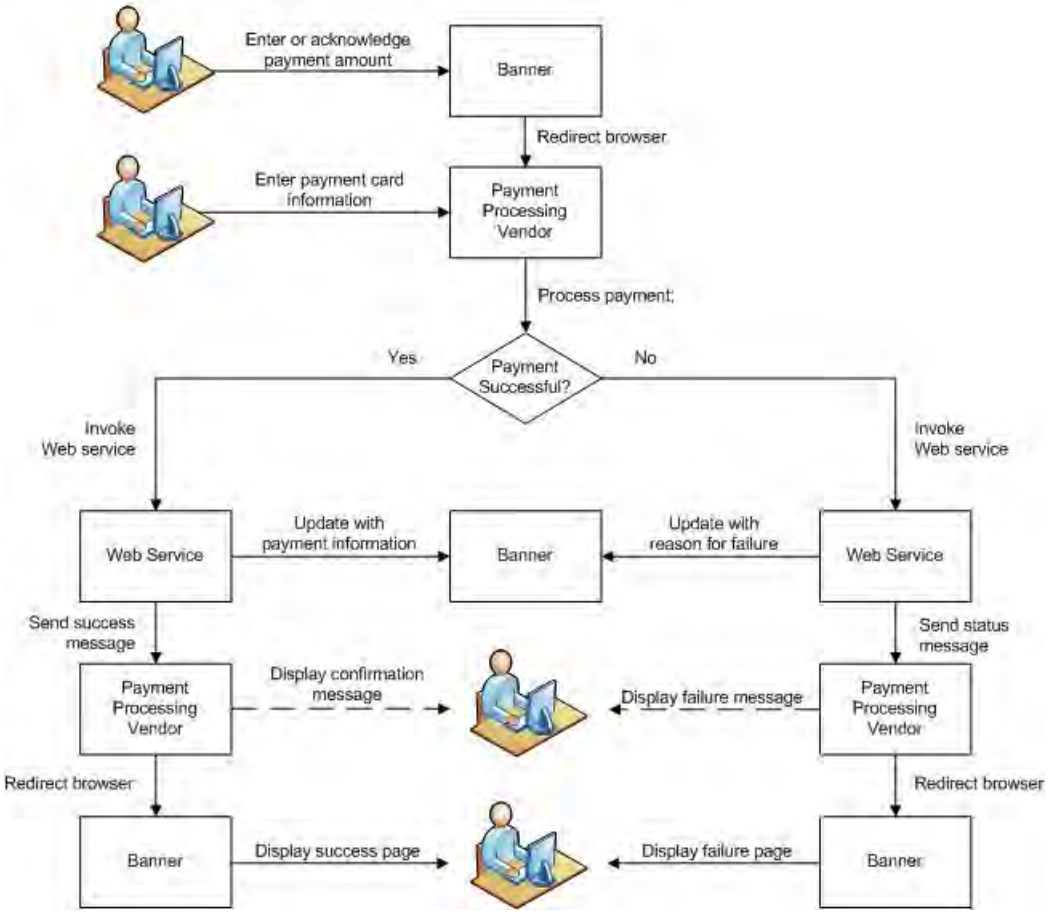
Supported vendors

Ellucian supports the following third-party vendors for payment card processing:

- CASHnet
- Nelnet
- Official Payments
- TouchNet

Processing flow

Certain tasks performed in Banner require a payment (for example, tuition payment or a donation). If the user chooses to pay with a payment card, the following processing occurs:



Detailed processing steps are as follows:

1. The user enters or acknowledges the payment amount in a Banner application. (If the amount is pre-determined, the amount does not have to be entered.) No personally identifiable information is entered in Banner.

2. The user clicks the button or link in the Banner application to submit the transaction and create a Banner transaction record. Once the transaction is submitted, the amount cannot be changed.

3. The Banner application redirects the browser to the payment processing vendor's Web site (identified by the Banner Web Tailor `PAYVEND_URL` parameter).

4. The user accesses the payment processing vendor's Web site. The access procedure depends on the payment processing vendor.

5. The user enters payment card information and follows the instructions on the payment processing vendor's Web site.

6. The payment processing vendor processes the payment. The subsequent processing depends on whether the payment processing succeeds or fails.

7. If the payment processing vendor successfully processes the payment, the following processing occurs:

    7.1. The payment processing vendor invokes a Web service, which is exposed by Banner, to update the Banner transaction record with the results of the payment transaction. No personally identifiable information is transmitted to or stored in Banner

    7.2. The Web service responds with a message to the payment processing vendor stating whether the Banner update was successful or not.

    7.3. The payment processing vendor might display a confirmation page or send a confirmation e-mail to the payee. If a confirmation page is displayed, the user follows instructions on the confirmation page to return to Banner.

    7.4. The payment processing vendor redirects the browser to Banner.

    7.5. Banner displays the success page.

8. If the payment processing vendor does not successfully process the payment, the following processing occurs:

    8.1. The payment processing vendor invokes a Web service, which is exposed by Banner, to update the Banner transaction record with the reason why the payment transaction failed.

8.2. The Web service responds with a message to the payment processing vendor providing an acknowledgement of the status update.

8.3. The payment processing vendor might display a failure confirmation page. Error messages are defined by the payment processing vendor. If a failure confirmation page is displayed, the user follows instructions on the failure confirmation page to resubmit the payment information (use a different payment card), cancel the transaction, or log out of the site.

8.4. The payment processing vendor redirects the browser to Banner.

8.5. Banner displays the failure page.

Data exchange

When a payment transaction is initiated, a new record is created in the Credit Card Audit Table (GORCCAU) to track the transaction. The following information is stored in the new record and is also transmitted to the payment processing vendor:

- Transaction ID
- Transaction amount
- Transaction description (defined by Banner Web Tailor information text)
- Merchant ID (used to determine available payment methods)

After the payment processing vendor processes the payment transaction, information is returned to Banner and the associated GORCCAU record is updated.

For transactions that process successfully, the following information is returned:

- Transaction ID that originated in Banner
- Transaction amount processed by the vendor
- Transaction date (optional)
- Type of payment card (for example, Visa)
- Payment card authorization code
- Merchant ID that originated in Banner
- Vendor reference number

For transactions that fail, the following information is returned:

- Transaction ID that originated in Banner
- Reason for failure

Banner dependencies

The Payment Processor Connection requires the following products:

Product	Minimum Version
Banner General	Deployment on Oracle Application Server 10.1.3.4/5: General 8.0 with patch p1-639eqc_gen80200 (required)* and patch p1-i6z6yh_gen8040104 (recommended)* Deployment on Oracle WebLogic Server 11g: General 8.0 with patch p1-639eqc_gen80200 (required)* and patch p1-i6z6yh_gen8040104 (required)*
Banner Web General	8.0 with patch p1-639es5_bwg80200
Banner Web Tailor	8.0 with patch p1-7549uz_twb80201

*Patch p1-i6z6yh_gen8040104 contains the newest version of the adapter that exposes Banner functions to payment processing vendors. If you install both patches, you should install the ear file that is delivered in p1-i6z6yh_gen8040104, rather than the ear file that is delivered in p1-639eqc_gen80200.

Contents of this handbook

This handbook provides the steps that systems administrators use to install the Banner Payment Transaction Service Adapter. This adapter exposes key Banner functions to payment processing vendors.

This handbook also provides the steps used to implement payment card processing in Banner. This includes common implementation steps, plus implementation steps specific to Banner Student, Banner Student Self-Service, Banner Flexible Registration, Banner Advancement, and Banner Advancement Self-Service.

2 PaymentTransaction Web Service

When a payment transaction is entered in a Banner® application, the transaction is automatically redirected to an external payment processing vendor for processing. When processing is complete, the payment processing vendor invokes the PaymentTransaction Web service to post the payment transaction, if successful, to the originating Banner application, or to report the result, if failed.

This chapter provides information about the PaymentTransaction Web service. This includes instructions for installing the adapter that exposes the Web service, a description of each operation in the Web service, the mapping of message elements/attributes to Banner columns, and data setup requirements.

What is a Web service?

A Web service exposes an application's processing logic to support a service-oriented architecture and to facilitate integration with external systems. A Web service allows an external system or business process to invoke the application's logic without having to understand the application's internal structure.

Web services are based on open, Internet-based standards. This makes them relevant to application integration within an organization and with external organizations. Standards such as XML, SOAP, WSDL, and UDDI provide cross-platform compatibility that does not depend on a single programming language or network transport.

What is the Banner Payment Transaction Service Adapter?

The Banner Payment Transaction Service Adapter is a Java-based application that exposes the PaymentTransaction Web service. This exposure makes certain Banner functions available to payment processing vendors using the SOAP protocol over HTTP/HTTPS. Payment processing vendors interact with the Web service, which in turn is supported by Banner APIs. This layered approach provides an insulating buffer between payment processing vendors and Banner. Payment processing vendors do not interact with Banner directly, but rather exchange XML messages with the exposed Web service.

The Banner Payment Transaction Service Adapter supports the synchronous, request/reply message exchange pattern as follows:

1. The payment processing vendor requests a service of Banner by sending an XML message to the Web service endpoint (URL) that is exposed by the adapter. The message contains the information required for Banner to service the request. The URL that is exposed by the adapter follows this pattern:

   ```
   <http or https>://<host>:<port>/pci/v1_0
   ```

   See "Web service operations" on page 2-56 for more information about the URL.

2. The Banner Payment Transaction Service Adapter invokes the appropriate Banner API.

3. The Banner API performs the necessary Banner processing logic.

4. One of the following occurs:

   4.1. If the action is completed successfully, the API provides a response message, which the adapter forwards to the payment processing vendor.

   4.2. If the action is not completed successfully, the adapter sends an error message (called a SOAP fault) to the payment processing vendor.

The adapter is packaged as a J2EE compatible enterprise archive file named `pci_services.ear` (Oracle Application Server) or `pci_services_weblogic.ear` (Oracle Weblogic Server). The adapter is available in the Banner General Java subdirectory.

Requirements

The Banner Payment Transaction Service Adapter has the following requirements.

Oracle application server and Java

The Banner Payment Transaction Service Adapter is certified on Oracle Application Server (OAS) 10.1.3.4/5 and Oracle WebLogic Server 11g with Java 1.6.

OAS 10.1.3.4/5 is delivered with Java 1.5. The following Oracle document provides instructions for changing to Java 1.6. If you contract with Ellucian for Oracle support, you can access the FAQ on the Customer Support Center. Otherwise, you can use your Oracle support account to access the document.

Document Title:	How to change the Java version used to run a specific OC4J instance
Ellucian FAQ:	1-AXZ803
Oracle Doc ID:	351476.1

Oracle database

The required Oracle database depends on the application server that you are using:

Application Server	Required Database
Oracle Application Server 10.1.3.4/5	Oracle Database 10gR2 or 11g
Oracle WebLogic Server 11g	Oracle Database 11g

Banner Translation Service

The Banner Payment Transaction Service Adapter *does not* require the Banner Translation Service. The adapter, however, does require the definition of a data source that refers to the Banner Translation Service. Installation instructions in this chapter include a step for creating the data source and its associated connection pool. These components can reference the `transsvc` schema if Banner Web Services is installed, or they can use the `integmgr` schema, as documented in this chapter.

Installation on Oracle Application Server 10.1.3.4/5

The Banner Payment Transaction Service Adapter is packaged as a J2EE compatible enterprise archive file (`pci_services.ear`), which is available in the Banner General Java subdirectory.

Use the following steps to install the adapter on OAS 10.1.3.4/5:

- Step 1, “Create an OC4J instance”
- Step 2, “Install the adapter”
- Step 3, “Define the adapter data source”
- Step 4, “Define the Translation Service data source”
- Step 6, “Configure the security role and user”
- Step 7, “Enable schema validation (optional)”

- Step 8, “Configure logging”
- Step 9, “Verify the deployment”

The adapter can be installed on an existing Oracle Application Server. A separate OC4J instance should be dedicated for the adapter so that the PaymentTransaction Web service environment can be independently managed.

Step 1 Create an OC4J instance

The adapter can be installed on an existing application server, but a separate OC4J instance should be dedicated for the adapter. This allows you to independently manage the PaymentTransaction Web service environment. Use the following steps to create a new OC4J instance for the Banner Payment Transaction Service Adapter.

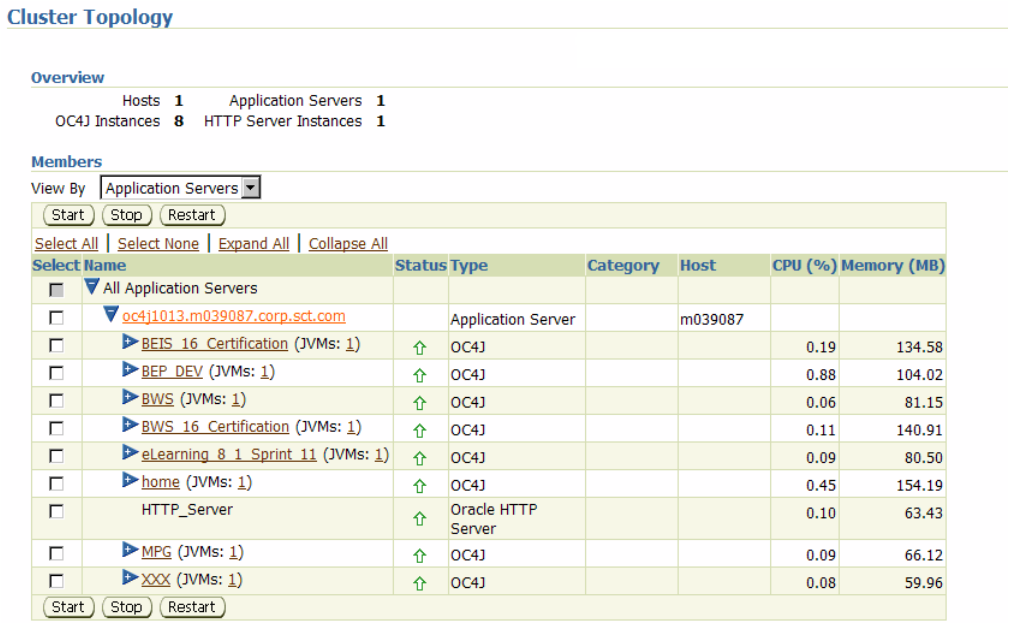
1. Connect to the Oracle Enterprise Manager:

   ```
   http://<host>:<port>/em
   ```

Note

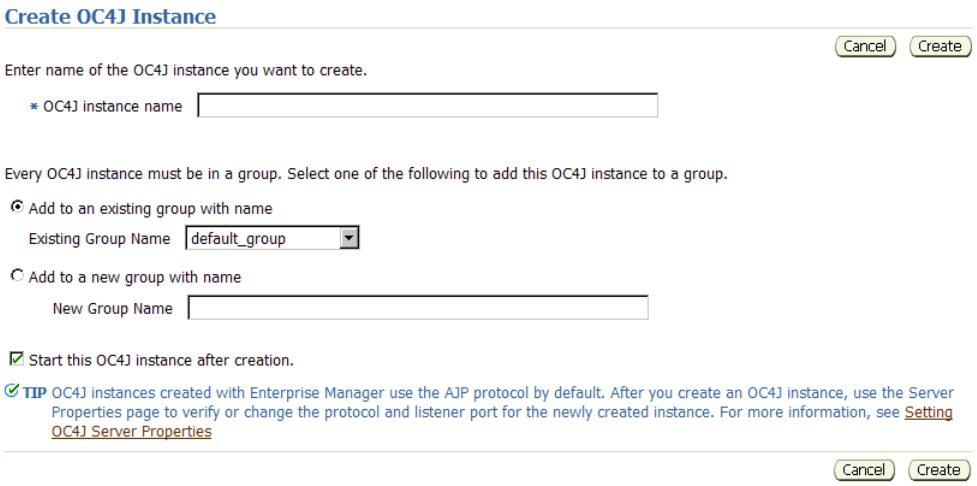
The adapter uses the server’s HTTP or HTTPS port. These ports are identified on the Runtime Ports page. ■

The console is displayed.



2. Click the name of the application server that will contain the new instance. (The value of the **Type** field should be *Application Server.*) The Application Server page is displayed.

3. Click **Create OC4J Instance**. The Create OC4J Instance page is displayed.



4. Enter the following information to create the new instance:

OC4J instance name	Enter a meaningful name for the new instance.
Start this OC4J instance after creation	Select the check box.

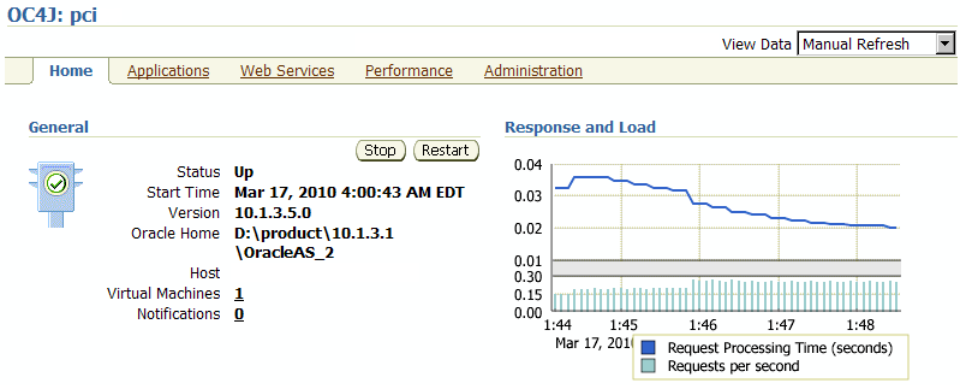
5. Click **Create**. The instance is created and started. A confirmation message is displayed.

6. Click **OK**. The System Components list is redisplayed with the new instance.

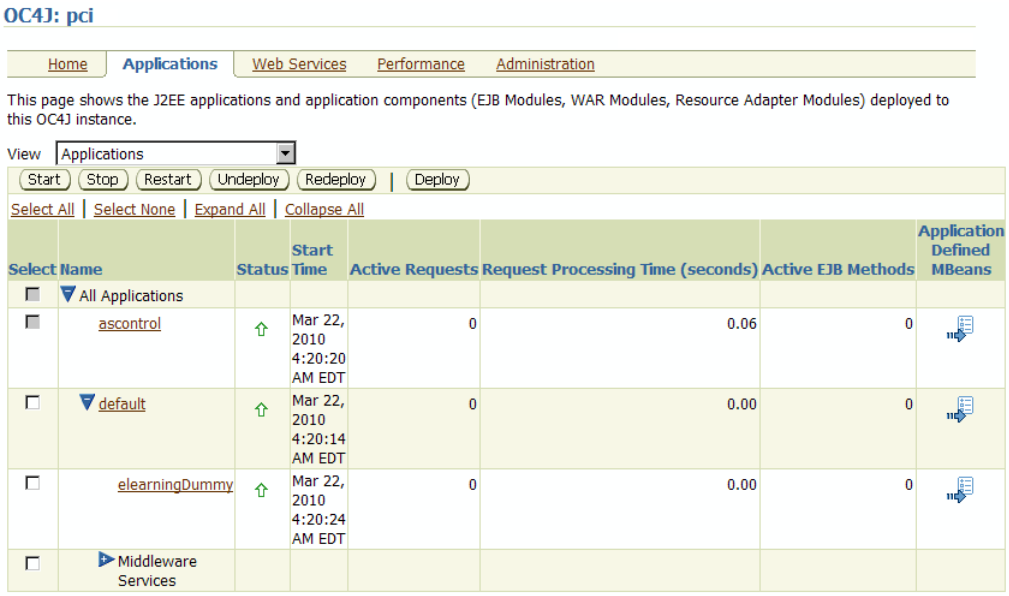
Step 2 Install the adapter

Use the following steps to deploy the adapter to the Oracle Application Server.

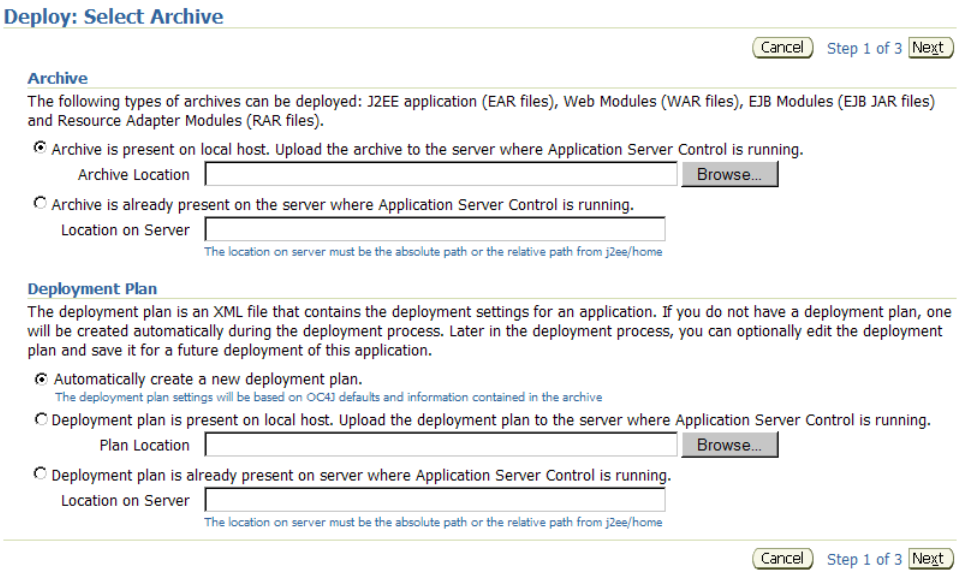
1. In the System Components list, click the name of the OC4J instance that you created in Step 1, "Create an OC4J instance". The Home page for the selected instance is displayed.



2. Select the **Applications** tab. A list of deployed applications is displayed.



3. Click **Deploy**. The Deploy: Select Archive page is displayed.



4. Select the file to be uploaded:

   4.1. In the Archive section, select **Archive is present on local host. Upload the archive to the server where Application Server Control is running.**

   4.1. In the **Archive Location** field, click **Browse** and navigate to the `pci_services.ear` file.

   4.2. Select the file and click **Open**.

5. If you are installing on OAS 10.1.3.5, go to step 6. A deployment plan is not needed.

   -or-

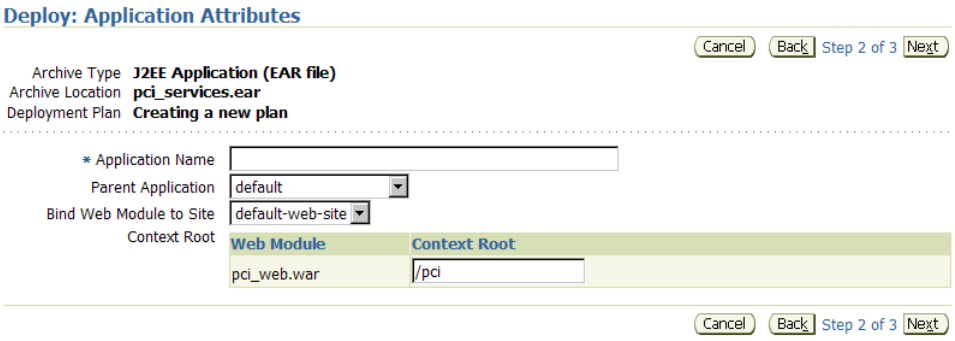
   If you are installing on OAS 10.1.3.4, select the deployment plan for the adapter:

   5.1. In the Deployment Plan section, select **Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running.**

   5.2. In the **Plan Location** field, click **Browse** and navigate to the `deployment_plans` folder.

   5.3. Select the deployment plan for OAS 10.1.3.4 and click **Open**.

6. Click **Next** on the Deploy: Select Archive page. The files are uploaded and the Deploy: Application Attributes page is displayed.



7. Enter a name for the application (for example, *Payment_Transaction_Service*) in the **Application Name** field.

Warning

If the adapter is being deployed in multiple OC4J instances on the same server, the application name must be unique for each deployment. ■

8. If the adapter is being deployed in one instance only, accept the default context root and go to step 9.

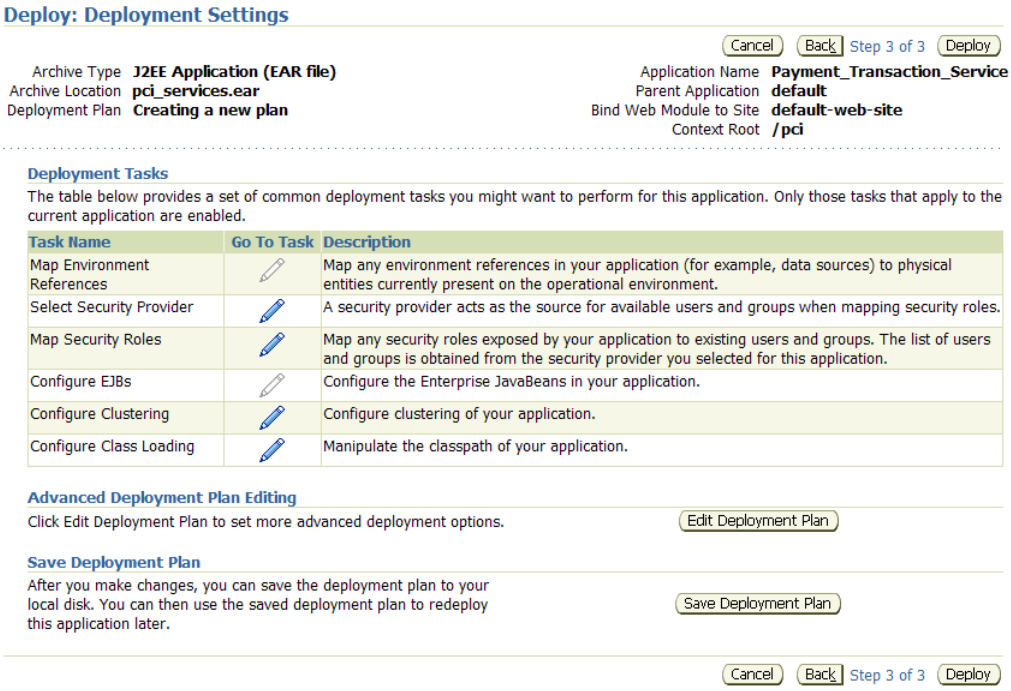
   -or-

   If the adapter is being deployed in multiple OC4J instances on the same server, add descriptive text to the default context root in the **Context Root** field (for example, /`pci/test` or /`pci/prod`). Adding descriptive text to the default string, rather than changing the entire default string, is preferable. Use this new context root in any subsequent steps that refer to the default URL.

Warning

If the adapter is being deployed in multiple OC4J instances on the same server, the context root must be unique for each deployment. ■

9. Click **Next**. The Deploy: Deployment Settings page is displayed.



10. Click **Deploy** to accept the values and install the adapter. A deployment confirmation page is displayed.

11. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

Step 3 Define the adapter data source

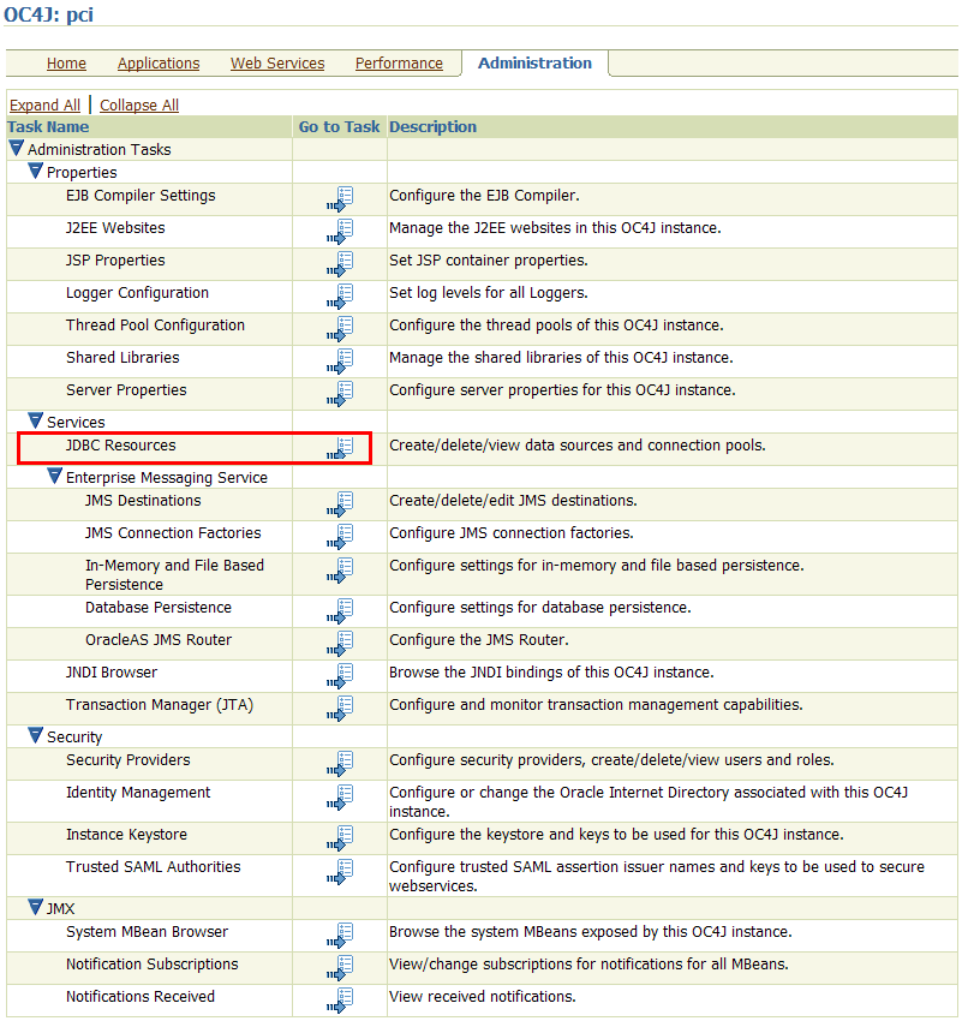
A data source provides the connection properties to the Banner database. By default, the adapter needs a data source with lookup name `jdbc/bannerws`.

There are two ways to define a data source:

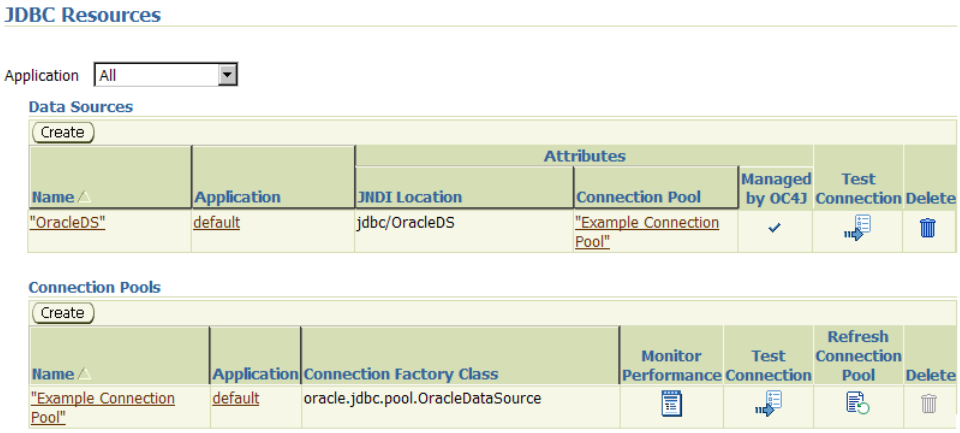
- **At the OC4J instance level** - This method promotes resource sharing, allowing multiple applications in the instance to use the same connection pool to connect to the database.
- **At the application level** - This method permits each application in the instance to access the database via an application-specific connection pool.

Use the following steps to define the connection pool and data source.

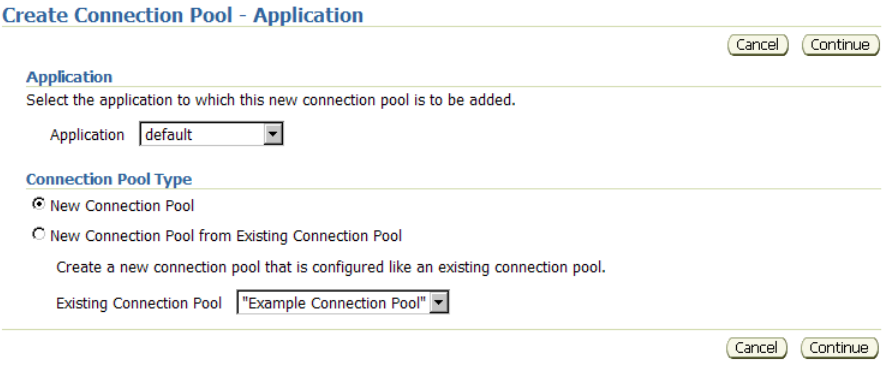
1. Select the **Administration** tab. A list of tasks is displayed.



2. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.



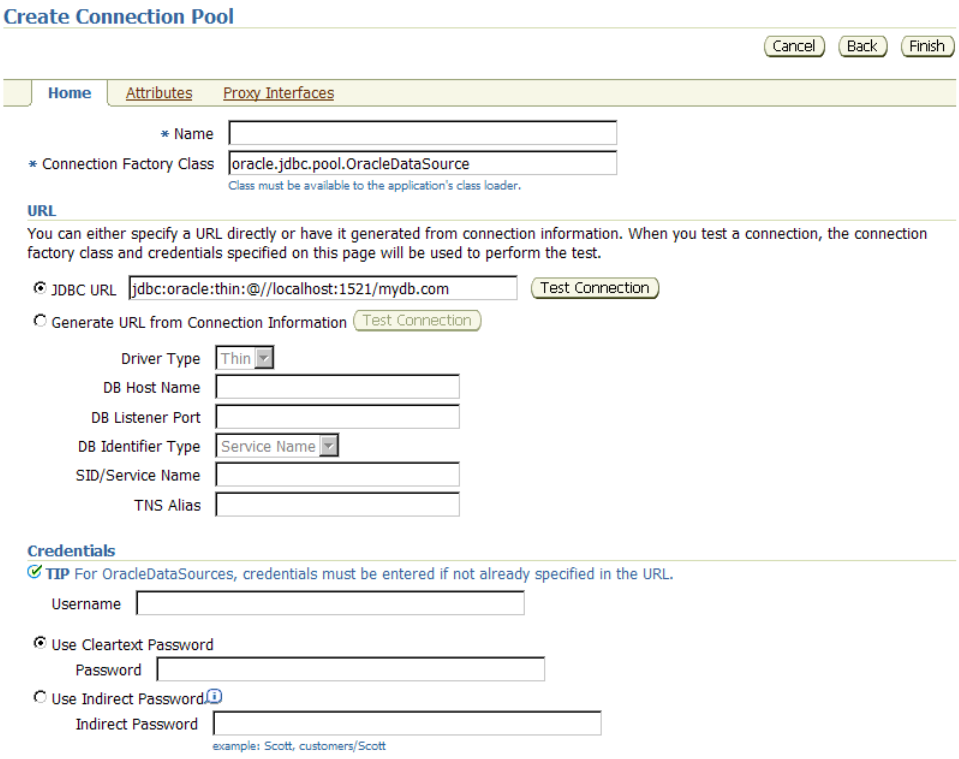
3. Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.



4. Select the application and connection pool type for the new pool:

Application	If you want to define the connection pool at the OC4J level, select *default*. All applications in the instance will use this connection pool. If you want to define the connection pool at the application level, select the adapter application.
New Connection Pool	Select the button.

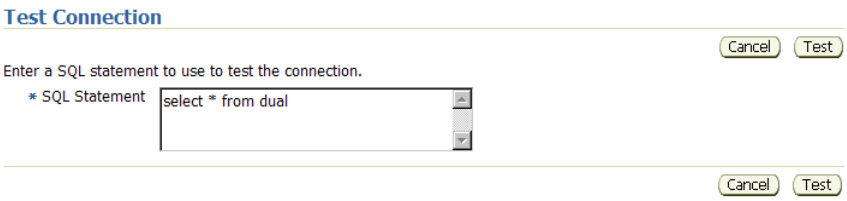
5. Click **Continue**. The Create Connection Pool page is displayed.



6. Enter the following information to set up the connection pool for the `integmgr` schema:

Name	*bannerWS_pool* (This is an example. Enter the name of your choice.)
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*integmgr*
Use Cleartext Password	Select **Use Cleartext Password** and enter a password for the `integmgr` schema.

7. Click **Test Connection**. The Test Connection page is displayed.



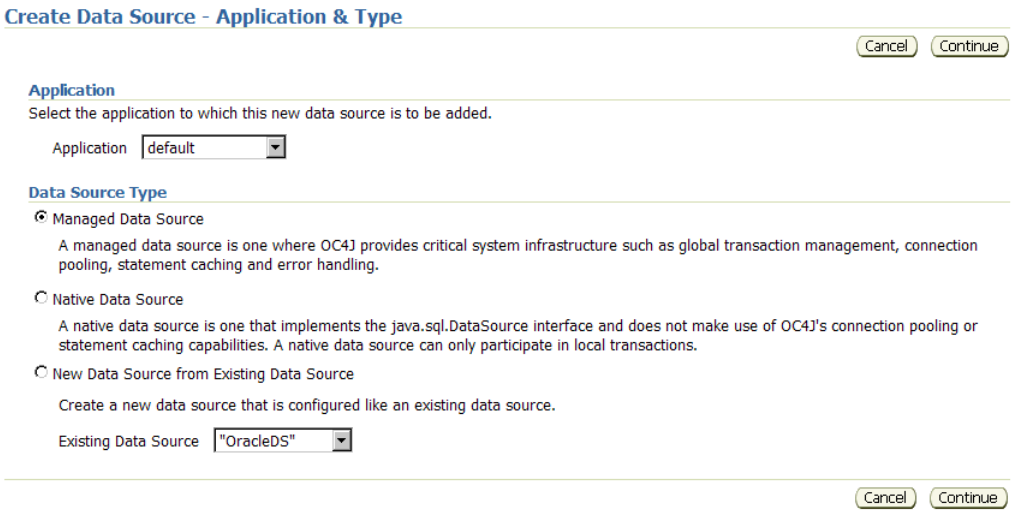
8. Click **Test** to test the connection pool for the `integmgr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

   8.1. If the test succeeds, continue with the next step.

   8.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

9. Click **Finish**.

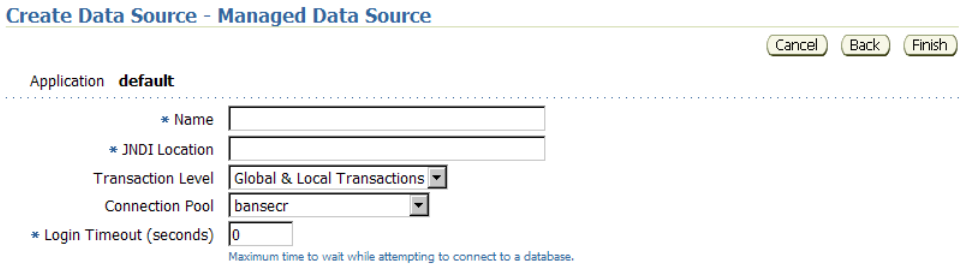
10. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.



11. Select the application and data source type for the data source:

Application	If you want to define the data source at the OC4J level, select *default*. All applications in the instance will use this data source. If you want to define the data source at the application level, select the adapter application.
Managed Data Source	Select the button.

12. Click **Continue**. The Create Data Source - Managed Data Source page is displayed.



13. Enter the following information to set up the `bannerWS` data source:

Name	*bannerWS*
JNDI Location	*jdbc/bannerws*
Connection Pool	*bannerWS_pool*

14. Click **Finish**.

Step 4 Define the Translation Service data source

By default, the adapter needs a data source with lookup name `jdbc/transsvc.` The PaymentTransaction Web service, however, does not request value translations. For this reason, the Translation Service data source and its associated connection pool can be defined to reference any database schema.

Use the following steps to define the connection pool and data source by copying the adapter connection pool and data source that you defined in step 3.

1. Select the **Administration** tab. A list of tasks is displayed.

2. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.

3. Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.

4. Select the application and connection pool type for the new pool:

Application	If you want to define the connection pool at the OC4J level, select *default*. All applications in the instance will use this connection pool. If you want to define the connection pool at the application level, select the name of the deployed adapter.
New Connection Pool from Existing Connection Pool	Select the button.
Existing Connection Pool	Select *"bannerWS_pool"*.

5. Click **Continue**. The Create Connection Pool page is displayed.

6. Enter *Transsvc_pool* in the **Name** field.

7. Click **Finish**.

8. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

9. Select the application and data source type for the data source:

Application	If you want to define the data source at the OC4J level, select *default*. All applications in the instance will use this data source. If you want to define the data source at the application level, select the name of the deployed adapter.
New Data Source from Existing Data Source	Select the button.
Existing Data Source	Select *"bannerWS"*.

10. Click **Continue**. The Create Data Source - Managed Data Source page is displayed.

11. Enter the following information to set up the `transsvc` data source:

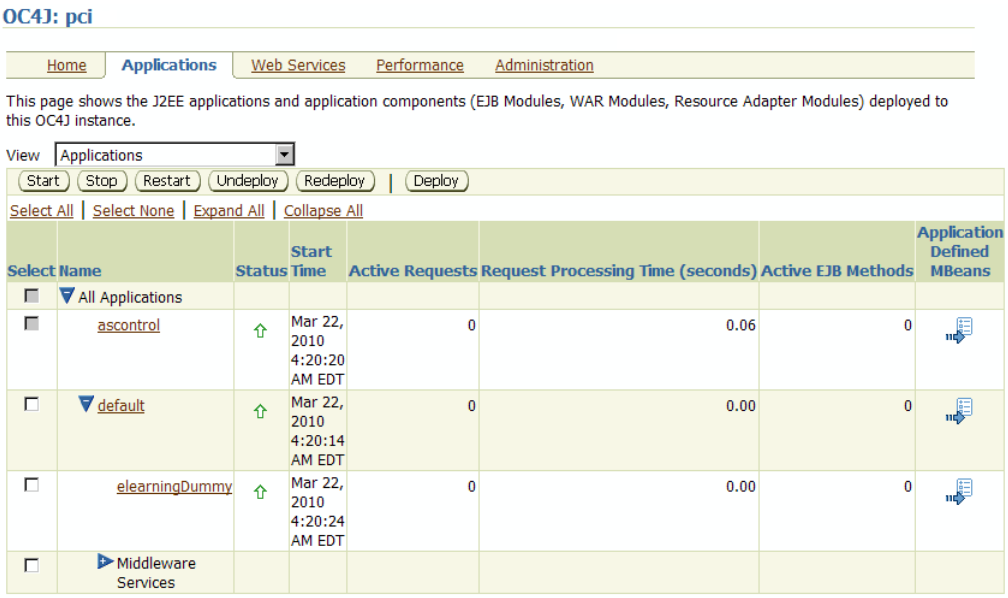
Name	*transsvc*
JNDI Location	*jdbc/transsvc*
Connection Pool	*Transsvc_pool*

12. Click **Finish**.

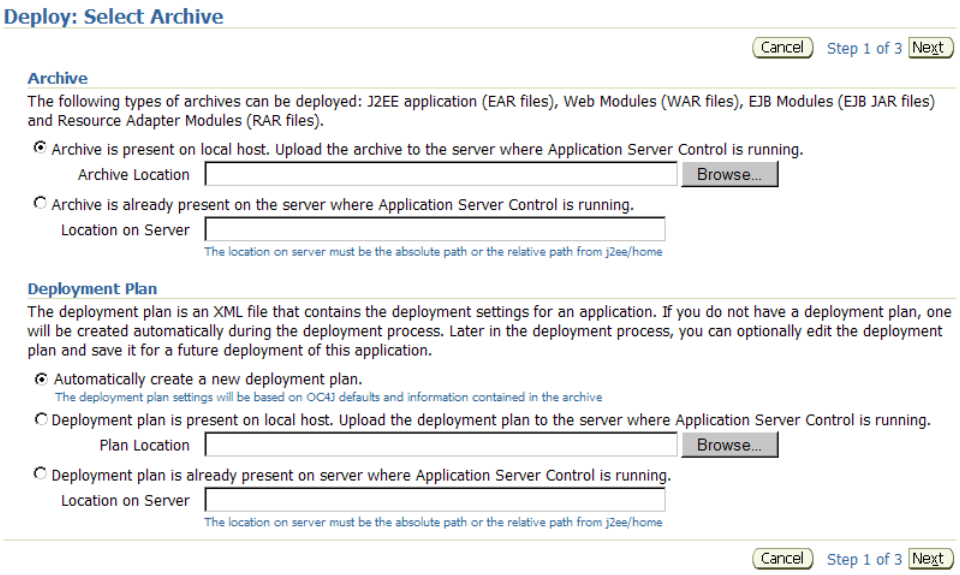
Step 5 Install the adapter

Use the following steps to deploy the adapter to the Oracle Application Server.

1. Select the **Applications** tab. A list of deployed applications is displayed.



2. Click **Deploy**. The Deploy: Select Archive page is displayed.



3. Select the file to be uploaded:

   3.1. In the Archive section, select **Archive is present on local host. Upload the archive to the server where Application Server Control is running.**

   3.1. In the **Archive Location** field, click **Browse** and navigate to the `pci_services.ear` file.

   3.2. Select the file and click **Open**.

4. If you are installing on OAS 10.1.3.5, go to step 6. A deployment plan is not needed.

   -or-

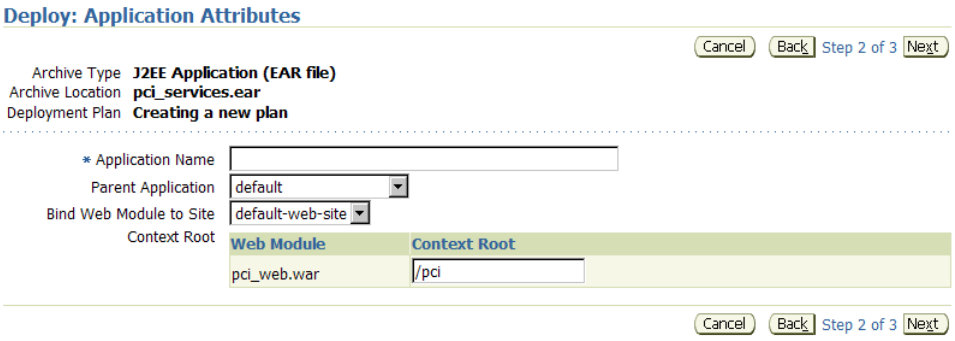
   If you are installing on OAS 10.1.3.4, select the deployment plan for the adapter:

   4.1. In the Deployment Plan section, select **Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running.**

   4.2. In the **Plan Location** field, click **Browse** and navigate to the `deployment_plans` folder.

   4.3. Select the deployment plan for OAS 10.1.3.4 and click **Open**.

5. Click **Next** on the Deploy: Select Archive page. The files are uploaded and the Deploy: Application Attributes page is displayed.



6. Enter a name for the application (for example, *Payment_Transaction_Service*) in the **Application Name** field.

Warning

If the adapter is being deployed in multiple OC4J instances on the same server, the application name must be unique for each deployment. ■

7. If the adapter is being deployed in one instance only, accept the default context root and go to step 9.

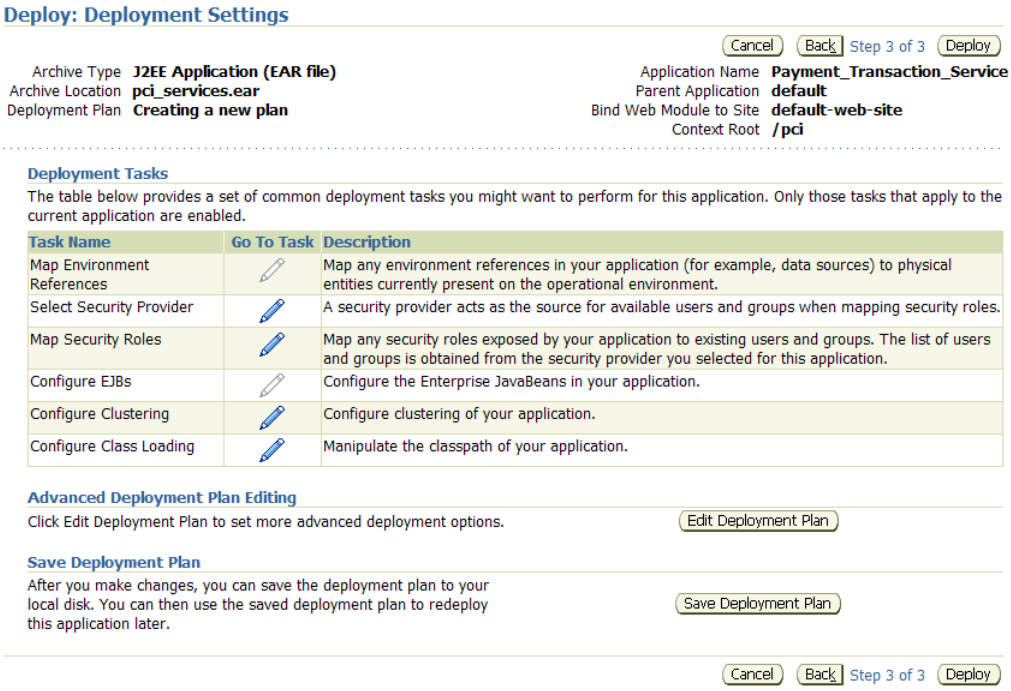
   -or-

   If the adapter is being deployed in multiple OC4J instances on the same server, add descriptive text to the default context root in the **Context Root** field (for example, /`pci/test` or /`pci/prod`). Adding descriptive text to the default string, rather than changing the entire default string, is preferable. Use this new context root in any subsequent steps that refer to the default URL.

Warning

If the adapter is being deployed in multiple OC4J instances on the same server, the context root must be unique for each deployment. ■

8. Click **Next**. The Deploy: Deployment Settings page is displayed.



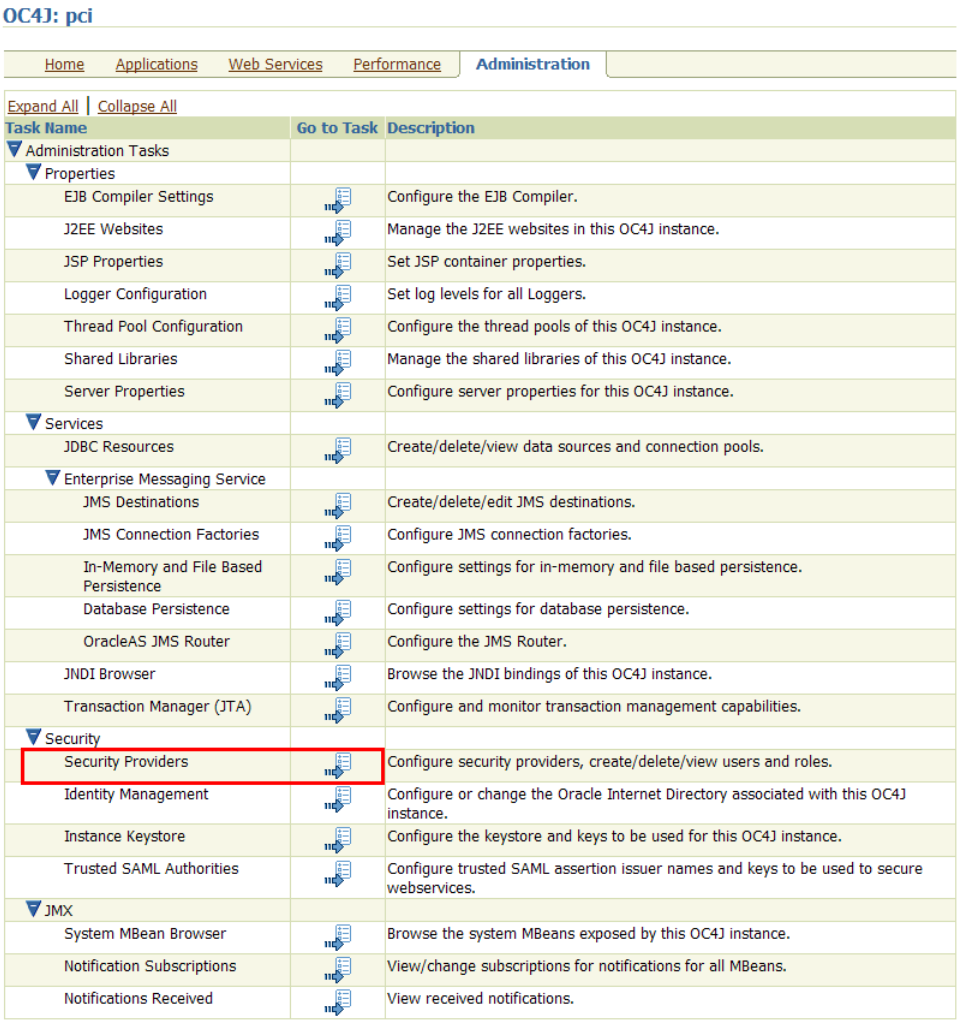
9. Click **Deploy** to accept the values and install the adapter. A deployment confirmation page is displayed.

10. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

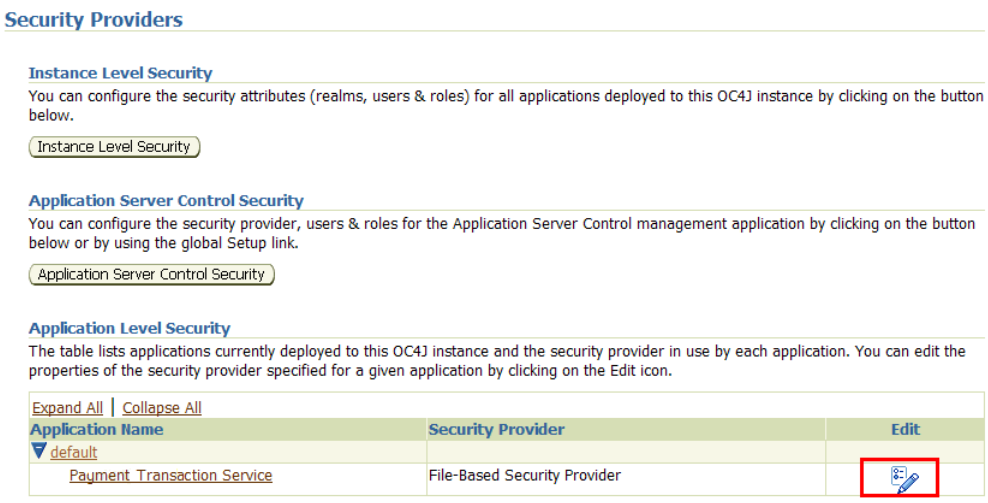
Step 6 Configure the security role and user

Use the following steps to add the `bannerws` role and an administrative user to the Banner Payment Transaction Service Adapter application. This role and user protect the defined endpoint.

1. Select the **Administration** tab. A list of tasks is displayed.

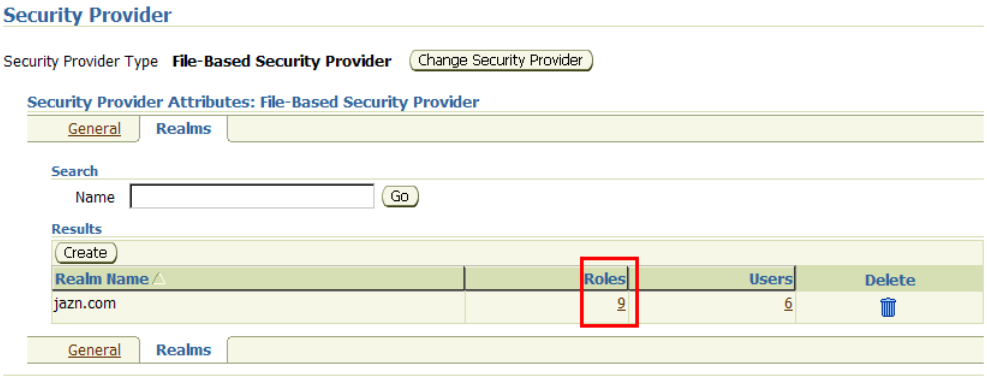


2. Select **Security Providers** in the Security section. The Security Providers page is displayed.



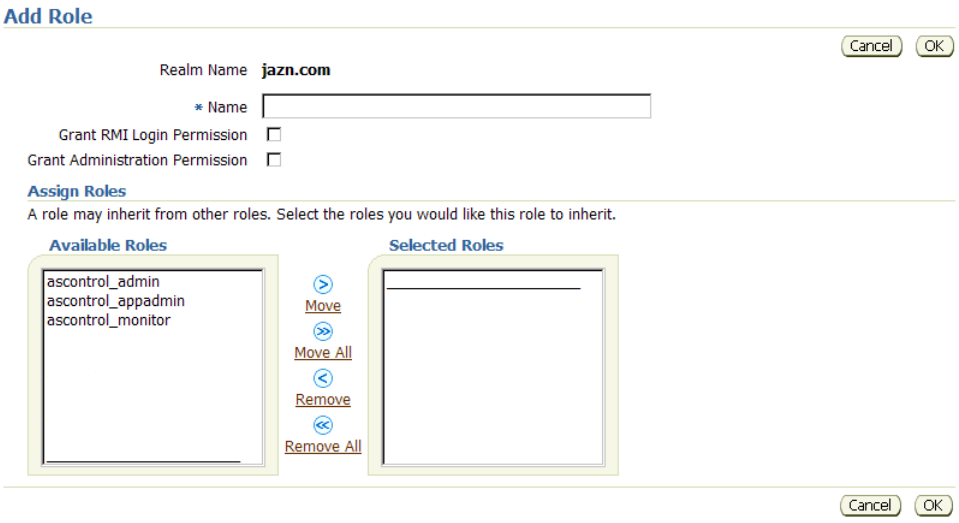
3. In the Application Level Security section, click the **Edit** button for the adapter application. The Security Provider page is displayed.

4. Select the **Realms** tab.



5. Click the link under the **Roles** column. The Roles page is displayed.

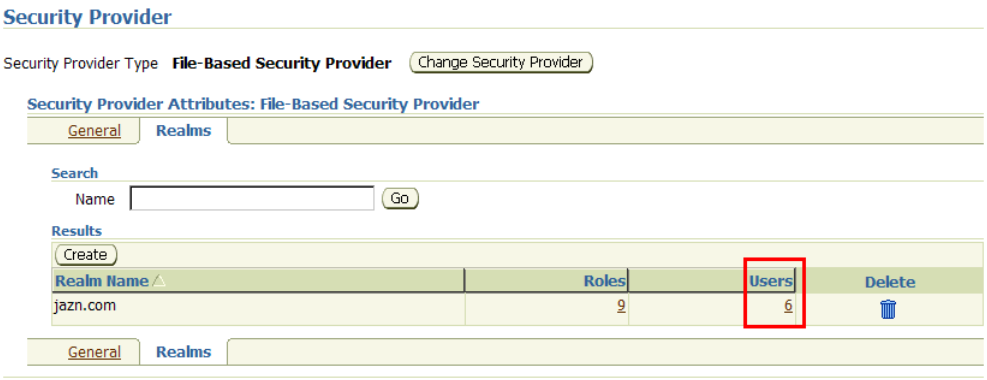
6. Click **Create**. The Add Role page is displayed.



7. Enter *bannerws* in the **Name** field.

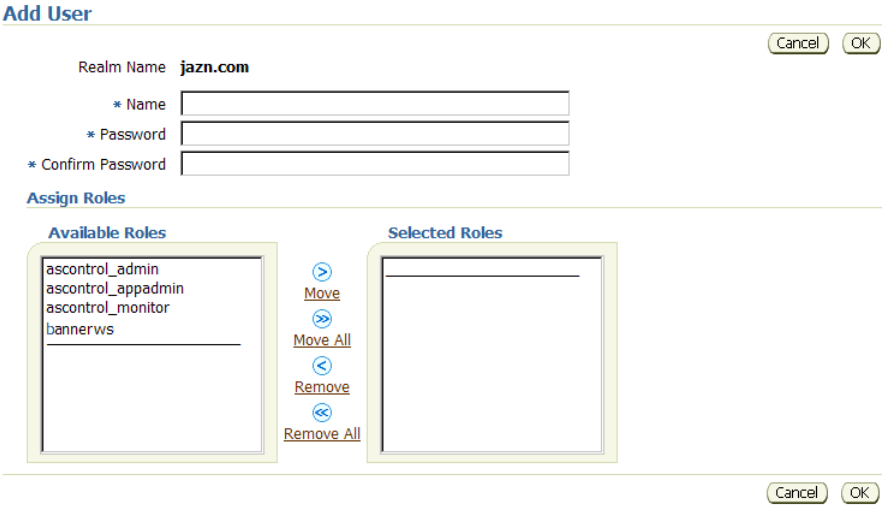
8. Click **OK**. The Roles page is redisplayed with the new role.

9. Return to the Security Provider page.



10. Click the link under the **Users** column. The Users page is displayed.

11. Click **Create**. The Add User page is displayed.



12. Enter the following information to create a user:

Name	*payment_vendor* (This is an example. Enter the name of your choice.)
Password	Password for the user being created
Confirm Password	Confirmation of the password

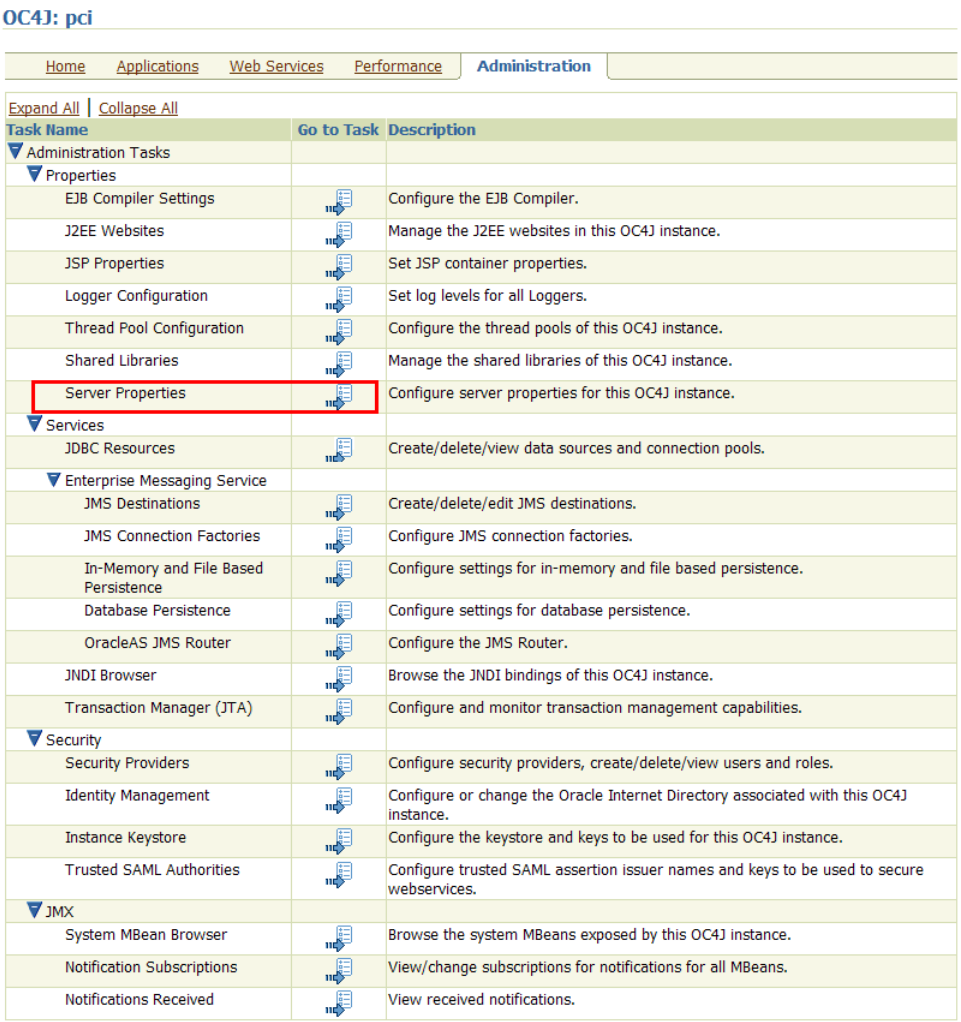
13. In the Assign Roles section, select the *bannerws* role in the **Available Roles** list and move it to the **Selected Roles** list.

14. Click **OK**. The Users page is redisplayed with the new user.

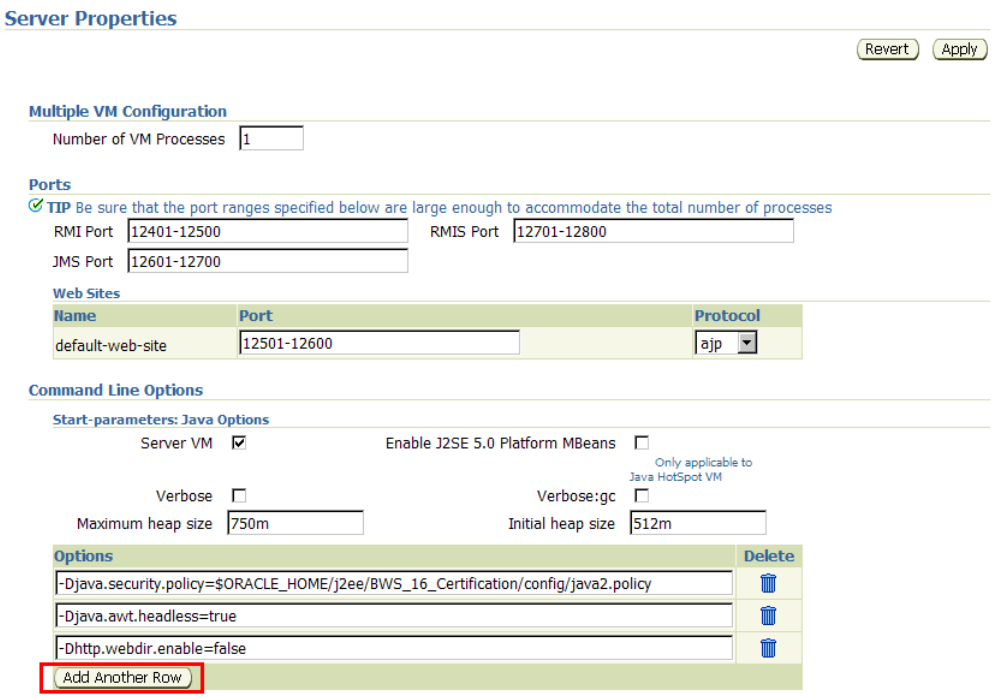
Step 7 Enable schema validation (optional)

Validating XML request and response messages for each Web service invocation degrades system performance. For this reason, schema validation is disabled by default. To enable schema validation, you must set system property `BANNERWS_SCHEMA_VALIDATION` with a value of *true* for the OC4J instance where the adapter is installed. Use the following steps to enable schema validation.

1. Select the **Administration** tab. A list of tasks is displayed.



2. Select **Server Properties** in the Properties section. The Server Properties page is displayed.



3. In the Command Line Options section, click **Add Another Row**.

4. Add the following system property:

```
-DBANNERWS_SCHEMA_VALIDATION=true
```

5. Click **Apply**. A confirmation message is displayed.

6. Click **Yes** to restart the server.

Step 8 Configure logging

The Banner Payment Transaction Service Adapter uses Apache's log4j to log the activities performed by the application at runtime. Log4j uses a properties file to establish specific runtime options. The following options should be reviewed and modified as appropriate:

- **Location of the log files**. The default location is `<IAS_HOME>/j2ee/home/log`. This location should be changed to the OC4J instance where the Banner Payment Transaction Service Adapter is installed.
- **Logging level**. The default level is *INFO*, resulting in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide detailed logging for initial operations, you should change the logging level to *DEBUG*.

Note

You should change the logging level for initial operation only. ■

Use the following steps to modify the logging options as appropriate.

1. Navigate to `<IAS_HOME>/j2ee/<OC4J instance>/applications/<Web services adapter name>/pci_web/WEB-INF/classes`.

2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
`log4j.appender.out.File`	`log/pci_ws.log`	`../<OC4J instance>/log/pci_ws.log`
`log4j.category.com.sungardsct`	`INFO`	`DEBUG`

3. Restart the OC4J instance for the changes to take effect.

Step 9 Verify the deployment

Use the following steps to verify that the adapter is successfully deployed.

1. Use a Web browser to access the URL mapped in Step 3, “Define the adapter data source”:

   `<http or https>://<host>:<port>/<context root>/`

   This URL is used to access an information page, not the endpoint. The context root is *pci*, unless the context root was changed when the adapter was installed.

2. Log in with the name and password configured in Step 6, “Configure the security role and user”. An information page for the adapter is displayed. This page shows the version of the adapter and provides a link to the Web service’s WSDL.

3. (Optional) If you need to determine the URL that is being used to listen for messages, click the link on the information page to display the associated WSDL. The `<soap:address location>` attribute under the `<service>` tag at the bottom of the WSDL identifies the URL that you should always use to invoke the Web service.

In some situations, you can access the home page via a browser, but payment transactions are not logged in the `pci_ws.log` or in the redirect log for the container where the adapter is installed. This may be due to a wrong port configuration. The Web service uses the HTTP or HTTPS port of the server. These ports are configurable, by institution, via the Oracle Enterprise Manager console and are identified on the Runtime Ports page:

Runtime Ports

Expand All | Collapse All

Name	Host	Port In Use	Port Range	Configure Port
All Application Servers				
oc4j1013.m039087.corp.sct.com	m039087			
OPMN				
Local		6100	6100	
Remote		6201	6201	
Request		6004	6004	
OC4J : Ant_Installer_Test				
JMS		12602	12601-12700	
AJP		12501	12501-12600	
RMIS		12702	12701-12800	
RMI		12402	12401-12500	
OC4J : BWS_812				
JMS		12603	12601-12700	
AJP		12502	12501-12600	
RMIS		12703	12701-12800	
RMI		12403	12401-12500	
OC4J : BEIS_16_Certification				
RMI		12414	12401-12500	
RMIS		12704	12701-12800	
JMS		12614	12601-12700	
AJP		12503	12501-12600	
OC4J : home				
JMS		12601	12601-12700	
AJP		8888	8888	
RMIS		12701	12701-12800	
RMI		12401	12401-12500	
Oracle HTTP Server : HTTP_Server				
HTTPS1		4443		
HTTP2		7201		
HTTP1		7777		

Installation on Oracle WebLogic Server 11g

The Banner Payment Transaction Service Adapter is packaged as a J2EE compatible enterprise archive file (`pci_services_weblogic.ear`), which is available in the Banner General Java subdirectory.

Recommended configuration

The adapter must be installed in a Basic WebLogic Server Domain 10.3.4.0 or above. It must not be installed using any other Oracle WebLogic template, especially the Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual server for Ellucian middle-tier components. This server would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server instance should consist of the default Admin Server and at least one Managed Server for the deployment of the Banner Payment Transaction Service Adapter.

If a domain based on the Basic Domain template already exists for middle-tier applications, the adapter can be installed in a separate Managed Server in that domain.

Refer to the Oracle WebLogic Server Documentation Library for details on creating a new domain and a new Managed Server.

Installation steps

Use the following steps to install the adapter on Oracle WebLogic Server 11g:

- Step 1, “Configure logging (optional)”
- Step 2, “Define the adapter data source”
- Step 3, “Define the Banner Translation Service data source”
- Step 4, “Install the adapter”
- Step 5, “Configure the security group and user”
- Step 6, “Enable schema validation (optional)”
- Step 7, “Verify the deployment”

The adapter can be installed on an existing WebLogic Basic domain. It cannot be installed on a Classic Domain that comes with the `WLS_FORMS` and `WLS_REPORTS` servers. A separate Managed Server should be dedicated for the adapter so that the PaymentTransaction Web service environment can be independently managed.

Step 1 Configure logging (optional)

The Banner Payment Transaction Service Adapter uses Apache’s log4j to log the activities performed by the application at runtime. The log file is located at the following location:

```
Oracle\Middleware\user_projects\domains\<domain_name>\log
```

where <domain_name> is the name of the domain where the Banner Payment Transaction Service Adapter will be installed. This location cannot be changed.

A property in the log4j.properties file determines the logging level. The default logging level is *INFO*, which results in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. Use the following steps to modify the logging level if you want more detailed logging for initial operations.

Note

You should change the logging level for initial operation only. ■

1. Copy pci_services_weblogic.ear to a temporary location. This location is referred to as <EAR_HOME>.

2. Navigate to <EAR_HOME> and execute the following command.

   ```
   jar xvf pci_services_weblogic.ear
   ```

   The extract contains a Web archive named pci_web.war.

3. Create a folder under <EAR_HOME> and name it war_home.

4. Navigate to war_home and execute the following command.

   ```
   jar xvf <EAR_HOME>/pci_web.war
   ```

5. Open war_home\WEB-INF\classes\log4j.properties.

6. Edit the log4j.category.com.sungardsct property as follows:

   Original value: *INFO*
   New value: *DEBUG*

7. Save the change.

8. From war_home execute the following command to rebuild the Web archive file.

   ```
   jar cvf <EAR_HOME>/pci_web.war META-INF/* WEB-INF/* ui/*
   index.jsp
   ```

9. From <EAR_HOME> execute the following command to rebuild the enterprise archive file.

   ```
   jar cvf pci_services_weblogic.ear *.war META-INF/* legal/*
   APP-INF/*
   ```

   The rebuilt ear file is used for installation.

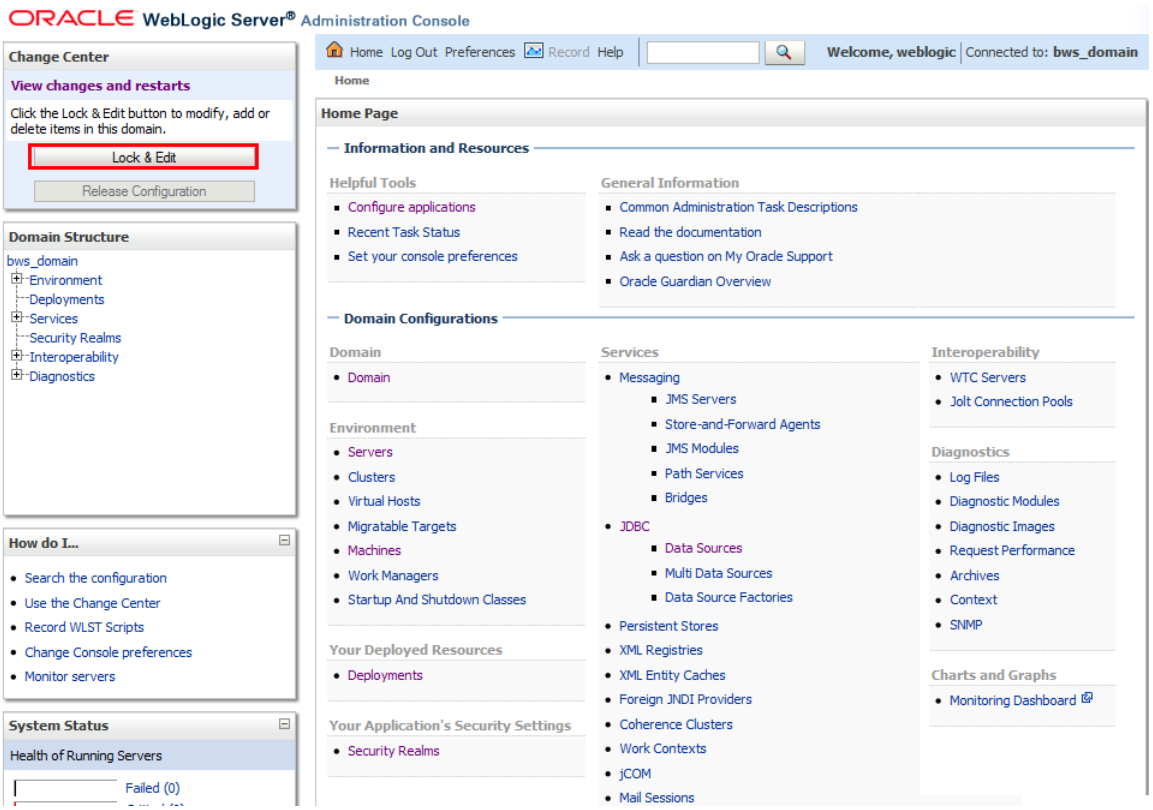
Step 2 Define the adapter data source

A data source provides the connection properties to the Banner database. By default, the adapter needs a data source with lookup name `jdbc/bannerws`. Use the following steps to define the data source.

1. Connect to the Oracle WebLogic Server Administration Console:

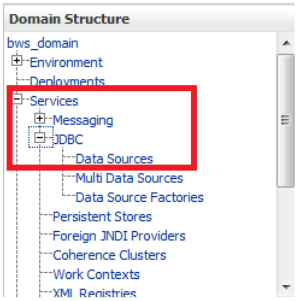
   `http://<host>:<port>/console`

   The Home Page is displayed.

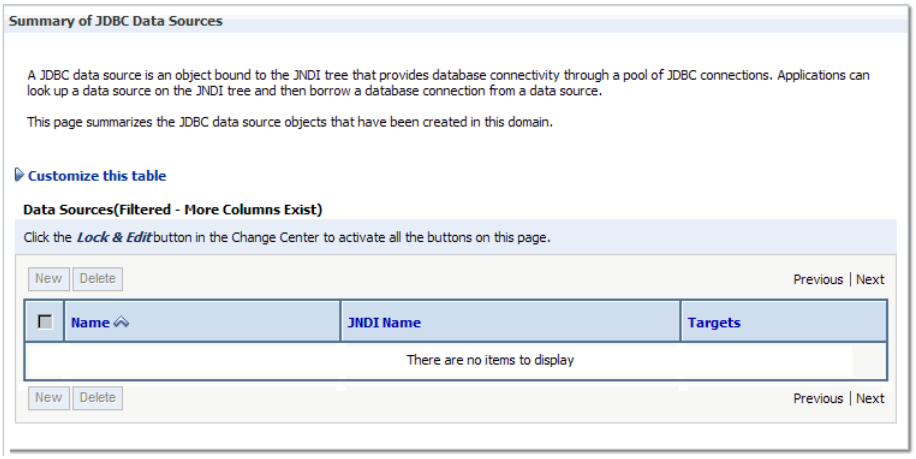


2. In the Change Center pane, click **Lock & Edit**.

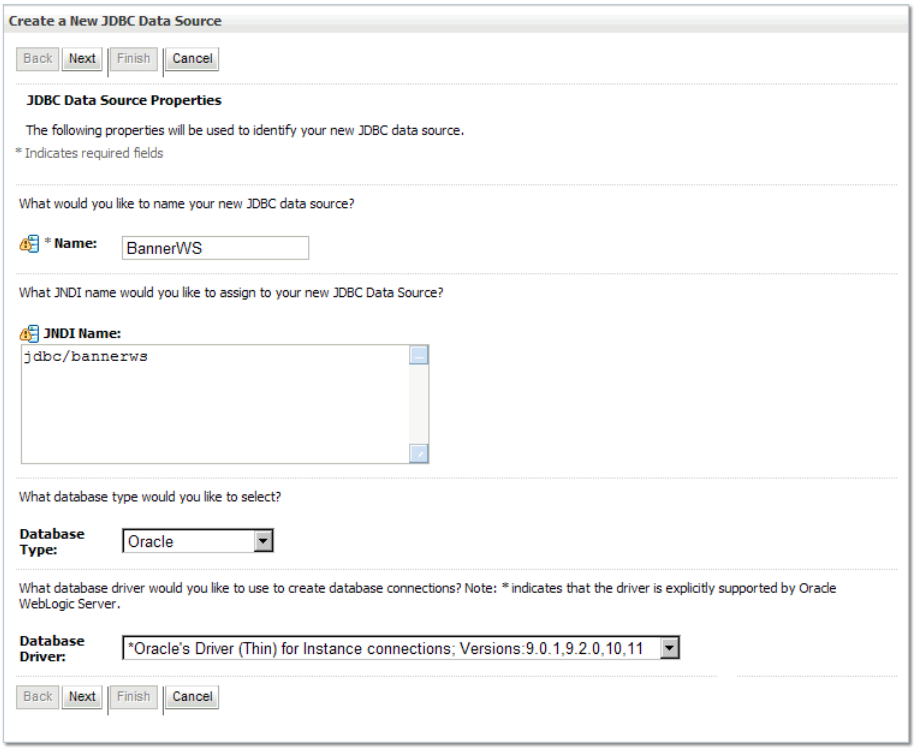
3. In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**.



The Summary of JDBC Data Sources page is displayed.



4. Click **New**. The Create a New JDBC Data Source page is displayed.

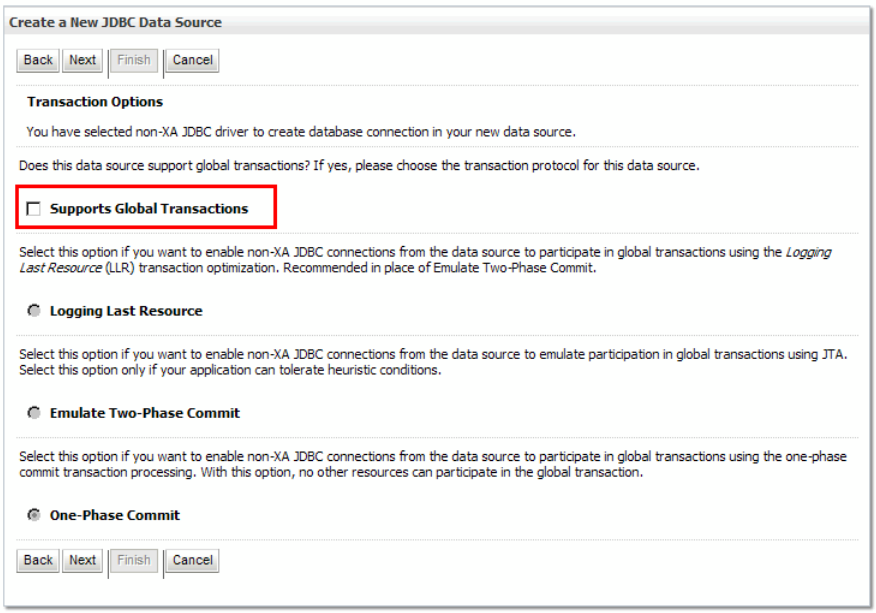


5. Enter the following data source properties:

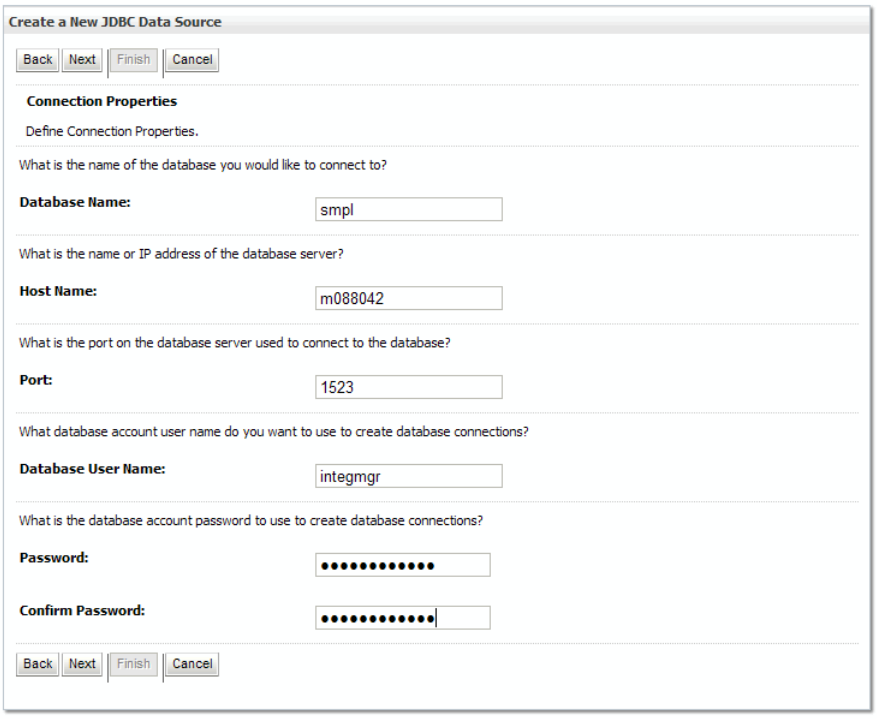
Name	*BannerWS*
JNDI Name	*jdbc/bannerws*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections: • If your database is RAC-based, select *Oracle's Driver (Thin) for RAC Service -Instance connections; Versions:10,11*. • Otherwise, select *Oracle's Driver (Thin) for Instance connections; Versions:9.0.1, 9.2.0,10,11*.

6. Click **Next**. The next page is displayed.

   6.1. In some domains, the Transaction Options are displayed. Clear the **Supports Global Transactions** check box and click **Next** to display the Connection Properties. Then go to step 7.



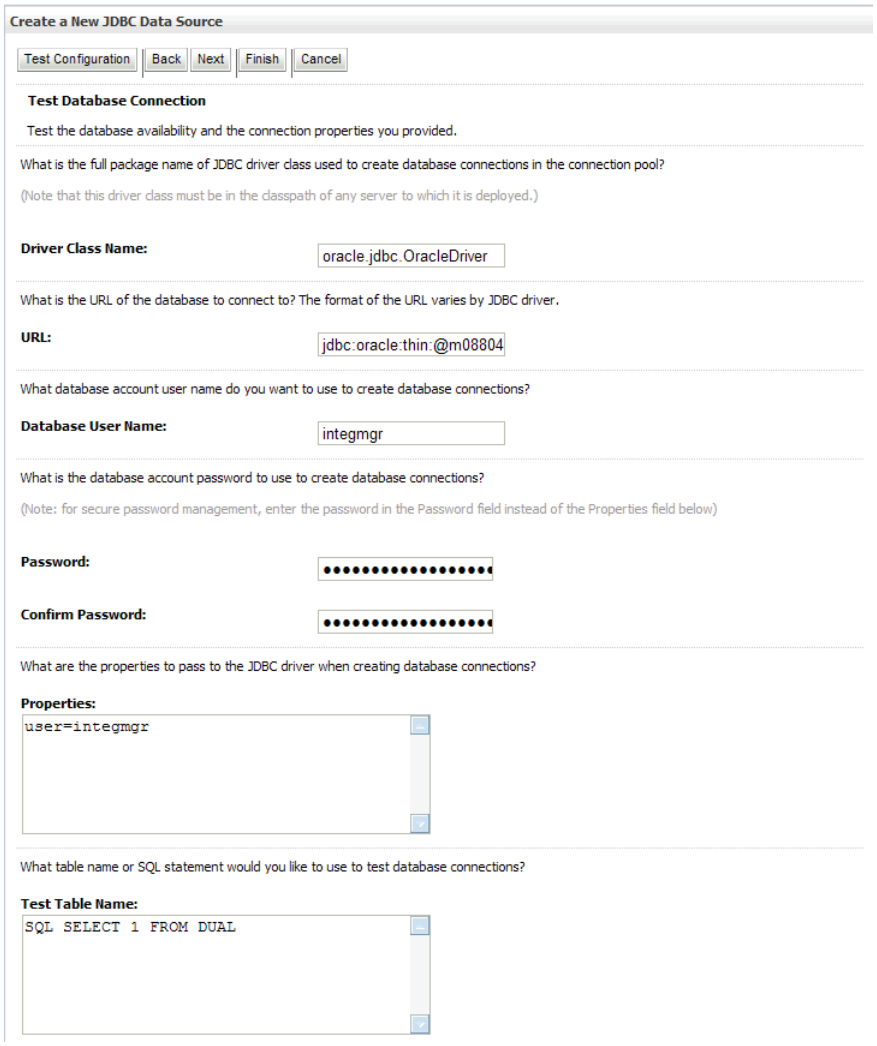
6.2. In some domains, the Transaction Options are skipped and the Connection Properties are displayed. Go directly to step 7.



7. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*integmgr*
Password	Password for the `integmgr` user
Confirm Password	Confirmation of the password

8. Click **Next**. The next page is displayed with the properties that you entered.



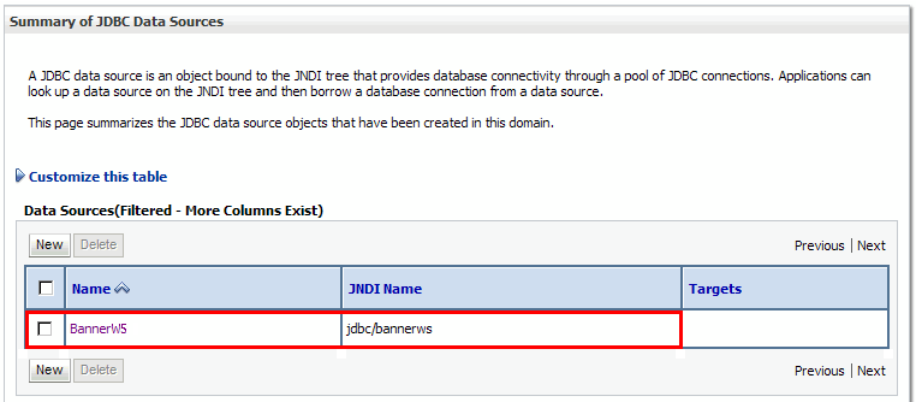
9. Verify the property values.

10. Click **Test Configuration**. The page is redisplayed with a success or failure message.

    10.1. If the test succeeds, continue with the next step.

    10.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

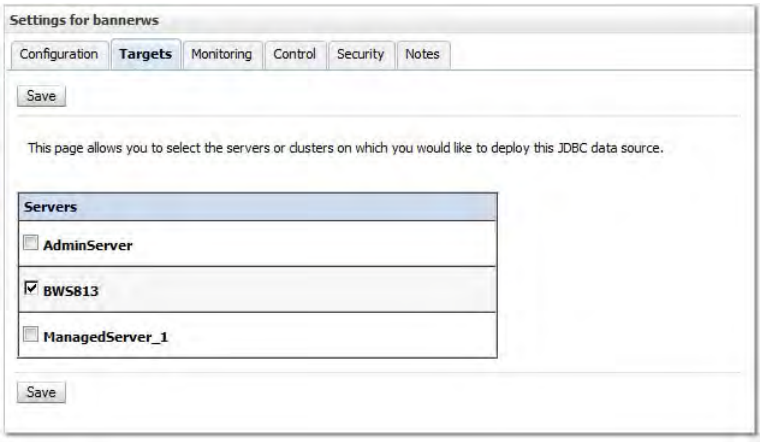
11. Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.



12. In the Change Center pane, click **Activate Changes**.

13. On the Summary of JDBC Data Sources page, click the name of the new data source. The Settings for BannerWS page is displayed.

14. Select the **Targets** tab.



15. In the Change Center pane, click **Lock & Edit**.

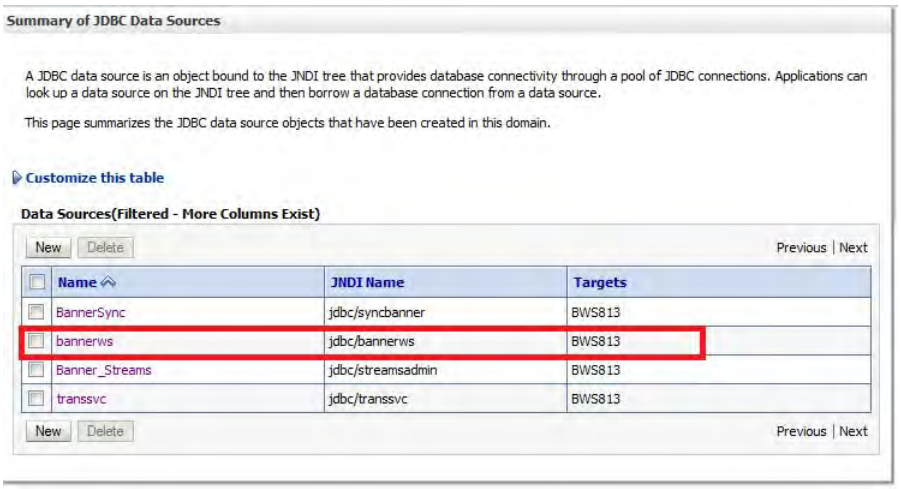
16. On the Settings for BannerWS page, select the server where the data source should be deployed.

17. Click **Save**.

18. In the Change Center pane, click **Activate Changes.**

19. In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**.

The Summary of JDBC Data Sources page is displayed.



20. Verify that the new data source is associated with the server.

Step 3 Define the Banner Translation Service data source

By default, the adapter needs a data source with lookup name `jdbc/transsvc.` Use the following steps to define the data source.

1. In the Change Center pane, click **Lock & Edit**.

2. Ensure that the Summary of JDBC Data Sources page is displayed. (If it is not displayed, expand and click **Services -> JDBC -> Data Sources** in the Domain Structure pane.)

3. Click **New** on the Summary of JDBC Data Sources page. The Create a New JDBC Data Source page is displayed.

4. Enter the following data source properties:

Name	*transsvc*
JNDI Name	*jdbc/transsvc*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections: • If your database is RAC-based, select *Oracle's Driver (Thin) for RAC Service -Instance connections; Versions:10,11*. • Otherwise, select *Oracle's Driver (Thin) for Instance connections; Versions:9.0.1, 9.2.0,10,11*.

5. Click **Next**. The next page is displayed.

   5.1. In some domains, the Transaction Options are displayed. Clear the **Supports Global Transactions** check box and click **Next** to display the Connection Properties. Then go to step 6.

   5.2. In some domains, the Transaction Options page is skipped and the Connection Properties are displayed. Go directly to step 6.

6. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*integmgr*
Password	Password for the `integmgr` user
Confirm Password	Confirmation of the password

7. Click **Next**. The next page is displayed with the properties that you entered.

8. Verify the property values.

9. Click **Test Configuration**. The page is redisplayed with a success or failure message.

    9.1. If the test succeeds, continue with the next step.

    9.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

10. Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.

11. In the Change Center pane, click **Activate Changes**.

12. On the Summary of JDBC Data Sources page, click the name of the new data source. The Settings for transsvc page is displayed.

13. Select the **Targets** tab.

14. In the Change Center pane, click **Lock & Edit**.

15. On the Settings for transsvc page, select the server where the data source should be deployed.

16. Click **Save**.

17. In the Change Center pane, click **Activate Changes.**

18. In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**. The Summary of JDBC Data Sources page is displayed.

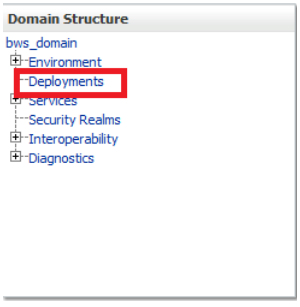
19. Verify that the new data source is associated with the server.

Step 4 Install the adapter

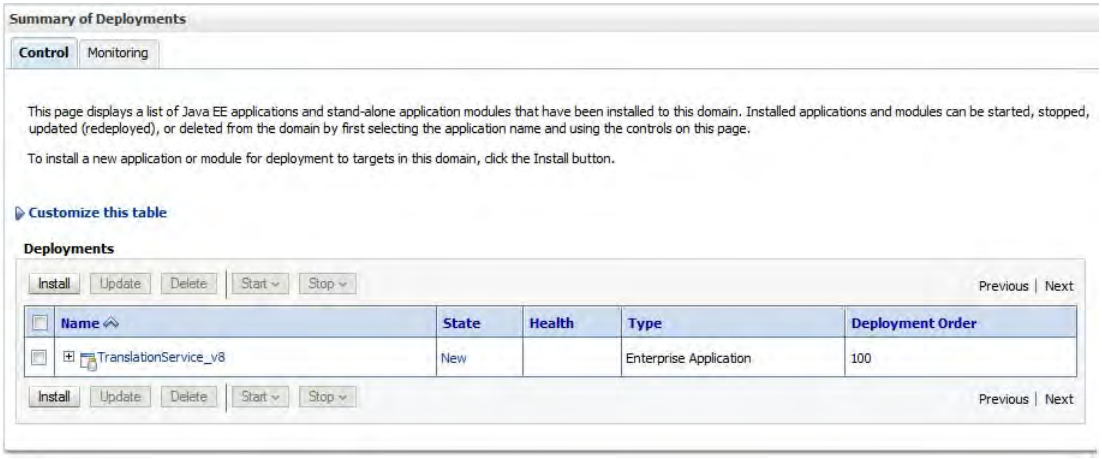
Use the following steps to install the adapter to the Oracle WebLogic Server.

1. In the Change Center pane, click **Lock & Edit**.

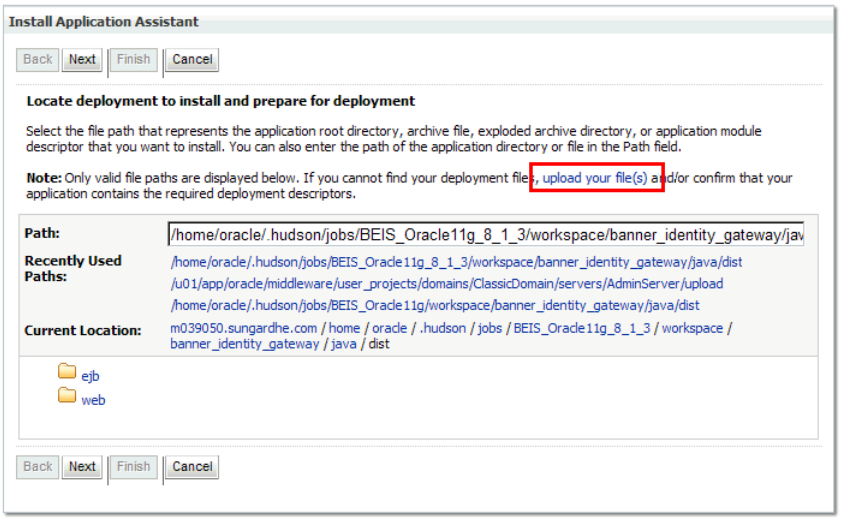
2. In the Domain Structure pane, click **Deployments**.



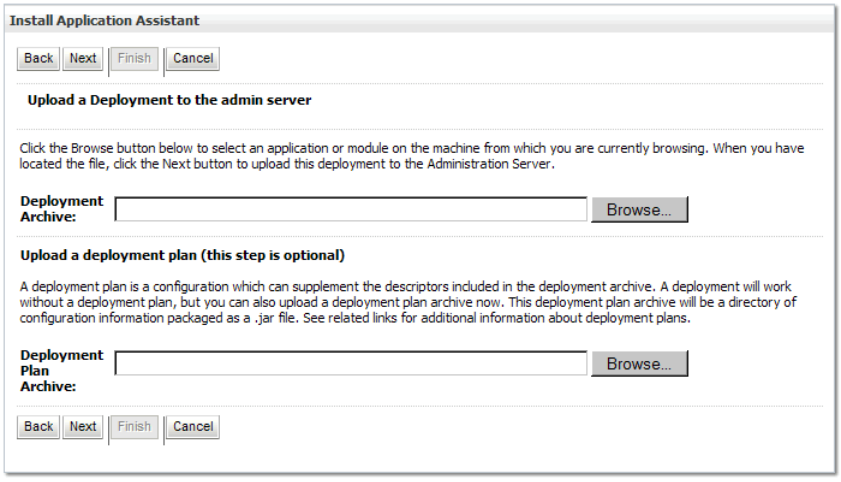
The Summary of Deployments page is displayed.



3. Click **Install**. The Install Application Assistant page is displayed.



4. Click **upload your file(s)**. The next installation page is displayed.



5. Select the file to be uploaded:

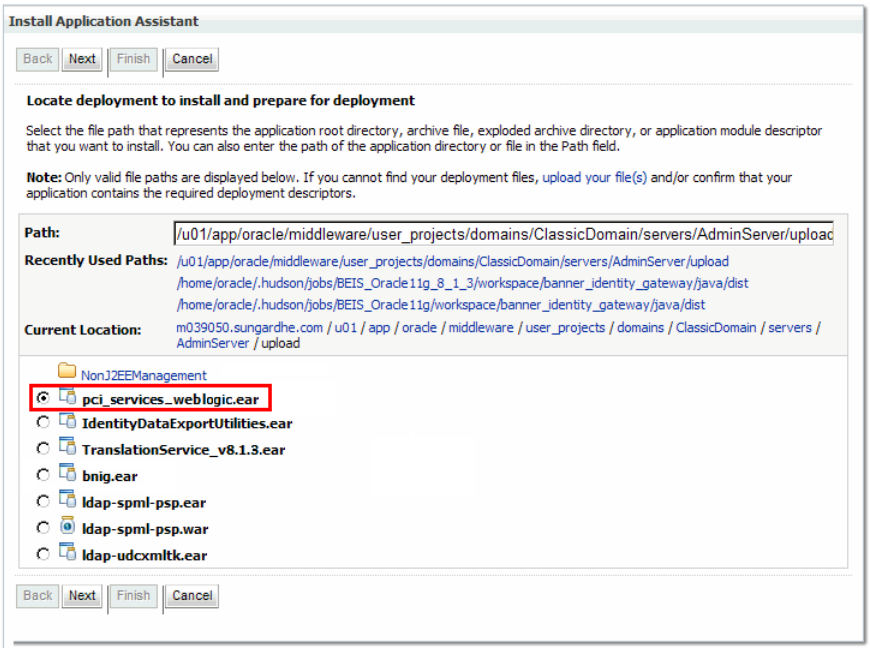
   5.1. In the **Deployment Archive** field, click **Browse** and navigate to `pci_services_weblogic.ear`.

   5.1. Select the file and click **Open**.

Note

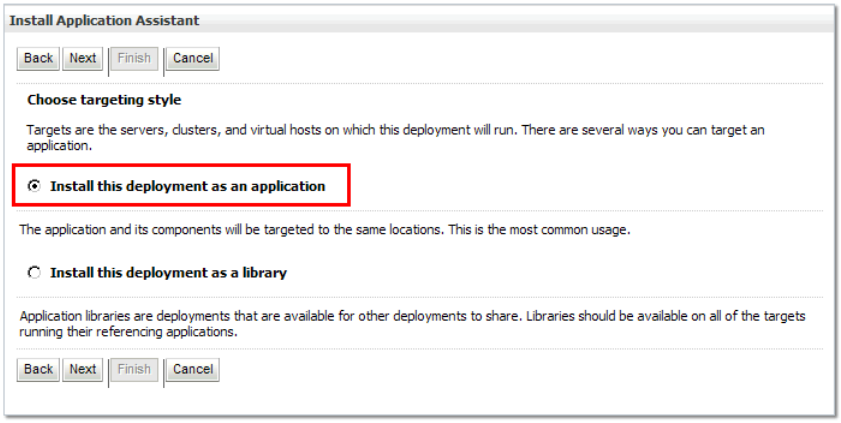
A deployment plan is not needed.

6. Click **Next**. The next installation page is displayed.



7. Select the `pci_services_weblogic.ear` file from the list.

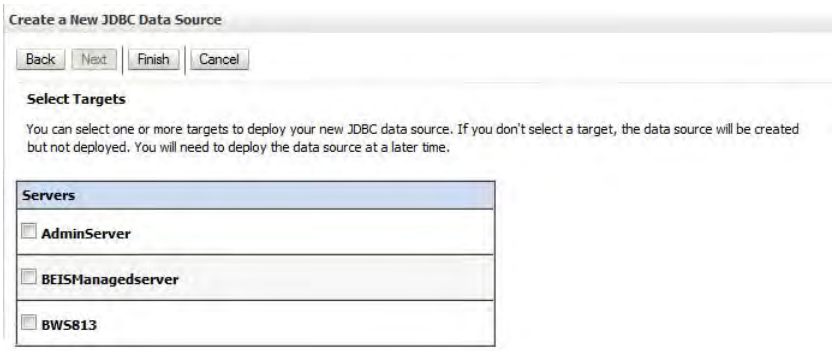
8. Click **Next**. The next installation page is displayed.



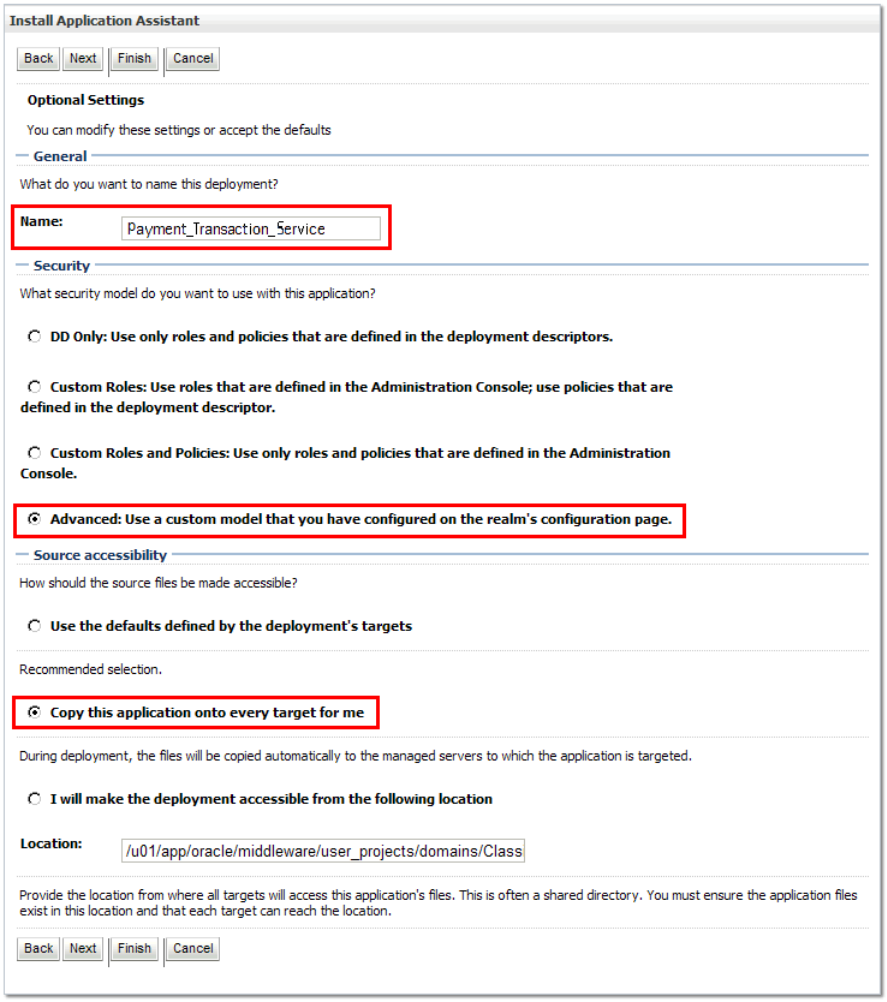
9. Select **Install this deployment as an application**.

10. Click **Next**. The next installation page is displayed.

    10.1. In some domains, the following page is displayed. Select the server where the application should be deployed and click **Next** to display the next installation page. Then go to step 11.



10.2. In some domains, the preceding page is skipped and the next installation page is displayed. Go directly to step 11.

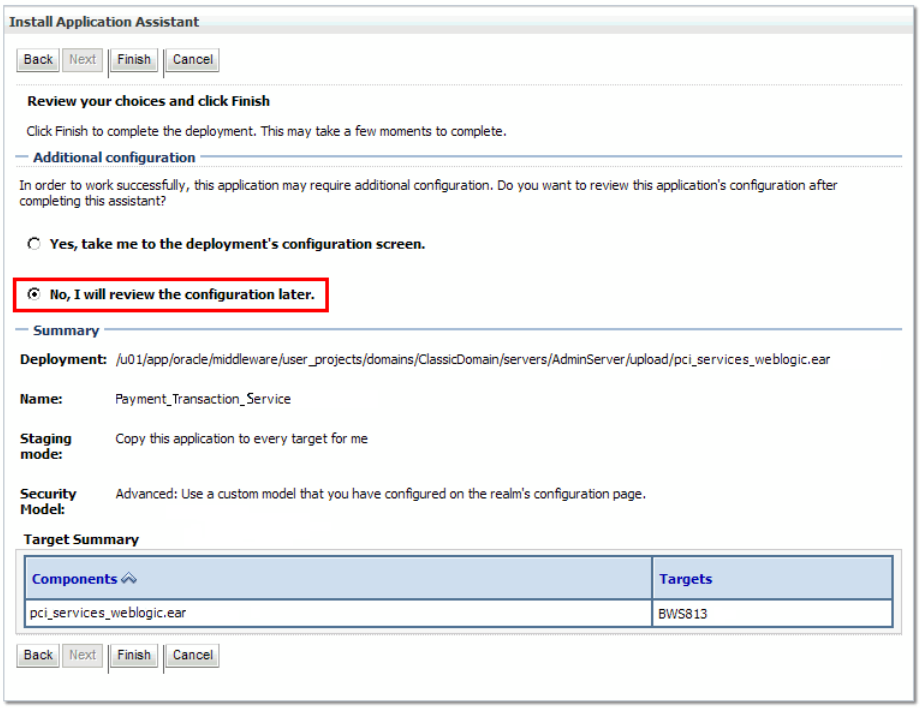


11. Enter a name for the application (for example, *Payment_Transaction_Service*) in the **Name** field.

12. Select **Advanced: Use a custom model that you have configured on the realm's configuration page**.

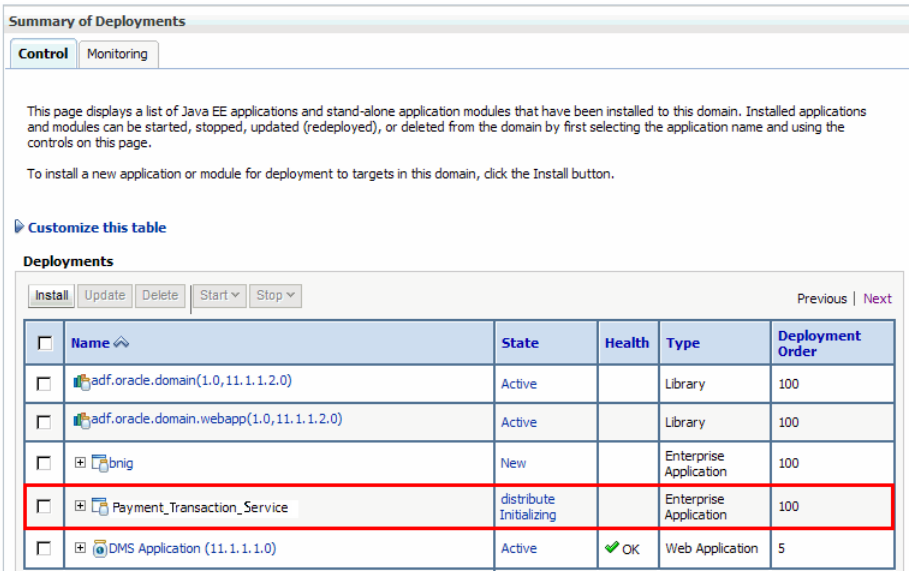
13. Select **Copy this application onto every target for me**.

14. Click **Next**. The next installation page is displayed.



15. Select **No, I will review the configuration later**.

16. Click **Finish** to start the deployment. When deployment is completed, the Summary of Deployments page is redisplayed with the newly deployed adapter.

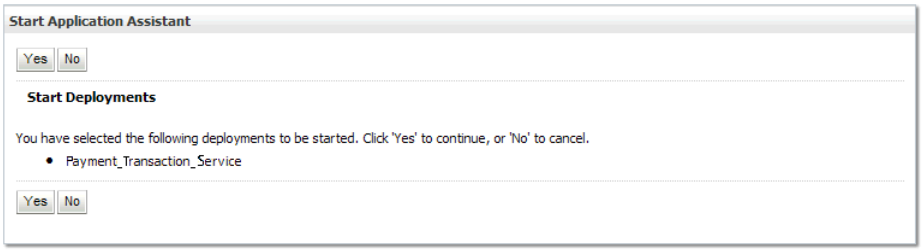


17. In the Change Center pane, click **Activate Changes**.

18. Start the newly deployed application as follows:

18.1. Select the newly deployed adapter.

18.2. Click **Start -> Servicing all requests**. The Start Application Assistant page is displayed.



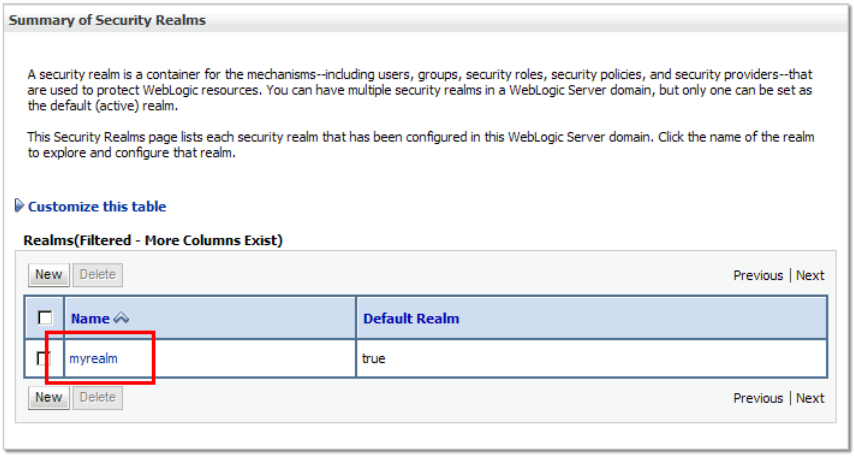
18.3. Click **Yes**.

Step 5 Configure the security group and user

Use the following steps to add the `bannerwsGroup` group and an administrative user to the adapter. This group and user protect the defined endpoint.

1. In the Domain Structure pane, click **Security Realms**.

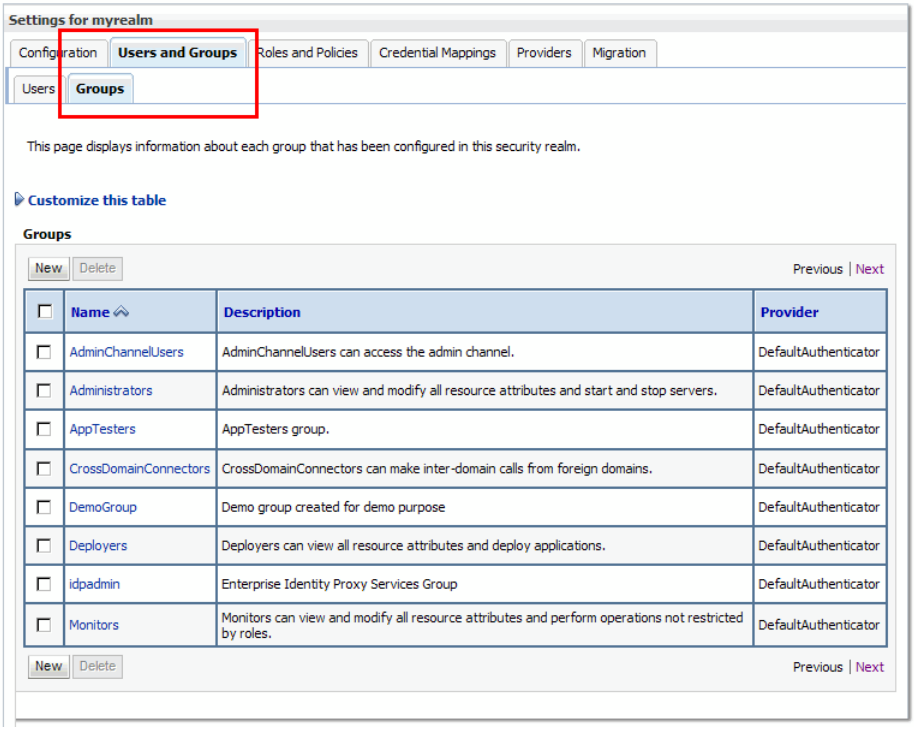
The Summary of Security Realms page is displayed.



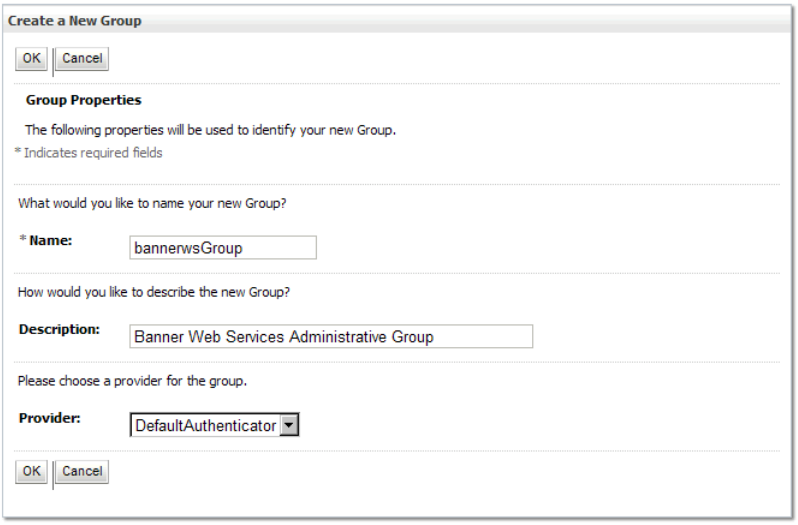
2. Click **myrealm**. The Settings for myrealm page is displayed.

3. Select the **Users and Groups** tab.

4. Select the **Groups** sub-tab. A table of existing groups is displayed.



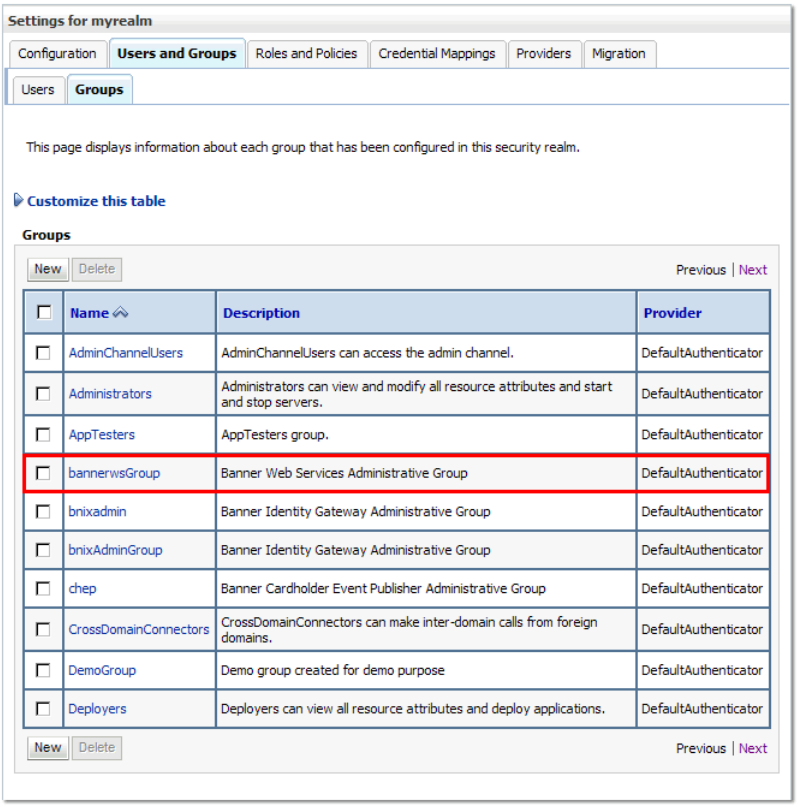
5. Click **New**. The Create a New Group page is displayed.



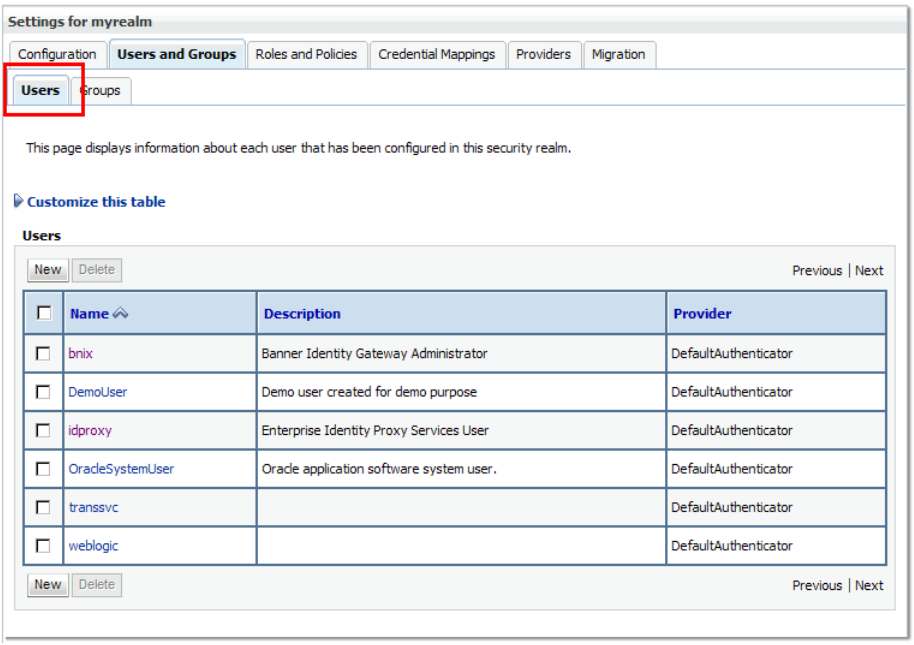
6. Enter the following information to create a group:

Name	*bannerwsGroup*
Description	*Banner Web Services Administrative Group*
Provider	*DefaultAuthenticator*

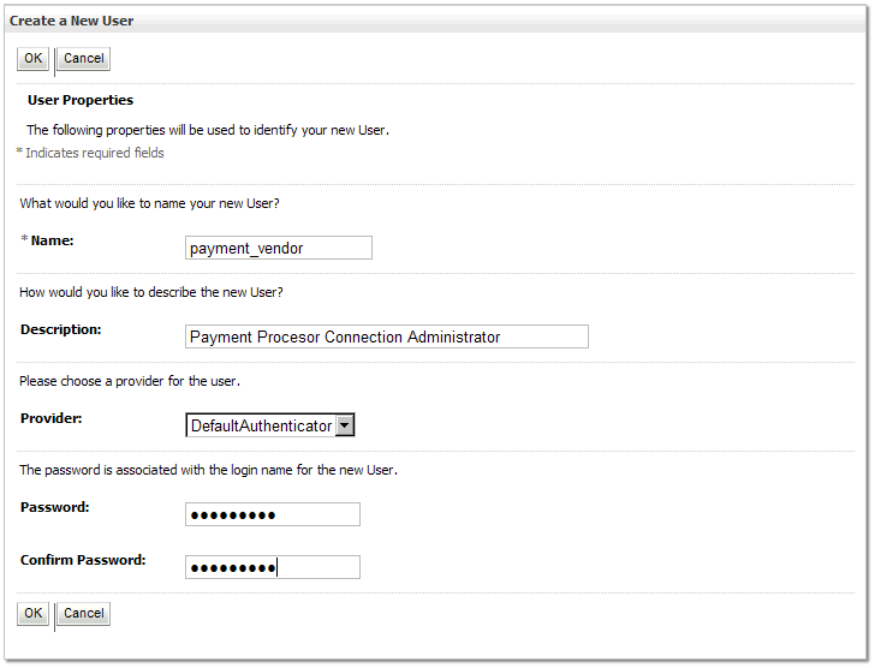
7. Click **OK**. The table of groups is redisplayed with the new group.



8. Select the **Users** sub-tab. A table of existing users is displayed.



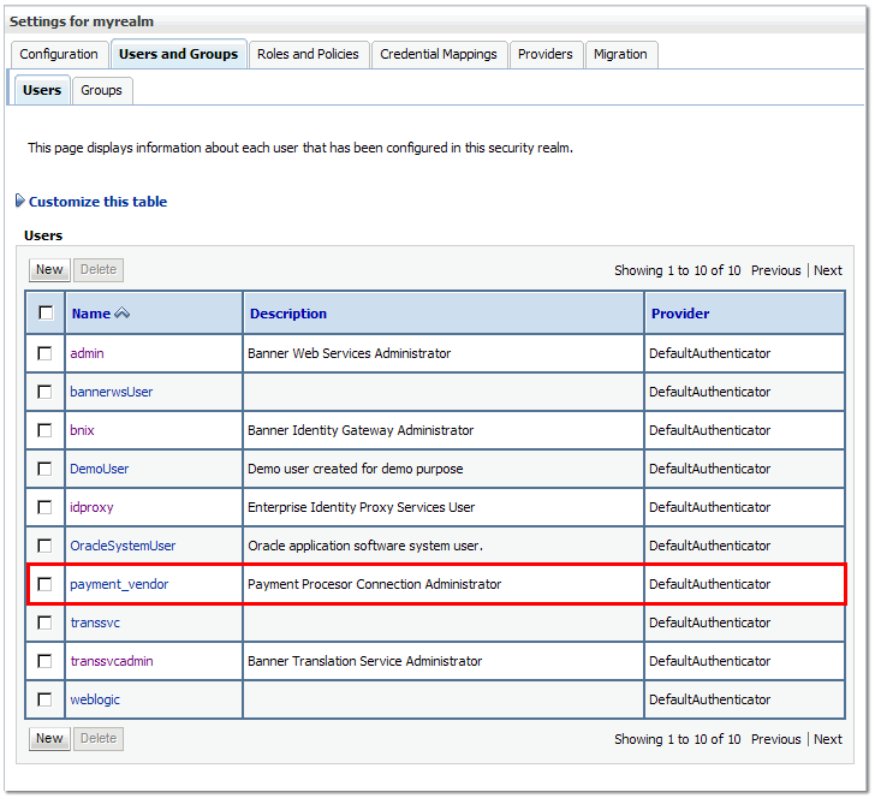
9. Click **New**. The Create a New User page is displayed.



10. Enter the following information to create a user:

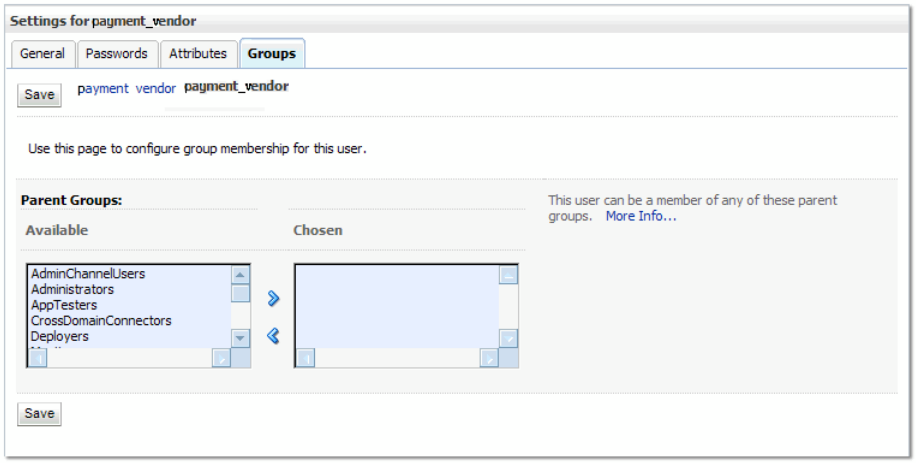
Name	*payment_vendor* (This is an example. Enter the name of your choice.)
Description	*Payment Processor Connection Administrator*
Provider	*DefaultAuthenticator*
Password	Password for the user being created
Confirm Password	Confirmation of the password

11. Click **OK**. The table of users is redisplayed with the new user.

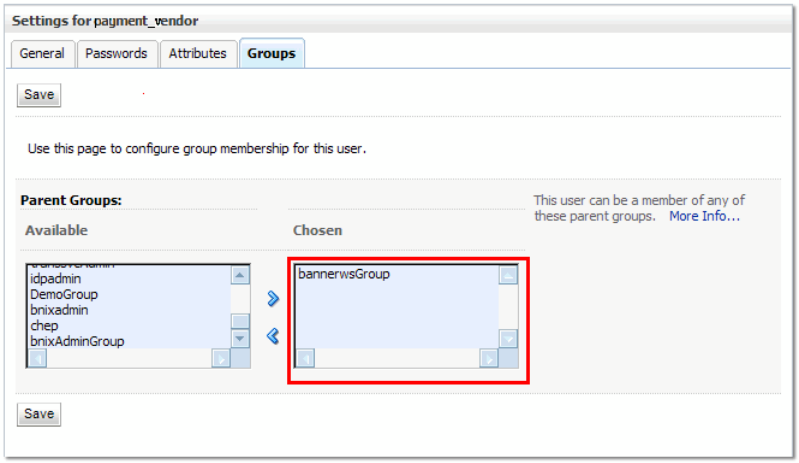


12. Click the name of the user that you just created. The Settings page for the user is displayed.

13. Select the **Groups** tab.



14. In the Parent Groups section, select *bannerwsGroup* in the **Available** list and move it to the **Chosen** list.

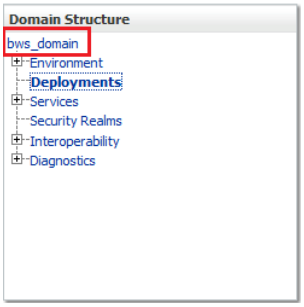


15. Click **Save**.

Step 6 Enable schema validation (optional)

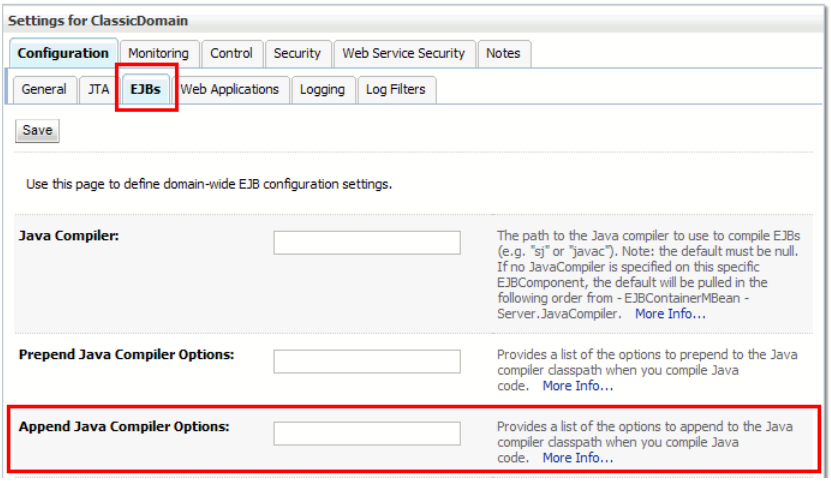
Validating XML request and response messages for each Web service invocation degrades system performance. For this reason, schema validation is turned off by default. To enable schema validation, you must set system property `BANNERWS_SCHEMA_VALIDATION` with a value of *true* for the Managed Server where the adapter is installed. Use the following steps to enable schema validation.

1. In the Domain Structure pane, click the name of the domain.



   The Settings page is displayed.

2. Select the **EJBs** tab.



3. Add the following value in the **Append Java Compiler Options** field:

   ```
   -DBANNERWS_SCHEMA_VALIDATION=true
   ```

4. Click **Save**.

5. Restart the server for the changes to take effect.

Step 7 Verify the deployment

Use the following steps to verify that the adapter is deployed successfully.

1. Use a Web browser to access the URL mapped in the following URL:

   ```
   <http or https>://<host>:<port>/pci/
   ```

   This URL is used to access an information page, not the endpoint.

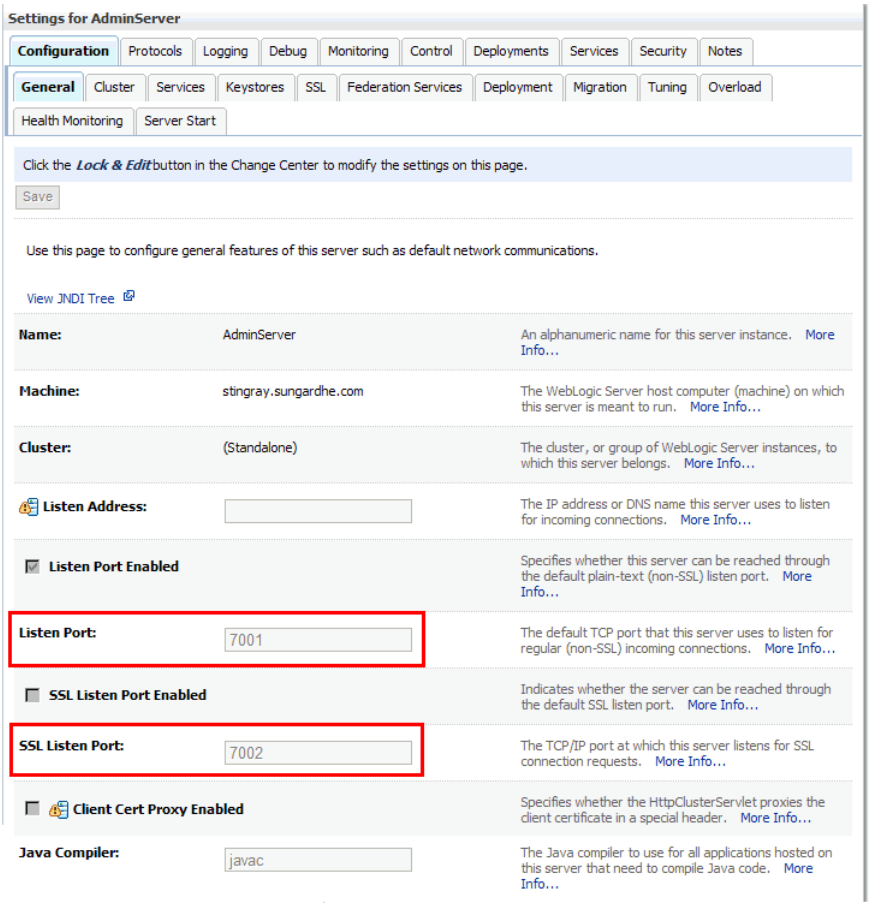
2. Log in with the name and password configured in Step 5, "Configure the security group and user". An information page for the adapter is displayed. This page shows the version of the adapter and provides a link to the Web service's WSDL.

3. Click the link on the information page to display the associated WSDL. The `<soap:address location>` attribute under the `<service>` tag at the bottom of the WSDL identifies the URL that you should always use to invoke the Web service.

Example:

```
<soap:address location="http(s)://<middle tier
host>.school.edu:<port>/pci/v1_0"/>
```

In some situations, you can access the home page via a browser, but payment transactions are not logged in the `pci_ws.log` or in the redirect log for the container where the adapter is installed. This may be due to a wrong port configuration. The Web service uses the Listen Port for http protocols and the SSL Listen Port for https protocols. These ports are configurable, by institution, via the Oracle WebLogic Server Administration Console and are displayed on the General tab of the Settings for Managed Server page:



Web service operations

After a payment processing vendor processes a payment transaction, the vendor invokes the PaymentTransaction Web service via the URL that is exposed by the adapter. This Web service has two operations:

- If the payment transaction was authorized, the *AddAccountTransaction* operation updates Banner with the results of the payment transaction.
- If the payment transaction failed, the *ReportTransactionError* operation provides Banner with the reason why the payment transaction failed.

Both operations can be invoked by sending the appropriate request message to the URL that is exposed by the adapter. The URL is similar to the following:

```
<protocol>://<host>:<port>/<context root>/v1_0
```

Components of the URL are defined as follows:

Component	Description	Default	Example
`protocol`	Protocol used to invoke the Web service. Valid values are *http* and *https*.	*http*	*http*
`host`	Server where the adapter is deployed	None	*appserv101.greatvalleyu.com*
`port`	Location where the adapter can receive messages. For example, the default port for http requests is 80.	None	*7019*
`context root`	Root path of URLs that the adapter handles. This can only be changed on OAS 10.1.3.x servers when the adapter is deployed.	*/pci*	*/pci/prod*

Example

```
http://appserv101.greatvalleyu.com:7019/pci/prod/v1_0
```

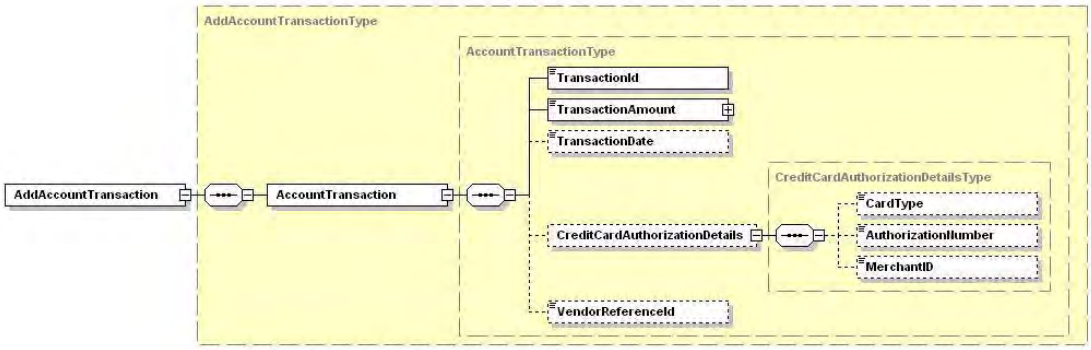
AddAccountTransaction operation

The AddAccountTransaction operation allows payment processing vendors to update Banner with the results of a successful payment authorization. This operation uses a request/reply exchange of the following messages using the SOAP protocol over HTTP:

- AddAccountTransation
- ConfirmAddAccountTransaction

AddAccountTransaction

A payment processing vendor uses the AddAccountTransaction message to request the update of a payment transaction initially created in Banner. The following diagram shows the structure of the AddAccountTransaction message schema:

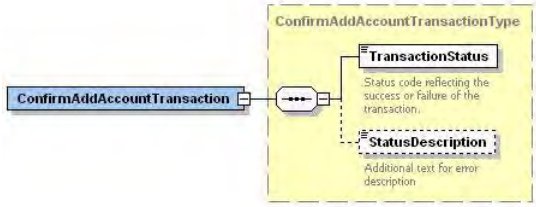


ConfirmAddAccountTransaction

ConfirmAddAccountTransaction is the response message of the operation. The response can be a success or failure:

- If the TransactionId provided in the request is found and the TransactionAmount provided in the request matches the original transaction amount, the application update process is called. If the update is successful, then the operation returns a TransactionStatus element with the value *Success*.
- If the TransactionId is not found or if the TransactionAmount is different than the original transaction amount, the operation returns a TransactionStatus element with the value *Failure*. In addition, a StatusDescription element contains a textual description of the error.

The following diagram shows the structure of the ConfirmAddAccountTransaction message schema.



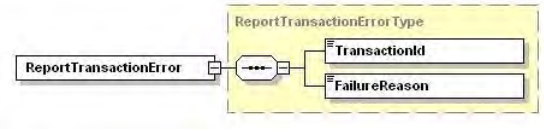
ReportTransactionError operation

The ReportTransactionError operation allows payment processing vendors to provide Banner with the results of a failed payment authorization. The type of failure depends on the payment processing vendor. In some cases a failure can occur if the card number or expiration date is invalid. In most cases, however, this situation results in a transaction cancellation instead of an error. The ReportTransactionError operation uses a request/reply exchange of the following messages using the SOAP protocol over HTTP:

- ReportTransactionError
- ConfirmReportTransactionError

ReportTransactionError

A payment processing vendor uses the ReportTransactionError message to report the failed authorization of a payment and to update the account transaction initially created in Banner. The following diagram shows the structure of the ReportTransactionError message schema.

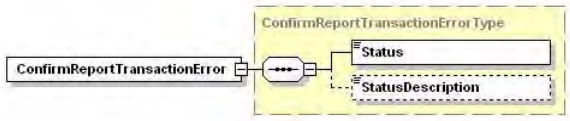


ConfirmReportTransactionError

When an error is reported, the operation returns the ConfirmReportTransactionError message. The message indicates one of the following, depending on whether the TransactionId in the request is found:

- If the TransactionId is found, the operation returns a Status element with the value *Acknowledged*.
- If the TransactionId is not found, the operation returns a Status element with the value *Error*. In addition, a StatusDescription element contains a textual description of the error.

The following diagram shows the structure of the ConfirmReportTransactionError message schema.



SOAP fault messages

If the Banner Payment Transaction Service Adapter has trouble processing an inbound request, a SOAP fault is raised. Situations that might cause a SOAP fault message include the following:

- The inbound request does not conform to the XML schema definition.
- A network, database, or other technical issue occurs.

Message mapping to Banner

The following tables provide a mapping between the message elements/attributes and Banner columns. The left vertical lines represent the nesting of the attributes inside the elements. Elements can nest inside other elements as well.

AddAccountTransaction

Element/Attribute	Database Mapping
`AddAccountTransaction`	
`AccountTransaction`	
`TransactionId`	Validated against `GORCCAU_PAY_TRANS_ID`
`TransactionAmount`	Validated against `GORCCAU_AMOUNT`

Element/Attribute	Database Mapping
TransactionDate	
CreditCardAuthorizationDetails	
CardType	GORCCAU_DETAIL
AuthorizationNumber	GORCCAU_VENDOR_AUTH_CODE
MerchantID	GORCCAU_MERCHANT_ID
VendorReferenceId	GORCCAU_VENDOR_REFER_NO

ConfirmAddAccountTransaction

Element/Attribute	Database Mapping
ConfirmAddAccountTransaction	
TransactionStatus	Generated/derived
StatusDescription	GORCCAU_VENDOR_ERROR_MSG

ReportTransactionError

Element/Attribute	Database Mapping
ReportTransactionError	
TransactionId	Validated against GORCCAU_PAY_TRANS_ID
FailureReason	GORCCAU_VENDOR_ ERROR_MSG

ConfirmReportTransactionError

Element/Attribute	Database Mapping
ConfirmReportTransactionError	
Status	Generated/derived
StatusDescription	Generated/derived

Setup requirements

The PaymentTransaction Web service uses the following data that must be set up in Banner.

Payment card types

Various payment cards can be used within Banner (for example, American Express, Visa, and MasterCard). You must cross-reference the payment card codes stored in Banner to the payment card codes sent by your payment processing vendor. Refer to "Define payment card types" on page 3-9 for setup instructions.

Accounting information

You must set up the various combinations of process code, payment card code, system code, and merchant ID that your institution uses. These combinations are used to assign a detail code and cashier ID to payment transactions. Refer to "Define accounting information" on page 3-10 for setup instructions.

3 Common Implementation

This chapter gives instructions for setting up the following options that are common to payment card processing throughout Banner®:

- Processes that use payment card processing (for example, admissions applications, registration fees, and gifts).
- Source code for writing account detail records.
- Printer code for printing receipts.
- Address hierarchy that determines which address is stored for payments processed for admissions application fees, transcript request fees, and enrollment verification fees.
- Merchant IDs that tell the payment processing vendor which payment profile to use. Multiple merchant IDs can be set up to simplify the settlement process for institutions that must separately settle payment card transactions based on level, campus, or college.
- Types of payment cards that can be used for payments (for example, Visa and MasterCard).
- Accounting information for each combination of process code, payment card code, system code, and merchant ID.
- Parameters that support connection with the payment processing vendor.
- Transaction descriptions that are sent to the payment processing vendor.
- Success and failure URLs used by the payment processing vendor to redirect the browser after payments are processed.

Implementation steps

The following steps are used to set up the options that are common to payment card processing:

- Step 1, “Define processes that use payment card processing”
- Step 2, “Define source code”
- Step 3, “Define printer code”
- Step 4, “Build address hierarchy”

- Step 5, “Define default merchant IDs”
- Step 6, “Enable multiple merchant ID option”
- Step 7, “Build multiple merchant ID hierarchy”
- Step 8, “Define payment card types”
- Step 9, “Define accounting information”
- Step 10, “Define Banner Web Tailor parameters”
- Step 11, “Define payment transaction descriptions”
- Step 12, “Provide success and failure URLs to payment processing vendor”

Additional product-specific setups are described in the remaining chapters of this handbook.

Step 1 Define processes that use payment card processing

The Process Name Validation Form (GTVPROC) identifies the Banner processes that use payment card processing at your institution. GTVPROC process codes can be used as merchant IDs on outgoing transactions to tell the payment processing vendor which Banner process produced a payment transaction.

Note

Your payment processing vendor might provide different codes to use for merchant IDs. ■

Use the following steps to identify the processes that use payment card processing at your institution.

1. Access the Process Name Validation Form (GTVPROC).

2. Define the process codes for the processes used at your institution:

Code	Description
FLEXREGCCREGFEES	Flexible Registration Credit Card Registration Fees Process
WEBCCALUGIFT	Web Credit Card Advancement Gift Process
WEBCCAPPFEES	Web Credit Card Application Fees Process
WEBCCEPRTREQ	Web Credit Card Enrollment Verification Charge
WEBCCGRADAPP	Web Credit Card Graduation Application Process
WEBCCREGFEES	Web Credit Card Registration Fees Process
WEBCCTRANSREQ	Web Credit Card Transcript Request Process

3. Save.

Step 2 Define source code

Create a GTVSDAX rule that defines the source code that is used for payment card transactions made in Banner. This source code is used when account detail records are written.

Internal Code	*PMTSRCE*
Sequence	none
Group	*PAYMENTVENDOR*
External Code	Source code from TTVSRCE
Description	*Source Code for Web Payment*
System Required	selected

Step 3 Define printer code

Create a GTVSDAX rule that specifies the printer code that is assigned to Banner payment card transactions. This code identifies Banner payment card transactions in the receipt collector table and controls the printing of hardcopy receipts.

Internal Code	*PRINTERDEF*
Sequence	none
Group	*PAYMENTVENDOR*
External Code	Printer code from GTVPRNT
Description	*Printer Definition for Web CC*
System Required	selected

Step 4 Build address hierarchy

This step applies only for payments for admissions application fees, transcript request fees, and enrollment verification fees. Address information can be automatically collected from an individual's address records and stored for a payment. An address hierarchy determines which address is collected for a payment. Create the following GTVSDAX rules to build an address hierarchy.

Internal Code	*WEBCCADDR*
Sequence	Sequence numbers that prioritize records in the hierarchy. Lowest number is the highest priority.
Group	*ADDRESS*
External Code	Address type code from STVATYP (for example, *PR* for permanent address)
Description	*Web CC Address Hierarchy*
System Required	cleared

This hierarchy provides the order in which address type codes are used to collect address information for payment card transactions. The system first checks to see if the person has an active address for the highest priority address. If there is none, the second address is checked, and so on. If no address matches the codes in the hierarchy, address information is not stored for the payment card transaction.

To disable the automatic population of address fields for all users, enter a nonexistent code in the **External Code** field (such as *XX*) for all records in the hierarchy. If no such address records exist, address information is not populated automatically.

Step 5 Define default merchant IDs

Merchant IDs are used on outgoing payment transactions to tell the payment processing vendor which payment profile should be used. Merchant IDs can be GTVPROC process codes or codes that your payment processing vendor provides.

Note

Some Banner products allow you to build a hierarchy of multiple merchant IDs to simplify the settlement process for payment card transactions. This is valuable if you must separately settle payments based on level, campus, or college. ■

You must define a system-wide default merchant ID plus product-specific defaults for the Banner products implemented at your institution. Default merchant IDs are used in the following situations:

- If your institution does not use multiple merchant IDs
- If the multiple merchant ID hierarchy does not provide a matching merchant ID
- If required information (such as term) is missing

If a payment cannot be routed based on the multiple merchant ID hierarchy or default merchant ID, then the payment is processed with *0* as the merchant ID.

Create the following GTVSDAX rules to define the default merchant IDs.

Internal Code	*DEFAULT*
Sequence	none
Group	*WEBAPPCCID* Default for Banner Student Self-Service admissions *WEBSTUCCID* Default for Banner Student Self-Service and Banner Flexible Registration *WEBALUCCID* Default for Banner Advancement Self-Service *WEBDEFCCID* System-wide default
External Code	Default merchant ID. This ID can be a GTVPROC process code or a code that your payment processing vendor provides. Any merchant ID can be used as a default, regardless of whether it is used otherwise.
Description	Description of the rule for easy identification (for example, *Default Appl CC Merchant ID*).
System Required	cleared

Example

Internal Code	*DEFAULT*
Group	*WEBSTUCCID*
External Code	*0*

Merchant ID 0 is the default for Banner Student Self-Service and Banner Flexible Registration. This ID is used for any payment that cannot be routed according to the multiple merchant ID hierarchy.

Step 6 Enable multiple merchant ID option

You can optionally build a hierarchy of multiple merchant IDs to simplify the settlement process for payment card transactions. This is valuable if you must separately settle payments based on level, campus, or college.

Note

The multiple merchant ID option is *not* available for Banner Advancement Self-Service. It only applies to Banner Student Self-Service and Banner Flexible Registration. ■

Create the following GTVSDAX rules to enable the multiple merchant ID options. One rule is used to enable the option for Banner Student Self-Service admissions payments.

Another rule is used to enable the option for all other Banner Student Self-Service payments and Banner Flexible Registration payments.

Internal Code	*USEAPPMMID*	Multiple merchant ID option for Banner Student Self-Service admissions payments
	USESTUMMID	Multiple merchant ID option for all other Banner Student Self-Service payments and Banner Flexible Registration payments
Sequence	none	
Group	*PAYMENTVENDOR*	
External Code	*Y*	Multiple merchant IDs can be used.
	N	Only one merchant ID is used.
Description	*Use Multiple Merchant IDs*	
System Required	cleared	

Step 7 Build multiple merchant ID hierarchy

If multiple merchant IDs are enabled, you can build a multiple merchant ID hierarchy. The hierarchy determines which merchant IDs are sent to the payment processing vendor. You can build one hierarchy for Banner Student Self-Service admission application payments. You can build another hierarchy for all other Banner Student Self-Service payments and Banner Flexible Registration payments.

A hierarchy is based on level, campus, or college. You have a great deal of flexibility in how you set up the hierarchy. For example, you can allow students from one campus to pay with American Express, but not students from other campuses.

Note

If your institution uses only one merchant ID, you do not need to build a hierarchy. You only need to set up default merchant IDs. See "Define default merchant IDs" on page 3-4 for details. ■

Create the following GTVSDAX rules to define a hierarchy of merchant IDs.

Internal Code	Criteria to be matched *(LEVEL, CAMPUS,* or *COLLEGE)*
Sequence	Sequence number used to match and prioritize records in the hierarchy. All rules in a set must have the same sequence number. Each set must have a different sequence number. Lowest number is the highest priority.

Group	*WEBAPPCCID* Hierarchy for Banner Student Self-Service admissions payments *WEBSTUCCID* Hierarchy for all other Banner Student Self-Service payments and Banner Flexible Registration payments
External Code	Merchant ID that identifies the profile used for the associated payment type. This ID can be a GTVPROC process code or a code that your payment processing vendor provides. All rules in a set must have the same external code. More than one set of rules can have the same external code.
Description	*Merchant ID for CC*
Translation Code	Value associated with the internal code. For example, if **Internal Code** equals *CAMPUS*, then **Translation Code** would equal a specific campus code. The **External Code** would define the merchant ID for the specific campus code.
System Required	cleared

To find a match, a student's data is compared to the record sets. The greatest number of matches without a divergence (non-match), regardless of priority, determines the merchant ID. When two or more sets have the same number of matches, the priority is used to determine the merchant ID. The highest priority (lowest sequence number) determines the merchant ID.

A null in a student's data does not match a null in the record set. A null neither disqualifies nor qualifies a match.

Example 1

	Rule 1	Rule 2
Internal Code	LEVEL	CAMPUS
Sequence	1	1
Group	WEBSTUCCID	WEBSTUCCID
External Code	4	4
Translation Code	UG	M

These two rules form a set because the sequence numbers are equal and the external codes are equal. More than one set may have the same external code, but within a set all records must be the same. Together, the rules route all payments in which the student's level is *UG* (undergraduate) and campus is *M* (main) to merchant ID *4*.

If a student's level (in the SGASTDN general student record) were *UG* but the campus did not exist (null), the payment would still be routed using external code *4*.

If another set of records, however, has more matching criteria, the payment would be routed using the merchant ID of the set that matches the most criteria without diverging on any. If two sets of records fit the student with an equal number of matching criteria, the set with the higher priority (lowest sequence number) would be used.

Example 2

	Rule 1	Rule 2
Internal Code	LEVEL	COLLEGE
Sequence	2	2
Group	WEBSTUCCID	WEBSTUCCID
External Code	6	6
Translation Code	UG	AS

Consider all of the rules in examples 1 and 2.

A payment for a student with level *UG*, campus null, and college *AS* would be routed using merchant code *6*. Two criteria match in the second set of rules; one criterion matches in the first set.

A payment for a student with level *UG*, campus *M*, and college *AS* would be routed using subcode *4*. Both sets of records match two criteria, but the first set has a higher priority (lower sequence number).

A payment for a student with level *UG*, campus *NE*, and college *AS* would be routed using merchant code *6*. The first set of records does not match the campus; the second set matches two criteria. If a third set matched all three criteria, then that set would be used.

Example 3

	Rule 1	Rule 2	Rule 3
Internal Code	LEVEL	CAMPUS	COLLEGE
Sequence	3	3	3
Group	WEBSTUCCID	WEBSTUCCID	WEBSTUCCID
External Code	7	7	7
Translation Code	UG	M	BS

Consider all of the rules in examples 1, 2, and 3.

A payment for a student with level *UG*, campus *M*, and college *BS* would be routed using merchant code *7*.

A payment for a student with level *UG*, campus null, and college *BS* also would be routed using merchant code 7 because the second example diverges on college.

Example 4

	Rule 1	Rule 2	Rule 3
Internal Code	LEVEL	CAMPUS	COLLEGE
Sequence	4	4	4
Group	WEBSTUCCID	WEBSTUCCID	WEBSTUCCID
External Code	8	8	8
Translation Code	UG	SW	AS

Consider all of the rules in examples 1, 2, 3, and 4.

A student with level *UG*, campus *SW*, and college *AS* potentially matches the rules in example 2 and example 4. The payment would be routed using merchant code *8* because this set matches three criteria. The set in example 2 matches only two criteria.

A payment for a student with level *UG*, campus *null*, and college *AS* would be routed using merchant ID *6*. The priority of the rules in example 2 exceeds the priority of the rules in example 4.

While campus is not used as a criteria in example 2, all campus values are possible matches (including null). However, a null value is not considered to be an exact match. If the priorities in examples 2 and 4 had been reversed, the outcome would have been reversed.

Example 5

	Rule 1
Internal Code	LEVEL
Sequence	5
Group	WEBSTUCCID
External Code	3
Translation Code	GR

Consider all of the rules in examples 1, 2, 3, 4, and 5.

A payment for a student with level *GR*, campus *M*, and college *AS* would be routed using merchant ID *3*. Data diverges from all other sets on level. One match is enough when all other sets diverge.

Step 8 Define payment card types

Various payment cards can be used within Banner. Examples include American Express, Visa, and MasterCard. Payment card codes stored in Banner must be cross-referenced to the payment card codes sent by your payment processing vendor.

Example

The payment processing vendor uses the code *DISCOVER*, which cross-references to the Banner code *DISC*.

Use the following steps to identify the payment cards that can be used for payments at your institution.

1. Confirm the payment card codes that your payment processing vendor uses.
2. Access the Credit Card Validation Form (GTVCCRD).
3. Enter the following information for each payment card that can be used at your institution:

Code	Payment card code that is stored in Banner. Examples include AMEX and VISA.
External Merchant ID	Payment card code that is sent by the payment processing vendor
Description	Description of the payment card

4. Save.

Step 9 Define accounting information

The Credit Card Merchant ID Form (GOAMERC) is used to identify each combination of process code, payment card code, system code, and merchant ID (third party transaction ID) that your institution uses. For each combination, you must define the associated payment detail code and cashier ID used in the settlement process.

When your payment processing vendor returns a payment transaction, the GOAMERC record that matches on process code, payment card code, system code, and merchant ID is used to determine the accounting information that is assigned to the transaction and inserted into Banner.

Use the following steps to set up records on GOAMERC.

1. Access the Credit Card Merchant ID Form (GOAMERC).

Note

The key block is used to enter search criteria for displaying information in the next block. Any or all fields in the key block can be left blank. ■

2. Go to the Credit Card Merchant ID block.

3. For every combination of process code, payment card code, system code, and merchant ID, insert a new record with the following information:

Process	Banner process that uses payment processing (for example, *WEBCCREGFEES*). Codes are defined on GTVPROC.
Credit Card	Payment card that is used with payment processing (for example, *VISA*). Codes are defined on GTVCCRD.
System	Banner system: *A* Banner Advancement *S* Banner Student or Banner Flexible Registration
Third Party Transaction	Merchant ID that identifies the profile used for payment transactions. IDs must be coordinated with your payment processing vendor and defined on GTVSDAX.
Detail	Payment detail code used when payments are inserted into Banner. • Banner Student - Enter the detail code used in Banner Accounts Receivable for the process, system, payment card type, and merchant ID. • Banner Advancement - Enter the gift type for the process, system, payment card type, and merchant ID.
Voice Response Message	Message number that is spoken to a Banner Voice Response user to identify the payment card type. Optional.
Active	Check box that indicates if the record is active.
Cashier ID	User ID for the cashiering session when payments for the process, system, payment card type, and merchant ID are inserted into Banner. The user must be a valid Oracle username. If the **Restricted User** check box is selected for this user on the User Profile Definition Form (TGAUPRF), that person cannot post payments to Banner unless he or she is given permission for the detail codes. If you use User Profile security, cashier entries must also be created and permissions must be granted as needed.

4. Save.

Step 10 Define Banner Web Tailor parameters

Two parameters in the Banner Web Tailor Web Parameters Table (TWGBPARM) support payment card processing. Use the following steps to define these parameters.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Web Tailor Parameters.
4. For each of the following parameters, click the parameter name, enter the parameter value, and click **Submit Changes**.

Parameter	Description
PAYVEND_TRANS_TIMEOUT	Number of minutes after which the payment processing vendor's Web site "times out" if there is no activity. An update received from the vendor after this time limit is treated as a "transaction not found." An error message indicates the current date/time, transaction date/time, and expiration date/time.
PAYVEND_URL	Payment processing vendor's URL. The user's browser is redirected from Banner to this URL to complete a payment transaction. Example: `http://my.pci.com:9999/gateway` Only one vendor can be active at a time.

Step 11 Define payment transaction descriptions

Information text defined in Banner Web Tailor is used as the transaction description that is sent to the payment processing vendor with a payment transaction. This transaction description depends on the process (donation, application fee, enrollment verification, graduation application, registration fee, or transcript request) that is being performed. Use the following steps to verify and customize information text, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Information Text.

4. Verify and customize, if necessary, the information text labels for `gokpven.f_process_payment_updates`. This is the baseline configuration.

Seq #	Label	Information Text
1	DEFAULTDESCRIPTION	Sungard HE University
1	WEBCCALUGIFT	Alumni Donation
1	WEBCCAPPFEES	Application Fees
1	WEBCCEPRTREQ	E Print Request
1	WEBCCGRADAPP	Graduation Application
1	WEBCCREGFEES	Registration Fees
1	WEBCCTRANSREQ	Transcript Request

Step 12 Provide success and failure URLs to payment processing vendor

After the payment processing vendor processes a payment, the vendor redirects the browser to a success or failure URL in Banner, depending on whether the transaction update is a success or failure. You must provide the success and failure URLs to your payment processing vendor. Use the following structures to determine your success and failure URLs:

Success URL	`<http or https>://<Banner server>/gokpurl.p_payment_success_return`
Failure URL	`<http or https>://<Banner server>/gokpurl.p_payment_failure_return`

For Banner Self-Service payments, the protocol specified in these URLs must be consistent with the protocol used for your Banner Self-Service login URLs. For example, if the redirect from the payment processing vendor to Banner must be via SSL, then Banner Self-Service sessions must be started via SSL. Conversely, if Banner Self-Service sessions are started without SSL (http rather than https), then the redirect from the payment processing vendor to Banner Self-Service must be http. This consistency ensures that the SESSID cookie is found and that the user is taken directly to the proper Banner Self-Service page without requiring the user to log in.

Implementation examples

Banner uses the process code, merchant ID, and payment card type to determine the accounting information that is stored for a payment transaction. The following examples show how these relationships are set up.

Example 1

Form	Setup
GTVSDAX	Group = WEBAPPCCID External code = 4
GTVCCRD	Code = VISA External Merchant ID = APPVISA
GOAMERC	Process = WEBCCAPPFEES Credit Card = VISA Third Party Transaction = 4

A prospective student pays a $50 admissions application fee in Banner Student Self-Service. The process code is WEBCCAPPFEES (Web Credit Card Application Fees Process). The WEBAPPCCID rule on GTVSDAX identifies the merchant ID to be 4. Banner sends the merchant ID (4) to the payment processing vendor. The merchant ID tells the payment processing vendor which payment profile to use. The merchant ID returns the payment card type (APPVISA). Banner uses GTVCCRD to translate the payment card type to VISA. The process code (WEBCCAPPFEES), payment card type (VISA), and merchant ID (4) are used to identify the record on GOAMERC that defines the accounting information for the payment.

Example 2

Form	Setup
GTVSDAX	Group = WEBALUCCID External code = 3
GTVCCRD	Code = VISA External Merchant ID = ALUVISA
GOAMERC	Process = WEBCCALUGIFT Credit Card = VISA Third Party Transaction = 3

An alumnus makes a $500 donation in Banner Advancement Self-Service. The process code is WEBCCALUGIFT (Web Credit Card Advancement Gift Process). The WEBALUCCID rule on GTVSDAX identifies the merchant ID to be 3. Banner sends the merchant ID (3) to the payment processing vendor. The merchant ID tells

the payment processing vendor which payment profile to use. The vendor returns the payment card type (ALUVISA). Banner uses GTVCCRD to translate the payment card type to VISA. The process code (WEBCCALUGIFT), payment card type (VISA), and merchant ID (3) are used to identify the record on GOAMERC that defines the accounting information for the donation.

Objects used with payment card processing

Payment card processing uses the following objects.

Vendor Payment Transaction Audit Form (GOICCAU)

GOICCAU is used to query payment transactions by payor ID, transaction ID, reference number, and/or transaction date.

Key block

This block is used enter search criteria for displaying information in the next block. Any or all fields in the key block can be left blank. The key block displays the following fields:

Field	Description
ID	ID and name of the person or organization for whom payment card transactions are being queried. List - Person Search Form (SOAIDEN) Count Query Hits - Non-Person Search Form (SOACOMP) Duplicate Item - SSN/SIN Alternate ID Search Form (GUIALTI)
Transaction ID	Unique ID associated with the payment card transaction.
Ref No	Reference number returned from the payment processing vendor.
Date	Date of the payment card transaction.

Data block

This block displays payment card details based on the search criteria in the key block. The data block displays the following fields:

Field	Description
Transaction ID	Unique ID associated with the payment card transaction.

Note: Either the reference number or the authorization code can be used to identify the transaction if you need to communicate to your payment processing vendor about a payment transaction (for example, a refund).

Ref No	Reference number returned from the payment processing vendor.
Auth Code	Authorization code returned from the payment processing vendor.
Amount	Amount of the payment card transaction.
Pay Card	Payment card associated with the transaction (for example, Visa).
Merchant	Merchant ID used on the outgoing payment transaction to tell the payment processing vendor which payment profile should be used.
ID	ID of the user who requested the payment card transaction.
System	Banner product associated with the transaction: *A* Banner Advancement *S* Banner Student or Banner Flexible Registration
Date	Date of the payment card transaction.
Term	Term code associated with the payment card transaction.
Application	Admissions application sequence number associated with the transaction.
Gift No	Advancement gift number associated with the transaction.
Process	Banner process that produced the payment transaction. Codes are defined on the Process Name Validation Form (GTVPROC).
Location	Not used.
Sub Code	Merchant ID based on the calling application.
Status	Status of the payment card transaction.
Banner Status	Last status of the transaction from a Banner perspective.

Field	Description
Vendor Status	Last status of the transaction from a payment processing vendor perspective.
Vendor Message	Error message from the payment processing vendor for a failed transaction, if applicable.
Appl Data	Application-specific data for the transaction. This data depends on the application used to create the transaction (Banner Advancement or Banner Student). This field is not used for all transactions.
Update Function	Function that is called when the payment processing vendor returns an authorized transaction.
Success URL	URL to which the transaction is redirected, if the payment process succeeds.
Failure URL	URL to which the transaction is redirected, if the payment process fails.

Object Library (GOQOLIB)

The following record groups support payment card processing:

```
GTVCCRD_RG
GTVPROC_RG
G$_TBBDETC_RGD
G$_ATVGIFT_RGD
```

The following lists of values (LOVs) support payment card processing:

```
GTVCCRD_LOV
GTVPROC_LOV
G$_TBBDETC_LOVD
G$_ATVGIFT_LOVD
```

The following program units populate the LOVs:

```
G$_POPULATE_TBBDETC_RG
G$_POPULATE_ATVGIFT_RG
```

Tables

Payment card processing uses the following tables:

GORCCAU	Credit Card Audit Table
GORMERC	Credit Card Type by Merchant ID Table
GTVCCRD	Credit Card Type Validation Table
GTVPROC	Process Name Validation Table

Packages

Payment card processing uses the following packages.

BWGKCCRD

This package contains the `p_acknowledgement_page` procedure. This procedure displays a printable acknowledgement page with the following information:

- Name formatted with prefix and suffix
- Date and time of the payment
- Authorization code

For payments made with Banner Advancement Self-Service, the donor ID and gift number can also be displayed. The display of these optional items is controlled by your institution. Refer to Chapter 8, “Banner Advancement Self-Service Implementation” for details on displaying these optional items.

GOKFUNC

This package contains the following procedures:

Procedure	Description
`p_validate_cc_amount`	Validates the character amount in the `CHAR_AMT_IN_OUT` parameter string, returns the amount in dollars and cents character format, returns the amount as a numeric value, and returns a flag that indicates whether the entered amount is valid.
`p_match_mmid`	Determines the appropriate merchant ID, based on the multiple merchant ID hierarchy in GTVSDAX.

GOKPURL

This package contains the following procedures:

Procedure	Description
p_payment_failure_return	Enables the payment processing vendor to redirect failed transactions to the specific URL. (Called by the vendor after a failed transaction.)
p_payment_success_return	Enables the payment processing vendor to redirect successful transactions to the specific URL. (Called by the vendor after a successful transaction.)

GOKPVEN

This package contains the following procedures and functions:

Procedure/Function	Description
p_redirecturl	Calls the Banner Web Tailor function that redirects the user to the specified URL.
p_payment_url_return	Verifies the transaction ID, updates the vendor status on GORCCAU, and redirects the browser to the application-specific success or failure URL.
f_fetchwtparam	Calls the Banner Web Tailor function that fetches the Banner Web Tailor parameter value.
f_encode	Calls the Banner Web Tailor function that encodes special characters.
f_get_info	Reads the Banner Web Tailor TWGRINFO table and retrieves information text for the transaction description.
f_add_transaction	Calls the update process (f_process_payment_updates) for successful transactions. (Called by the Web service via a request from the payment processing vendor.)
f_vendor_failure	Calls the update process (f_process_payment_updates) for failed transactions. (Called by the Web service via a request from the payment processing vendor.)

Procedure/Function	Description
f_process_payment_updates	Verifies payment information passed by the payment processing vendor, updates GORCCAU, calls the application-specific update procedure, receives the results of the application-specific update procedure, and returns an update status. (Called by the f_add_transaction function.)
f_collect_payment_info	Collects payment information, inserts a GORCCAU record, and redirects the browser to the payment processing vendor. (Called by the application process.)

GOKSELS

This package contains the following procedure and cursors:

Procedure/Cursor	Description
p_get_merchant_ids	Retrieves and stores the criteria used to determine which merchant ID is included with the payment card transaction.
gorccrdc_row_by_ccrd_code_c	Retrieves a row from GORCCRD for the specified payment card code.
gormerc_row_by_multi_c and f_get_gormerc_row	Retrieves a row from GORMERC based on the specified merchant ID, process code, system code, and optionally payment card code.
c_retrieve_gtvsdax_row	Retrieves a GTVSDAX row based on the specified internal group, internal code, and sequence number.

GOKTABS

The merch_id_rec_type record type holds a single set of criteria for determining which merchant ID is included with a payment card transaction.

The merch_id_tab_type PL/SQL table of the record type merch_id_rec_type stores all sets of criteria that are used to determine which merchant ID is included with a payment card transaction.

GTKCCRD

This package contains a cursor that returns all rows containing the specified payment card code (GTVCCRD_CODE).

GTKPROC

This package contains a cursor that returns all rows containing the specified process code (GTVPROC_CODE).

API

The Payment API (gb_payment) standardizes payment card processing in Banner. This API includes the following packages:

:

```
gokb_payment0.sql
gokb_payment1.sql
gokb_payment_r0.sql
gokb_payment_r1.sql
gokb_payment_s0.sql
gokb_payment_s1.sql
gokb_payment_build.sql
gokb_payment_message.sql
gokb_gorccau0.sql
gokb_gorccau1.sql
gos_payment_seq.sql
```

4 Banner Student Implementation

Students can use payment cards to pay fees associated with the following types of Banner® Student Self-Service transactions:

- Admissions applications
- Enrollment verification requests
- Registration and student accounts
- Transcript requests
- Graduation applications

This chapter gives instructions for setting up Banner Student to support these payment card transactions.

Form setup for admissions applications

The following Banner Student forms support the use of payment cards to pay admissions application fees with Banner Student Self-Service:

STVWAIV	Application Fee Waiver Reason Validation Form
SAAWADP	Web Application Customized Lists Form
SAAADMS	Admissions Application Form
SAAETBL	Electronic Application Submitted Form
SAAWADF	Electronic Applicant Web Default Rules Form

Application Fee Waiver Reason Validation Form (STVWAIV)

This form is used to define the codes that describe application fee waivers and their associated discounts. The discount is subtracted from the specified application fee.

Applicants can select a waiver reason on the Select a Waiver page in Banner Student Self-Service if they are allowed to waive all or part of the fee. Waiver reasons may be specific to application types.

STVWAIV contains the following fields:

Field	Description
Code	Application waiver code that indicates why the application fee is waived.
Description	Description of the application fee waiver.
Discount Amount	Amount that is subtracted from the application fee.
Activity Date	Date when the record was created or last updated. Display only.

Web Application Customized Lists Form (SAAWADP)

This form is used to identify which race codes, interest codes, requested materials, test codes, and application waiver codes are displayed in the appropriate drop-down lists on Banner Student Self-Service pages, based on application type. For example, a graduate school can customize its Web admissions application to show only graduate-related test codes such as GMAT and GRE.

Values in the drop-down lists are populated from the SARWADP table. If no values exist on SARWADP, then the values come from the Web-enabled rows on the SORXREF table that have the corresponding label. If no Web-enabled rows exist on SORXREF, then the values come from the validation table (for example, STVMATL for requested materials).

Key block

This block contains the following fields:

Field	Description
Web Application Type	Type of Web admissions application. List - Web Application Code (STVWAPP)
Validation Table Name	Validation table that provides codes for the Web pull-down lists: *GORRACE* Race codes *STVINTS* Interest codes *STVMATL* Requested material codes *STVTESC* Test codes *STVWAIV* Application waiver codes If no table name is entered, all tables and codes defined for the specified admissions application type are displayed. List - EDI Verification Label Validation (STVXLBL)

Main block

This block contains the following fields:

Field	Description
Table	Validation table that provides codes for the Web drop-down lists. List - EDI Verification Label Validation (STVXLBL)
Code	Specific value from the corresponding validation table that should appear in the Web drop-down list for the admissions application type. List - Depends on the table
Description	Text that is displayed in the Web drop-down list.
User	User who created or last updated the record. Display only.
Activity Date	Date when the record was created or last updated. Display only.

Admissions Application Form (SAAADMS)

This form provides payment and fee waiver information if the applicant applied with Banner Student Self-Service and used a payment card to pay the application fee.

Credit Card Rules window

This window displays payment information. This window is active only if payment card data exists for the applicant. This window is accessed from the Web Credit Card Payment Information option on the Options menu.

This window contains the following fields:

Field	Description	
Miscellaneous Account/ Transaction Account/ None	Radio group that indicates where the payment card transaction is stored:	
	Miscellaneous Account	Miscellaneous transaction stored on TBRMISC
	Transaction Account	Individual transaction stored on TBRACCD
	None	No record stored
Receipt Number	Receipt number for the payment card transaction. The Options List allows you to choose the receipt number from one of the following forms: • Student Account Detail (TSADETL) • Miscellaneous Transaction (TSAMISC) • Miscellaneous Transaction - Finance (TFAMISC)	
Amount	Amount of the application fee paid with a payment card.	
Transaction Number	Transaction number of the payment card transaction.	

Fees Mail Submission, Withdrawal Data window

This window contains the following information for application fee waivers:

Field	Description
Application Fee	Radio group that determines whether the application fee is charged or waived: *Charge Fee* Fee charged *Waive Fee* Fee waived *None* Fee not charged. No billing record generated.
Application Fee Waiver	Application waiver code that indicates why the application fee is waived. A code can be entered in this field if **Application Fee** is *Waive Fee*. List - Application Waiver Code (STVWAIV)

Electronic Application Submitted Form (SAAETBL)

This form is used to display and review information that is received from a Banner Student Self-Service admissions application before it is loaded into permanent Banner tables.

The Credit Card Details window displays payment card information. To access this window, enter a Web ID that has a payment card transaction associated with it. Then select Next Block to access the Application Data block. Select *Financial Details* in the **Select** drop-down list. This window contains the following display-only fields:

Field	Description
Transaction Number	Transaction number for the payment card transaction.
Amount	Amount of the application fee paid with a payment card.
Receipt Number	Receipt number for the payment card transaction.
Miscellaneous	Accounts Receivable form where the transaction detail is displayed: *T* TBRACCD (viewed on TSAAREV and TSADETL forms) *M* TBRMISD (viewed on TSAMISC and TFAMISC forms)
Waiver	Application fee waiver code.

Electronic Applicant Web Default Rules Form (SAAWADF)

This form is used to define the rules for processing admissions applications that are entered with Banner Student Self-Service.

The E-Mail and Credit Card Rules window defines the payment card rules for admissions application fees. This window contains the following fields:

Field	Description
Charge Detail	Detail code associated with the application fee. This code is used when the fee is recorded in Banner. This field is required if payment card processing is allowed or required. Detail codes for application fees are validated against the Detail Charge/Payment Code Definition Table (TBBDETC). If payment card processing is allowed or required, the detail code must have category code *APF*, must have type *C* (charge), and must be active. List - Detail Code Validation (TSADETC)
Amount	Default amount associated with the detail code (for example, $25.00 for the application fee). This field is required if payment card processing is allowed or required.
Application Fee Admission Request	Admissions application checklist code associated with the application fee. List - Admission Request Code Validation (STVADMR)
Processing	Radio group that determines whether card payments are allowed, required, or not allowed for Banner Student Self-Service application fees.
Allow Waiver	Check box that indicates whether application fee waivers are allowed on applications made with Banner Student Self-Service.

Field	Description
Transactions to Miscellaneous Table	Check box that indicates whether payment card transactions are recorded in the Miscellaneous Transaction Charge/Payment Detail Table (TBRMISD) and/or Miscellaneous Transaction Receipt Header Table (TBBMISC). If an applicant does not exist in Banner, the transaction is recorded in TBRMISD and/or TBBMISC, regardless of the setting of this check box. The transaction is viewable on the Miscellaneous Transaction Form - Finance (TFAMISC) if Banner Finance is installed, or on the Miscellaneous Transaction Form (TSAMISC) if Banner Finance is not installed. If an applicant exists in Banner and this check box is selected, the transaction is recorded on TBRMISD and/or TBBMISC. The transaction is viewable on TFAMISC or TSAMISC. If an applicant exists in Banner and this check box is cleared, the transaction is recorded in the Account Charge/Payment Detail Table (TBRACCD). The transaction is viewable on the Account Detail Form (TSADETL) and on the Account Detail Review Form - Student (TSAAREV).

Refer to SAAWADF online help for more details on how admissions application fees are processed.

Form setup for enrollment verification requests

The following Banner Student forms support the use of payment cards to pay enrollment verification request fees with Banner Student Self-Service:

STVEPRT	Enrollment Verification Type Code Validation Form
STVWSSO	Web Self Service Options Validation Form
STVWPYO	Web Payment Options Validation Form
SFAEPRT	Enrollment Verification Request Rules Form

Enrollment Verification Type Code Validation Form (STVEPRT)

This form is used to define enrollment verification type codes. These codes identify types of enrollment verification such as business verification or full disclosure.

STVEPRT contains the following fields:

Field	Description
Code	Enrollment verification type code. Once an enrollment verification type code record is saved, the code cannot be changed. Once a code is used in any other record, the enrollment verification type code record cannot be deleted.
Description	Description of the enrollment verification type code.
Activity Date	Date when the record was created or last updated. Display only.

Web Self Service Options Validation Form (STVWSSO)

This form is used to define self-service option codes. These codes are used with enrollment verifications and transcript requests that are created with Banner Student Self-Service.

STVWSSO contains the following fields:

Field	Description
Code	Self-service option code. Once a code is saved, the code cannot be changed. Once a code is used in any other record, the record cannot be deleted. Required.
Description	Description of the self-service option code. Required.
Charge	Monetary amount associated with the self-service option code. Valid values are *00.00* through *9999999999.99*. Optional.
Issued To	Text to be inserted into the Issued To line on the enrollment verification or paper transcript for the self-service option code. Optional. Default is null. ***Example*** The text *Hold for pickup* can be associated with the *HOLD* option code. When the *HOLD* option code is used for an enrollment verification or paper transcript, the comment *Hold for pickup* is inserted into the Issued To line on the document.

Field	Description
Printer Code	Printer ID associated with enrollment verifications and transcript requests created using the self-service option code via Banner Student Self-Service. Optional. When you run the Academic Transcript Process (SHRTRTC) or the Enrollment Verification Report (SFRENRL), you can specify this printer code in the appropriate parameter to automatically print all requests on the designated printer. List - Printer Validation (GTVPRNT)
User	User who created or last updated the record. Display only.
Activity Date	Date when the record was created or last updated. Display only.

Web Payment Options Validation Form (STVWPYO)

This form is used to define payment option codes. These codes are used with enrollment verifications, transcript requests, and graduation applications that are created with Banner Student Self-Service.

STVWPYO contains the following fields:

Field	Description
Code	Payment option code. Once a code is saved, the code cannot be changed. Once a code is used in any other record, the record cannot be deleted. Required.
Description	Description of the payment option code. Required.
Credit Card Indicator	Check box that indicates whether this payment option code invokes the payment card process.
Detail Code	Detail code used to bill enrollment verifications, transcript requests, or graduation applications created with this Web payment option code. Required. Detail codes are validated against the Detail Code Control Form (TSADETC) for detail codes with a detail category of *TRN*. Accounts Receivable charges are posted only when the **Billing Term** (if being posted to the student's account), **Detail Code**, and **Amount** fields are all populated and valid. List - Detail Code Control Form (TSADETC)

Field	Description
User ID	User who created or last updated the record. Display only.
Activity Date	Date when the record was created or last updated. Display only.

Enrollment Verification Request Rules Form (SFAEPRT)

This form is used to define the rules for printing enrollment verification documents. You can create and maintain an unlimited number of types of enrollment verification documents. For example, you can set up one type (GSL Enrollment Verification Type) to list the schedule of classes and cumulative hours information. You can set up another type (Military Enrollment Verification Type) to print academic standing and tuition information.

Enrollment verification type codes must be created on the Enrollment Verification Type Code Validation Form (STVEPRT) before rules are created on SFAEPRT.

Note

The Select function can only be used to return a value when this form is called from another form.

Print Options window

This window is used to specify the information that is printed on the enrollment verification documents. This window contains the following fields:

Field	Description
Banner ID	Check box that indicates whether Banner IDs, as defined in the **ID** field on the General Person Form (SPAPERS), are printed on enrollment verification documents. The default value is checked when creating an enrollment verification rule.
SSN/SIN/TIN	Check box that indicates whether social security numbers/social insurance numbers/tax identification numbers, as defined in the **SSN/SIN/TIN** field on the General Person Form (SPAPERS), are printed on enrollment verification documents. The default value is checked when creating an enrollment verification rule.

Field	Description
SSN/SIN/TIN Mask	Format mask for displaying the SSN/SIN/TIN on enrollment verification documents. Enter *X* to display a value and * to hide a value.
Birth Date Mask	Format mask for displaying the birth date on enrollment verification documents. List - Sample Date Format Masks

Self-Service Print Options window

This window is used to specify detailed information for enrollment verification requests that are created with Banner Student Self-Service.

Processing Control block

This block contains the following fields:

Field	Description
Self-Service Request	Check box that indicates paper enrollment verification requests can be processed from Banner Student Self-Service.
Self-Service Academic Year	Check box that indicates enrollment verification requests can be selected by academic year when processed from Banner Student Self-Service.
Self-Service Confirmation Letter	Self-service confirmation letter to be used for enrollment verifications. List - Letter Code Validation (GTVLETR)
Self-Service Printers	Destination printer where the confirmation letters are printed. List - Printer Validation (GTVPRNT)

Service Level block

This block contains the following fields:

Field	Description
Code	Self-service option code associated with the enrollment verification type code. Required. List - Web Self Service Options Validation (STVWSSO)
Description	Description of the self-service option code. Display only.
Type	Type of Accounts Receivable account to which charges for the enrollment verification request are posted. Required. *Student* Post to student's account. *Miscellaneous* Post to miscellaneous account (default).
Charge	Fee associated with the self-service option code. This amount comes from the Web Self Service Options Validation Form (STVWSSO). When it is populated, it can be updated. Valid values are *0.00* through *999999999.99*. Optional.
Per	Indicator that determines how the fee is charged. Required if a value is entered in **Charge**. *R* Per request *C* Per copy (default)
User ID	User who created or last updated the record. Display only
Activity Date	Date when the record was created or last updated. Display only.

Payment Options block

This block contains the following fields:

Field	Description
Code	Payment option code associated with the enrollment verification type code. Required. List - Web Payment Options Validation Form (STVWPYO)
Description	Description of the payment option code. Display only.
Detail Code	Detail code associated with the payment option code. This code comes from the Web Payment Options Validation Form (STVWPYO). Display only.

Field	Description
User ID	User who created or last updated the record. Display only.
Activity Date	Date when the record was created or last updated. Display only.

Form setup for registration and student accounts

Banner Student does not require any setup to support payment card transactions that are entered in Banner Student Self-Service for registration fees and student accounts.

Form setup for transcript requests

The following Banner Student forms support the use of payment cards to pay transcript request fees with Banner Student Self-Service:

STVTPRT	Transcript Type Code Validation Form
STVWSSO	Web Self Service Options Validation Form
STVWPYO	Web Payment Options Validation Form
SHATPRT	Transcript Type Rules Form
SHAWTRR	Web Transcript Request Rules Form

Transcript Type Code Validation Form (STVTPRT)

This form is used to define transcript type codes. These codes identify transcript types such as official, internal, and advising.

STVTPRT contains the following fields:

Field	Description
Code	Transcript type code. Once a transcript type code record is saved, the code cannot be changed. Once a code is used in any other record, the transept type code record cannot be deleted.
Description	Description of the transcript type code.

Field	Description
Web Indicator	Check box that indicates whether the transcript type code is displayed in Banner Student Self-Service. You can limit the types of transcripts that can be requested with Banner Student Self-Service to a subset of all transcript types. The default is unchecked when a new record is added, but you can change it at any time.
Web Request Indicator	Check box that indicates whether the transcript type code can be used by students when creating transcript requests with Banner Student Self-Service.
Activity Date	Date when the record was created or last updated. Display only.

Web Self Service Options Validation Form (STVWSSO)

This form is used to define self-service option codes. These codes are used with enrollment verifications and transcript requests that are created with Banner Student Self-Service. Refer to "Web Self Service Options Validation Form (STVWSSO)" on page 4-8 for details about the form.

Web Payment Options Validation Form (STVWPYO)

This form is used to define payment option codes. These codes are used with enrollment verifications, transcript requests, and graduation applications that are created with Banner Student Self-Service. Refer to "Web Payment Options Validation Form (STVWPYO)" on page 4-9 for details about the form.

Transcript Type Rules Form (SHATPRT)

This form is used to define the rules for printing transcripts.

Transcript type codes must be created on the Transcript Type Code Validation Form (STVTPRT) before rules are created on SHATPRT.

Main window - key block

This block contains the following field:

Field	Description
Type	Transcript type code associated with the rules.

Self-Service Print Options window

This window is used to specify detailed information for transcript requests that are created with Banner Student Self-Service.

Processing Control block

This block contains the following fields:

Field	Description
Allow Hold for End of Term Processing	Check box that indicates whether students can request that their transcripts be printed after the end of term grades are processed. Default is unchecked.
Allow Electronic Transcripts on the Web	Check box that indicates whether students can request that their transcripts be sent electronically to other institutions. Default is unchecked.
Allow Hold for Degree Processing	Check box that indicates whether students can request that their transcripts be printed after their degrees are posted. Default is unchecked.
Electronic Letter Code	Electronic letter used on the Confirm Transcript Request page. Optional. List - Letter Code Validation (GTVLETR)

Service Level block

This block contains the following fields:

Field	Description
Code	Self-service option code associated with the transcript type code. Required. List - Web Self Service Options Validation (STVWSSO)
Description	Description of the self-service option code. Display only.
Type	Type of Accounts Receivable account to which charges for the transcript request are posted. Required. *Student* Post to student's account. *Miscellaneous* Post to miscellaneous account (default).

Field	Description
Charge	Fee associated with the self-service option code. This amount comes from the Web Self Service Options Validation Form (STVWSSO). When it is populated, it can be updated. Valid values are *0.00* through *999999999.99*. Optional.
Per	Indicator that determines how the fee is charged. Required if a value is entered in **Charge**. *R* Charge per transcript request. Self-service fees are always charged *C* Charge per transcript copy (default). Self-service fees depend on the number of free copies defined on the Web Transcript Request Rules Form (SHAWTRR).
User ID	User who created or last updated the record. Display only
Activity Date	Date when the record was created or last updated. Display only.

Payment Options block

This block contains the following fields:

Field	Description
Code	Payment option code associated with the transcript type code. Required. List - Web Payment Options Validation Form (STVWPYO)
Description	Description of the payment option code. Display only.
Detail Code	Detail code associated with the payment option code. This code comes from the Web Payment Options Validation Form (STVWPYO). Display only.
User ID	User who created or last updated the record. Display only.
Activity Date	Date when the record was created or last updated. Display only.

Web Transcript Request Rules Form (SHAWTRR)

This form is used to define the institutional rules for processing transcript requests created with Banner Student Self-Service.

SHAWTRR contains the following fields:

Field	Description
Maximum transcript requests per day	Limit on the number of transcript requests a student may create per unique date (DD-MON-YYYY excluding HH:MM:SS). Valid values are *01* through *99*. Default is *99*.
Maximum transcripts allowed per day	Limit on the overall number of transcripts (copies) a student may request per unique date (DD-MON-YYYY excluding HH:MM:SS). Valid values are *001* through *999*. Default is *999*.
Maximum transcripts per request	Limit on the number of transcripts (copies) a student may request per each transcript request. Valid values are *001* through *999*. Default is *999*.
Maximum free transcripts before charges	Limit on the overall number of transcripts (copies) a student may request before incurring charges for transcript processing services. Valid values are *000* through *999*. Default is *999*. **Note:** If a student has not received all of his or her free transcripts, Banner does not prompt for a payment card payment.
Default course level to ALL on transcript	Check box that determines the course level printed on transcripts: *selected* — Course level defaults to *ALL*. Course level and student term information fields are not displayed on the Web page. Values for those fields default as they do on SHARQTC when the course level is set to *AL*. *cleared* — Student may request a transcript for a specific level of courses (default).
User ID	User who created or last updated the record.
Activity Date	Date when the record was created or last updated.

Form setup for graduation applications

The following Banner Student forms support the use of payment cards to pay graduation application fees with Banner Student Self-Service:

STVWPYO	Web Payment Options Validation Form
STVGADR	Graduation Application Display Rule Code Validation Form
SHAGADR	Self-Service Graduation Application Display Rules Form

Web Payment Options Validation Form (STVWPYO)

This form is used to define payment option codes. These codes are used with enrollment verifications, transcript requests, and graduation applications that are created with Banner Student Self-Service. Refer to "Web Payment Options Validation Form (STVWPYO)" on page 4-9 for details about the form.

Graduation Application Display Rule Code Validation Form (STVGADR)

This form is used to define graduation application display rule codes. These codes can be used to identify different rules for processing graduation applications.

STVGADR contains the following fields:

Field	Description
Graduation Application Display Rule Code	Graduation application display rule code.
Description	Description of the graduation application display rule code.
User ID	User who created or last updated the record. Display only.
Activity Date	Date when the record was created or last updated. Display only.

Self-Service Graduation Application Display Rules Form (SHAGADR)

This form is used to define the rules for processing graduation applications created with Banner Student Self-Service.

Main window - key block

This block contains the following field:

Field	Description
Graduation Application Display Rule	Graduation application display rule code. List - Graduation Application Display Rule Code (STVGADR)

Payment Options window

This window is used to specify payment options for graduation applications that are created with Banner Student Self-Service.

Processing Control block

This block contains the following field:

Field	Description
Charge Graduation Fee	Check box that indicates whether graduation fees are charged for the rule. If the check box is selected, a charge is applied to the student's account (TBRACCD) when a graduation application is submitted and the associated graduation application display rule is selected. If this check box is selected, at least one detail code and amount must be defined in the Payment Options block.

Payment Options block

This block is used to set up detail codes and charges. This block contains the following fields:

Field	Description
Payment Option	Payment option code associated with the graduation application display rule code. Required. List - Web Payment Options Validation Form (STVWPYO)
Description	Description of the payment option code.
Detail Code	Detail code associated with the payment option code. This code comes from TSADETC for the payment option.
Charge	Fee associated with the payment option code.
User ID	User who created or last updated the record.
Activity Date	Date when the record was created or last updated.

5 Banner Student Self-Service Implementation

Students can use payment cards to pay fees associated with the following types of Banner® Student Self-Service transactions:

- Admissions applications
- Enrollment verification requests
- Registration and student accounts
- Transcript requests
- Graduation applications

This chapter describes the processing features that apply to Banner Student Self-Service payment card transactions. This chapter also gives instructions for setting up Banner Student Self-Service to support these payment card transactions.

Processing features

Chapter 1, "Overview" describes the processing flow that is common to all Banner payment card processing. In addition, the following processing features apply to payment card processing for Banner Student Self-Service transactions.

- Address information can be automatically collected from a student's address records and stored for payments that are processed for admissions application fees, transcript request fees, and enrollment verification fees. For detailed information, see "Build address hierarchy" on page 3-3.
- Merchant IDs are used on outgoing payment transactions to tell the payment processing vendor which payment profile should be used. You can optionally set up multiple merchant IDs to simplify the settlement process for payment card transactions. This is valuable if you must separately settle payments based on level, campus, or college. For detailed information, see "Build multiple merchant ID hierarchy" on page 3-6.
- You can use an option on the Crosswalk Validation Form (GTVSDAX) to specify whether students can use payment cards if their Accounts Receivable accounts have holds.

- When Quick Start processing is used, the payment card transaction must be processed successfully before the student is admitted, before the student record is created, and before the student can register for classes.
- If a full waiver is allowed for a student's payment transaction, the student is not redirected to the payment processing vendor's Web site. If no waiver is in place or if a partial waiver is in place, the student is redirected to the payment processing vendor's Web site where the student enters payment card information.
- Your institution can specify whether a signature page or acknowledgement letter is displayed when a payment card transaction is processed successfully. The student can print this page or letter to verify the payment card transaction.
- A successfully processed payment card transaction is stored on one of the following Banner Accounts Receivable tables:

TBRACCD	Account Charge/Payment Detail Table
TBRMISD	Miscellaneous Transaction Charge/Payment Detail Table

  A failed transaction is not stored in Banner Accounts Receivable.
- A successfully processed payment card transaction can be viewed on one of the following Banner Accounts Receivable forms:

TSAAREV	Account Detail Review Form - Student
TSADETL	Student Account Detail Form
TFAMISC	Miscellaneous Transaction Form - Finance
- Miscellaneous transactions are used for accounts that do not have permanent records in Banner Accounts Receivable. Miscellaneous transactions can be used for the fees associated with admissions applications, enrollment verification requests, and transcript requests. An address is required if a student uses a payment card to pay a fee that creates a miscellaneous account transaction.

  For admissions applications that have no Electronic Admissions Address Table (SARADDR) record or Address Table (SPRADDR) record, the student is taken to a Web page to enter the address information.

  For enrollment verification requests and transcript requests that have no SPRADDR record, the student is taken to a Web page to enter the address information.

Note

Miscellaneous transactions are not used for registration fees or graduation application fees. A student does not need an address record to process payment for those fees.

Common implementation for Banner Student Self-Service

The following steps are used to implement payment card processing for all Banner Student Self-Service applications:

- Step 1, “Verify common implementation”
- Step 2, “Define AR holds indicator”
- Step 3, “Define default term codes”

Additional implementation steps specific to admissions applications, enrollment verification requests, registration fees, student accounts, transcript requests, and graduation applications are described later in this chapter.

Step 1 Verify common implementation

Verify that the implementation steps common to all Banner payment card processing have been performed. See Chapter 3, “Common Implementation” for details.

Step 2 Define AR holds indicator

Create a GTVSDAX rule that defines the accounts receivable holds indicator.

Note

Once a student is referred for collection, the student cannot make payment card payments regardless of this rule.

Internal Code	*WEBCCHOLDS*	
Sequence	none	
Group	*PAYMENTVENDOR*	
External Code	*Y*	Students can make payment card payments even if there are holds on their AR records.
	N	Students cannot make payment card payments if there are holds on their AR records.
Description	*Allow CC Payments if A/R Holds*	
System Required	selected	

Step 3 Define default term codes

Some payment card transactions require a term code. For example, the Account Summary page requires a student to select a term. If a student is on the Registration Fee Assessment page, a term is already selected and the payment is made for that term.

Payment card processing determines the term code as follows:

- Payment card processing first checks for the term code that the student selected in Banner Student Self-Service.
- If the student did not select a term code, payment card processing checks the TBBDETC table for the default term code assigned to the associated detail code on the Credit Card Merchant ID Form (GOAMERC).
- If the detail code does not have a default term code on TBBDETC, the default term code defined on GTVSDAX is used.
- If a student still does not have a specified term, the student receives an insert error message with instructions to contact the Bursar's Office. In this case, the payment is approved but is not inserted into Banner.

Use the following steps to define the default term codes on TBBDETC and GTVSDAX.

1. Access the Detail Code Control Form - Student (TSADETC).
2. For each detail code associated with payment card transactions:
   2.1. Query for the detail code.
   2.2. Enter the default term code in the **Term** field.
   2.3. Save.
3. Access the Crosswalk Validation Form (GTVSDAX).
4. Create the following GTVSDAX rule. This rule identifies the default term code that is used if a detail code on TBBDETC does not have a default term code.

Internal Code	*DEFAULT*
Sequence	none
Group	*WEBCCDEFTERM*
External Code	Default term from STVTERM to use for a payment card transaction if no term is associated with the detail code on TBBDETC.

Description	*Default Term for Web CC Insert*
System Required	cleared

5. Save.

Implementation for admissions applications

The following steps are used to implement payment card processing for admissions applications:

- Step 1, “Verify form setup”
- Step 2, “Customize procedure definitions in Banner Web Tailor”
- Step 3, “Customize information text in Banner Web Tailor”

Step 1 Verify form setup

Verify that the supporting Banner Student forms for admissions applications are set up. See “Form setup for admissions applications” on page 4-1 for details.

Step 2 Customize procedure definitions in Banner Web Tailor

Payment card processing for admissions applications uses the following procedures:

```
bwskapmt.P_SelectWaiver
bwskapmt.P_ProcessWaiver
bwskalog.P_ProcIndex
```

The procedure definitions are populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Web Menus and Procedures.
4. Verify that procedure definitions for the following pages exist and are configured correctly. Baseline configurations are described in the following text.

Select a Waiver (bwskapmt.P_SelectWaiver)

Label	Value
Page Name	`bwskapmt.P_SelectWaiver`
Description	Select a Waiver
Module	Student Services
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Select a Waiver
Header Text	Select a Waiver
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	none
Back Link Text	none
Back Link Image	none
Back Link Menu Indicator	No
Admin Secured	No
Associated Roles	Student

Process a Waiver (bwskapmt.P_ProcessWaiver

Label	Value
Page Name	`bwskapmt.P_ProcessWaiver`
Description	Process Waiver
Module	Student Services
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Process Waiver
Header Text	Process Waiver
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	none

Label	Value
Back Link Text	none
Back Link Image	none
Back Link Menu Indicator	No
Admin Secured	No
Associated Roles	Student

Application Fee Payment (bwskalog.P_ProcIndex)

Label	Value
Page Name	`bwskalog.P_ProcIndex`
Description	Submit Admissions Application
Module	Student Self-Service
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Application Fee Payment
Header Text	Application Fee Payment
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	none
Back Link Text	none
Back Link Image	none
Back Link Menu Indicator	No
Admin Secured	No
Associated Roles	Student

Step 3 Customize information text in Banner Web Tailor

Information text is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Information Text.

4. Verify the information text labels for `bwskapmt.P_SelectWaiver.` This is the baseline configuration.

Seq #	Label	Information Text
1	DEFAULT	Please select an applicable waiver from the list.

Implementation for enrollment verification requests

The following steps are used to implement payment card processing for enrollment verification requests:

- Step 1, “Verify form setup”
- Step 2, “Customize procedure definition in Banner Web Tailor”

Step 1 Verify form setup

Verify that the supporting Banner Student forms for enrollment verification requests are set up. See “Form setup for enrollment verification requests” on page 4-7 for details.

Step 2 Customize procedure definition in Banner Web Tailor

Payment card processing for enrollment verification requests uses the `bwskrqst.P_Disp_Confirm` procedure. The procedure definition is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Web Menus and Procedures.
4. Verify that the procedure definition for `bwskrqst.P_Disp_Confirm` exists and is configured correctly. This is the baseline configuration:

Label	Value
Page Name	`bwskrqst.P_Disp_Confirm`
Description	Confirm Enrollment Verification
Module	Student Self-Service
Comments	none
Enabled Indicator	Yes

Label	Value
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Enrollment Verification Request Summary
Header Text	Enrollment Verification Request Summary
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	`bmenu.P_AdminMnu`
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes
Admin Secured	No
Associated Roles	Student

Implementation for registration and student accounts

The following steps are used to implement payment card processing for registration and student accounts:

- Step 1, “Customize procedure definition in Banner Web Tailor”
- Step 2, “Establish link to payment card payments”

Step 1 Customize procedure definition in Banner Web Tailor

Payment card processing for registration and student accounts uses the `bwckcpmt.P_CCPaymentTermSelected` procedure. The procedure definition is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Web Menus and Procedures.
4. Verify that the procedure definition for `bwckcpmt.P_CCPaymentTermSelected` exists and is configured correctly. This is the baseline configuration:

Label	Value
Page Name	bwckcpmt.P_CCPaymentTermSelected
Description	Credit Card Payment if Term is Selected
Module	Common
Comments	Used in Fee Assessment Link
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Credit Card Payment
Header Text	Credit Card Payment
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	bmenu.P_StuMainMnu
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes
Admin Secured	No
Associated Roles	Student

Step 2 Establish link to payment card payments

Payments for registration fees and student accounts can be made through links on the Account Summary, Account Summary by Term, and Registration Fee Assessment pages. The term must be selected before payment processing can start. Because the Registration Fee Assessment page already forces a term selection, selecting a term again is redundant. You can control the destination of the **Credit Card Payment** link as follows:

- To force the student to select the term again, the link should point to the Registration Term page (bwskflib.P_SelDefTerm).
- To eliminate the redundancy and use the term that was already selected, the link should point to the Tuition and Fees Payment page (bwckcpmt.P_CCPaymentTermSelected).

Use the following steps to define the destination of the **Credit Card Payment** link:

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Menu Items.

4. Enter *bwskffee.P_FeeAsses* in the **Search by Name** field.

5. Click **Search**.

6. Click the procedure name *bwskffee.P_FeeAsses* in the returned list of procedures.

7. On the Reorder or Customize Menu Items page, change the URL of the first menu item (Sequence Number 1, Credit Card Payment) to one of the following procedures:

`bwskflib.P_SelDefTerm`	Maps the link to the Registration Term page
`bwckcpmt.P_CCPaymentTermSelected`	Maps the link to the Tuition and Fee Payment page

Implementation for transcript requests

The following steps are used to implement payment card processing for transcript requests:

- Step 1, “Verify form setup”
- Step 2, “Customize procedure definitions in Banner Web Tailor”

Step 1 Verify form setup

Verify that the supporting Banner Student forms for transcript requests are set up. See “Form setup for transcript requests” on page 4-13 for details.

Step 2 Customize procedure definitions in Banner Web Tailor

Payment card processing for transcript requests uses the `bwskwtrr.P_Disp_Confirm` procedure. The procedure definition is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.

2. Navigate to Web Tailor Administration.

3. From the Web Tailor menu, select Web Menus and Procedures.

4. Verify that the procedure definition for `bwskwtrr.P_Disp_Confirm` exists and is configured correctly. This is the baseline configuration:

Label	Value
Page Name	bwskwtrr.P_Disp_Confirm
Description	Confirm Transcript Request
Module	Student Self-Service
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Confirm Transcript Request
Header Text	Transcript Request Summary
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	bmenu.P_AdminMnu
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes
Admin Secured	No
Associated Roles	Student

Implementation for graduation applications

The following steps are used to implement payment card processing for graduation applications:

- Step 1, “Verify form setup”
- Step 2, “Customize procedure definition in Banner Web Tailor”
- Step 3, “Customize information text in Banner Web Tailor”

Step 1 Verify form setup

Verify that the supporting Banner Student forms for graduation applications are set up. See “Form setup for graduation applications” on page 4-17 for details.

Step 2 Customize procedure definition in Banner Web Tailor

Payment card processing for graduation applications uses the `bwskgrad.P_Disp_Confirm` procedure. The procedure definition is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Web Menus and Procedures.
4. Verify that the procedure definition for `bwskgrad.P_Disp_Confirm` exists and is configured correctly. This is the baseline configuration:

Label	Value
Page Name	`bwskgrad.P_Disp_Confirm`
Description	Graduation Application Summary
Module	Student Self-Service
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Graduation Application Summary
Header Text	Graduation Application Summary
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	`bmenu.P_AdminMnu`
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes
Admin Secured	No
Associated Roles	Student

Step 3 Customize information text in Banner Web Tailor

Information text is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Information Text.
4. Verify the information text labels for `bwskgrad.P_Disp_Payment.` This is the baseline configuration.

Seq #	Label	Information Text
1	DEFAULT	This is the information that will be submitted for your application to graduate.

Web pages

Banner Student Self-Service uses the following Web pages for payment card processing. Refer to the *Banner Student Self-Service User Guide* for more details about these pages.

Select a Waiver (bwskapmt.P_SelectWaiver)

This page allows a student to select an application fee waiver. If a waiver is selected, the **Continue** button displays the Process a Waiver page.

Process a Waiver (bwskapmt.P_ProcessWaiver)

This page displays the final payment amount, after any applicable waiver is subtracted from the application fee. The **Submit Payment** button transmits the transaction to the selected external vendor for payment card processing.

Application Fee Payment (bwskalog.P_ProcIndex)

This page confirms the amount of the admissions application fee. The **Submit Payment** button transmits the transaction to the selected external vendor for payment card processing.

Enrollment Verification Request Summary (bwskrqst.P_Disp_Confirm)

This page confirms the details of the enrollment verification request before it is submitted. The **Submit Request** button transmits a payment transaction to the selected external vendor for payment card processing.

Credit Card Payment (bwckcpmt.P_CCPaymentTermSelected)

This page allows a student to enter a payment amount for registration fees or student accounts. The **Submit** button transmits the transaction to the selected external vendor for payment card processing.

Transcript Request Summary (bwskwtrr.P_Disp_Confirm)

This page confirms the details of the transcript request before it is submitted. The **Submit Request** button transmits a payment transaction to the selected external vendor for payment card processing.

Graduation Application Summary (bwskgrad.P_Disp_Payment)

This page confirms the details of the graduation application before it is submitted. The **Submit Request** button transmits a payment transaction to the selected external vendor for payment card processing.

6 Banner Flexible Registration Implementation

Students can use Banner® Flexible Registration to register for non-credit classes typically associated with continuing education and community service programs as well as traditional credit classes. Students can use payment cards to pay the associated registration fees.

This chapter describes the processing features that apply to Banner Flexible Registration payment card transactions. This chapter also gives instructions for setting up Banner Flexible Registration to support these payment card transactions.

Note

Banner Flexible Registration is based on Banner Student. Therefore, some of the setups may already be implemented.

Processing features

Chapter 1, "Overview" describes the processing flow that is common to all Banner payment card processing. In addition, the following processing features apply to payment card processing for Banner Flexible Registration transactions.

- Merchant IDs are used on outgoing payment transactions to tell the payment processing vendor which payment profile should be used. You can optionally set up multiple merchant IDs to simplify the settlement process for payment card transactions. This is valuable if you must separately settle payments based on level, campus, or college. For detailed information, see "Build multiple merchant ID hierarchy" on page 3-6.
- You can use an option on the Crosswalk Validation Form (GTVSDAX) to specify whether students can use payment cards if their Accounts Receivable accounts have holds.
- A successfully processed payment card transaction is stored on the Account Charge/Payment Detail Table (TBRACCD). A failed transaction is not stored in Banner Accounts Receivable.
- A successfully processed payment card transaction can be viewed on the Account Detail Review Form - Student (TSAAREV) and on the Student Account Detail Form (TSADETL).

Implementation

The following steps are used to implement payment card processing for Banner Flexible Registration:

- Step 1, “Verify common implementation”
- Step 2, “Verify AR holds indicator”
- Step 3, “Verify default term code”
- Step 4, “Verify process code”
- Step 5, “Enable payment card processing”
- Step 6, “Configure success and failure URLs”

Step 1 Verify common implementation

Verify that the implementation steps common to all Banner payment card processing have been performed. See Chapter 3, “Common Implementation” for details.

Step 2 Verify AR holds indicator

Verify that the accounts receivable holds indicator is defined on the Crosswalk Validation Form (GTVSDAX). See “Define AR holds indicator” on page 5-3 for details.

Step 3 Verify default term code

Verify that the default term code is defined. See “Define default term codes” on page 5-4 for details.

Step 4 Verify process code

Verify that the FLEXREGCCREGFEES process code is defined on the Process Name Validation Form (GTVPROC). See “Define processes that use payment card processing” on page 3-2 for details.

Step 5 Enable payment card processing

Payment card processing for Banner Flexible Registration is enabled at the catalog level. Use the following steps to enable payment card processing for a catalog.

1. Access the Flexible Registration Catalog Rules Form (SFRACTLG).
2. Enter the name of the catalog in the key block.
3. Go to the Rules window.

4. Enter the following information:

Group	*PAYOPT*	
Parameter	*Pay Using Payment Processor Connection*	
Value	*Y*	Allow students to pay with payment cards.
	N	Do not allow students to pay with payment cards.

5. Save.

Step 6 Configure success and failure URLs

After the payment processing vendor processes a payment, the vendor redirects the browser to a success or failure URL in Banner, depending on whether the transaction update is a success or failure. Banner then redirects the browser to a product-specific success or failure URL. The success and failure URLs for Banner Flexible Registration are specified in the following Spring configuration file on the application server:

```
%OC4J_DEPLOY_DIRECTORY%/WEB-INF/pcipaymenturl-
applicationcontext.xml
```

The URL values are specified as follows:

```
<property name="successURL" value="/pciPaymentSuccess.jsp?" />
<property name="failureURL" value="/pciPaymentFailure.jsp?" />
```

The URL values specified in this file do *not* need to be changed. If you choose to modify the URL values, follow these guidelines:

- URL values that begin with a period (.) or slash (/) are considered to be relative URLs. Banner Flexible Registration will build the complete URLs at runtime.

  URL values that do not begin with a period (.) or slash (/) are considered to be absolute URLs. Banner Flexible Registration will use the exact URL values.

- URL values must end with a question mark (?). When the final redirect occurs, this ensures that the appropriate URL request parameters can be appended at runtime to the URLs as needed.

- If URL values and the default context root (`flexibleregistration`) are changed to new values, the new context root value must be included in the changed absolute URL values.

Use the following steps to modify the success and failure URL values.

1. Deploy Flexible Registration to an OC4J application server.
2. Shut down the OC4J instance where Banner Flexible Registration is deployed.
3. Locate and open the `pcipaymenturl-applicationcontext.xml` file.
4. Change the `successURL` and `failureURL` values.
5. Save the `pcipaymenturl-applicationcontext.xml` file.
6. Restart the OC4J instance where Banner Flexible Registration is deployed.

7 Banner Advancement Implementation

Individuals and organizations can use payment cards to make gifts and pledge payments via Banner® Advancement Self-Service. The following Banner Advancement forms support the use of payment cards to make gifts and pledge payments with Banner Advancement Self-Service:

AFACAMP	Campaign Detail Form
ADADESG	Designations Form
AMAINFO	Advancement Prospect Information Form
ATVFISC	Fiscal Year Validation Form

This chapter gives instructions for setting up Banner Advancement to support payment card transactions.

Campaign Detail Form (AFACAMP)

This form is used to maintain information for a fund-raising campaign. The following fields on the Base window are used for payment card processing:

Field	Description
Allow Web Gifts	Check box that determines whether gifts to this campaign can be made via the Web: *selected* — Gifts can be made via the Web. *cleared* — Gifts cannot be made via the Web.
Start Date End Date	Start and end dates for the campaign. Gifts to the campaign can be made via the Web during this date range only.

Designations Form (ADADESG)

This form is used to maintain information for designations. The following fields on the Header Information window are used for payment card processing:

Field	Description	
Allow Web Gifts	Check box that determines whether gifts to this designation can be made via the Web:	
	selected	Gifts can be made via the Web.
	cleared	Gifts cannot be made via the Web.
Start Date End Date	Start and end dates for the designation. Gifts to the designation can be made via the Web during this date range only.	

Advancement Prospect Information Form (AMAINFO)

This form is used to maintain information about prospects. The following field on the General Information window is used for payment card processing:

Field	Description	
Allow Web Gifts	Check box that determines whether the prospect can make gifts via the Web:	
	selected	The prospect can make gifts via the Web.
	cleared	The prospect cannot make gifts via the Web.

Fiscal Year Validation Form (ATVFISC)

This form is used to define the fiscal year codes that are used in gift processing. The fiscal year must be defined on this form for the date when a payment card transaction is processed.

8 Banner Advancement Self-Service Implementation

Individuals and organizations can use payment cards to make gifts and pledge payments with Banner® Advancement Self-Service. This chapter describes the processing features that apply to Banner Advancement Self-Service payment card transactions. This chapter also gives instructions for setting up Banner Advancement Self-Service to support these payment card transactions.

Processing features

Chapter 1, "Overview" describes the processing flow that is common to all Banner payment card processing. In addition, the following processing features apply to payment card processing for gifts and pledge payments made with Banner Advancement Self-Service.

- The gift number can be displayed on the online receipt.
- The donor ID can be displayed on the online receipt.

Implementation for Banner Advancement Self-Service

The following steps are used to implement payment card processing for Banner Advancement Self-Service:

- Step 1, "Verify common implementation"
- Step 2, "Verify form setup"
- Step 3, "Indicate display of gift number on receipts"
- Step 4, "Indicate display of donor ID on receipts"
- Step 5, "Customize procedure definitions in Banner Web Tailor"
- Step 6, "Customize information text in Banner Web Tailor"
- Step 7, "Customize rules for gifts made with Banner Advancement Self-Service"

Step 1 Verify common implementation

Verify that the implementation steps common to all Banner payment card processing have been performed. See Chapter 3, "Common Implementation" for details. This includes the definition of the default merchant ID for Banner Advancement Self-Service (see "Define default merchant IDs" on page 3-4).

Step 2 Verify form setup

Verify that the supporting Banner Advancement forms are set up. See Chapter 7, "Banner Advancement Implementation" for details.

Step 3 Indicate display of gift number on receipts

Create a GTVSDAX rule that indicates whether the gift number is displayed on online receipts.

Internal Code	*DSPALUGIFT*
Sequence	none
Group	*PAYMENTVENDOR*
External Code	*Y* Display gift number on online receipts. *N* Do not display gift number on online receipts.
Description	*Display Adv Gift Number Web CC*
System Required	cleared

Step 4 Indicate display of donor ID on receipts

Create a GTVSDAX rule that indicates whether the donor ID is displayed on online receipts.

Internal Code	*DSPALUID*
Sequence	none
Group	*PAYMENTVENDOR*
External Code	*Y* Display donor ID on online receipts. *N* Do not display donor ID on online receipts.
Description	*Display Adv Donor ID Web CC*
System Required	cleared

Step 5 Customize procedure definitions in Banner Web Tailor

Payment card processing for Banner Advancement Self-Service uses the following procedures:

```
bwakgift.P_Make_A_Donation
bwakgift.P_Pledge_Payment
bwakgift.P_Donation_Receipt
bwakngft.P_Make_A_Donation
bwakngft.P_Donation_Receipt
```

The procedure definitions are populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Web Menus and Procedures.
4. Verify that procedure definitions for the following pages exist and are configured correctly. Baseline configurations are described in the following text.

Make a Donation (bwakgift.P_Make_A_Donation)

Label	Value
Page Name	`bwakgift.P_Make_A_Donation`
Description	Make a Donation
Module	Alumni & Friends
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Make a Donation
Header Text	Make a Donation
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	`bmenu.P_GivingMnu`
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes

Label	Value
Admin Secured	No
Associated Roles	Alumni, Friend user

Make a Donation (bwakgift.P_Pledge_Payment)

Label	Value
Page Name	`bwakgift.P_Pledge_Payment`
Description	Make a Donation
Module	Alumni & Friends
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Make a Donation
Header Text	Make a Donation
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	`bmenu.P_GivingMnu`
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes
Admin Secured	No
Associated Roles	Alumni, Friend user

Online Receipt (bwakgift.P_Donation_Receipt)

Label	Value
Page Name	`bwakgift.P_Donation_Receipt`
Description	Online Receipt
Module	Alumni & Friends
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	No
Web Page Caching Override	Use System Setting
Page Title	Online Receipt
Header Text	Online Receipt
Header Graphic	none
Page CSS URL	none

Label	Value
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	`bmenu.P_GivingMnu`
Back Link Text	Return to Menu
Back Link Image	none
Back Link Menu Indicator	Yes
Admin Secured	No
Associated Roles	Alumni, Friend user

Make a Donation (bwakngft.P_Make_A_Donation)

Label	Value
Page Name	`bwakngft.P_Make_A_Donation`
Description	No PIDM Credit Card gift page
Module	Alumni & Friends
Comments	none
Enabled Indicator	Yes
Non Secured Access Allowed	Yes
Web Page Caching Override	Use System Setting
Page Title	Make a Donation
Header Text	Make a Donation
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	none
Back Link Text	none
Back Link Image	none
Back Link Menu Indicator	No
Admin Secured	No
Associated Roles	none

Online Receipt (bwakngft.P_Donation_Receipt)

Label	Value
Page Name	`bwakngft.P_Donation_Receipt`
Description	Web No PIDM Gift Receipt
Module	Alumni & Friends
Comments	none

Label	Value
Enabled Indicator	Yes
Non Secured Access Allowed	Yes
Web Page Caching Override	Use System Setting
Page Title	Online Receipt
Header Text	Online Receipt
Header Graphic	none
Page CSS URL	none
Map Title	none
Help Link URL	none
Help CSS URL	none
Back Link URL	`bmenu.P_AluNonSecureMnu`
Back Link Text	Return to Alumni and Friends
Back Link Image	none
Back Link Menu Indicator	No
Admin Secured	No
Associated Roles	none

Step 6 Customize information text in Banner Web Tailor

Information text is populated by installs or upgrades. Use the following steps to verify data and customize settings, if necessary.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Information Text.
4. Verify the information text labels for the following pages. Baseline labels are described in the following text.

Make a Donation (bwakgift.P_Make_A_Donation)

Seq #	Label	Information Text
1	NOT_ALLOWED	Please contact the Advancement Office to make a donation.
1	NO_DEFAULT	Please contact your institution's Advancement Office regarding the use of this feature.

Make a Donation (bwakgift.P_Pledge_Payment)

Seq #	Label	Information Text
1	DEFAULT	You have outstanding pledges. Do you wish to make a payment on one of those pledges?
1	INSTALLMENT	You have elected to pay this pledge in installments. If you choose, you may pay the next installment you have due.
1	INVALID AMT	You may not pay an amount greater than your remaining balance.
1	SELECT AMOUNT	Please select the amount you wish to donate. Should you select an amount other than the balance and wish to donate more than the remaining balance of your pledge, please pay the remainder now and make an additional gift.
1	SELECT PLEDGE	Select a pledge for payment. To exit this process select a menu link at the bottom of this page.
2	SELECT PLEDGE	Records with N/A in the selection column are not available for online payments.

Online Receipt (bwakgift.P_Donation_Receipt)

Seq #	Label	Information Text
1	DEFAULT	Thank you for your contribution. Please print a copy of this page for your records. We appreciate your support.

Step 7 Customize rules for gifts made with Banner Advancement Self-Service

Use the following steps to customize the Banner Web Tailor rules that support gifts that are made with Banner Advancement Self-Service.

1. Enter the Secure Area of Banner Self-Service.
2. Navigate to Web Tailor Administration.
3. From the Web Tailor menu, select Advancement Self-Service Rules.

4. Establish the following rules:

Rule	Description
Default Gift Vehicle Code	Default gift vehicle code from the Pledge/Gift Vehicle Code Validation Form (ATVPGVE). When a user submits a Web gift, Banner automatically uses the vehicle code specified in this rule.
Default Gift Class Code	Default gift class code from the Gift Classification Code Validation Form (ATVGCLS). When a user submits a Web gift, Banner automatically uses the class code specified in this rule as class 1.
Default Gift Solicitation Code	Default gift solicitation code from the Solicitation Type Code Validation Form (ATVSOLC). When a user submits a Web gift, Banner automatically uses the solicitation code specified in this rule.
Default New User Gift Notification User ID	ID of the person who reviews Web gifts made by donors who do not have PIDMs. This person assigns PIDMS to the donors, reviews the gifts, and approves the gifts to be loaded into the Banner database.
Allow Pledge Payment Option	If checked, a donor can pay toward existing pledges. If cleared, a donor can only make one-time gifts.
Allow Gift Split Option	If checked, a donor can split gifts with a spouse, or another existing cross-reference if no spouse record exists. If cleared, a donor cannot split a gift with anyone.
Display Employer Match Information	If checked, matching gift detail for the current employer is displayed on the Credit Card Payment page (`bwakgift.P_Pledge_Payment`). If cleared, the matching gift detail is not displayed.
Allow Web to Satisfy Pledges	If checked, a donor can pay a pledge in full via payment card over the Web. The pledge status in Banner is updated from active to paid. If cleared, you must manually change the pledge status in Banner when the pledge is fully paid.

Rule	Description
Allow Online Receipt	If checked, the donor can view or print an online receipt on the Online Receipt page (`bwakgift.P_Donation_Receipt`). If cleared, the donor must arrange for a receipt through the institution.
Allow Online Matching Gift Processing	If checked, matching gift processing occurs when the gift is created, automatically checking to see if the gift is eligible to be matched. If cleared, you must run the Matching Gift Allocations Report (AGPMATG) in Banner Advancement to see if a gift is eligible to be matched.

Web pages

Banner Advancement Self-Service uses the following Web pages for payment card processing. Refer to the *Banner Advancement Self-Service User Guide* for more details about these pages.

Make a Donation (bwakgift.P_Make_A_Donation and bwakgift.P_Pledge_Payment)

The first Make a Donation page allows a donor to choose whether to make a new gift or make a pledge payment:

- If the donor chooses to make a new gift, the Credit Card Payment Page is displayed for entering gift details.
- If the donor chooses to make a pledge payment, the next Make a Donation page lists the outstanding pledges. The donor selects the pledge to pay. The next Make a Donation page allows the donor to pay the pledge balance or enter another amount.

Credit Card Payment (bwakgift.P_Make_A_Donation and bwakgift.P_Pledge_Payment)

When a donor has a PIDM, this page collects and displays the following information for a Web gift or pledge payment:

- Campaign and designation
- Gift amount

- Choice to split the gift with a spouse
- Employer matching gift information. If the information is not already stored in Banner, the donor can send an e-mail with matching gift information (not saved in Banner).
- Free-form comment

The **Submit Payment** button transmits the transaction to the selected external vendor for payment card processing.

Online Receipt (bwakgift.P_Donation_Receipt)

When a donor has a PIDM, this page provides summary information that the donor can print as a record of the payment. Your institution can customize the text by using Banner Web Tailor.

Make a Donation (bwakngft.P_Make_A_Donation)

When a donor does not have a PIDM, this page collects and displays the following information for a Web gift or pledge payment:

- Campaign and designation
- Gift amount
- Employer matching gift information. The donor can send an e-mail with matching gift information (not saved in Banner).
- Free-form comment

The **Submit Payment** button transmits the transaction to the selected external vendor for payment card processing.

Online Receipt (bwakngft.P_Donation_Receipt)

When a donor does not have a PIDM, this page provides summary information that the donor can print as a record of the payment. Your institution can customize the text by using Banner Web Tailor.

Troubleshooting

This chapter provides solutions for problems that may arise with payment card processing.

Payment not applied to account

Question: Why did I receive the following error message in the browser after submitting a payment card payment?

> **SERIOUS ERROR* A charge has been applied to your credit card, but no payment has been applied to your account. Please PRINT this message and contact the Bursar's office immediately.*

Answer: Accounting information used in the settlement process is not set up correctly on the Credit Card Merchant ID Form (GOAMERC). Review "Define accounting information" on page 3-10 and verify your setup.

Missing payment processing Web page

Question: Why did I receive the following error message when navigating the payment processing Web pages?

> *The procedure name is not found in the database: <package.procedure>*

Answer: Banner® Web Tailor data is missing. Use the following steps to correct the error:

1. Enter the Secure Area of Banner Self-Service.
2. Log in as the a Banner Web Tailor administrator.
3. Navigate to Web Tailor Administration.
4. From the Web Tailor menu, select Web Menus and Procedures.
5. Verify that the package.procedure referenced in the error message is not displayed in the list of procedures.
6. Create a new entry for the missing package.procedure. Review the product-specific chapter of this document for the values that should be entered for each item.