ellucian™

Banner Web Services
Handbook

Release 8.1.3
January 2012

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

Revision History

Publication Date	Summary
January 2012	New version that supports Banner Web Services 8.1.3 software.

Banner Web Services 8.1.3 Handbook

Contents

1 Introduction

This handbook is intended for integration analysts, business analysts, and others interested in the functionality provided by Banner® Web services. This handbook contains the following information for each Banner Web service:

- General description
- Message exchange pattern and specific messages used
- Mapping between message elements/attributes and Banner columns
- Intended usage
- Setup requirements
- Dynamic and static translations that occur between external systems and Banner APIs

Requirements

The Banner Web services require the following components.

Oracle application server and Java

The Banner Web Services Adapters expose Banner functions as Web services. These adapters are 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

You *must* install the Banner Translation Service before you deploy the Banner Web Services Adapters. Refer to the *Banner Translation Service Installation and Administration Guide* for details.

Banner products

The following Banner products support the Banner Web services components and must be installed:

Banner Web Services Adapter for Campus Card Systems

Product	Minimum Version - 8.x
Banner General	8.0 plus patch p1-46c8mj_gen80100

Banner Web Services Adapter for Housing Systems

Product	Minimum Version - 8.x
Banner General	8.0 plus patch p1-46c8mj_gen80100
Banner Student	8.0
Banner Accounts Receivable	8.0

Banner Cardholder Event Publisher

Product	Minimum Version - 8.x
Banner General	8.0 plus patch p1-46c8mj_gen80100

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.

Web services use XML extensively. SOAP and WSDL are also based on XML. Because these technologies are an important foundation to the information in this handbook, a brief description of each is provided here.

XML

eXtensible Markup Language (XML) is a self-describing markup language that allows authors to define data structures that can be understood by multiple systems. The following example shows a complex element (`PersonName`) that contains three simple elements. The tags serve as labels to describe the information contained in the elements.

```
<PersonName>
   <FirstName>Jonathan</FirstName>
   <MiddleName>Q</MiddleName>
   <LastName>Smith</LastName>
</PersonName>
```

XML document content is standardized through the application of schema documents in which the data type, occurrence, and allowable content of the document are defined. W3C XML Schema is the standard for describing XML document content and is the schema mechanism used by Banner Web services.

Within Web services, XML Schema documents define the structure and content of messages that are exchanged between systems. These message "payloads" can be application-specific, based on other industry-standard definitions, or a combination of

both. Ellucian complies with accepted standards groups for common XML Schema-based definitions when defining enterprise-level message schemas. Ellucian intends to develop and propose new standards where none exists.

SOAP

Simple Object Access Protocol (SOAP) defines a standard communications protocol for Web services. It provides a simple, consistent mechanism that allows one system to send an XML message to another system.

WSDL

Web Services Description Language (WSDL) is an XML format that describes a Web service as follows:

- Location of a Web service application
- Operations that a Web service supports
- Valid responses expected for each operation request

How do Banner Web services work?

Java-based adapters expose Banner functions as Web services. This exposure makes the Banner functions available to external systems using the SOAP protocol over HTTP/HTTPS. External systems interact with the Web services, which in turn are supported by Banner APIs. This layered approach provides an insulating buffer between external systems and Banner. External systems do not interact with Banner directly, but rather exchange XML messages with the exposed Web services.

The Banner Web Services Adapters support the synchronous, request-reply message exchange pattern as follows:

1. The external system requests a service of Banner by sending an XML message to the Web service endpoint that is exposed by the adapter. The message contains the information required for Banner to service the request.
2. The Banner Web Services 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 external system.

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

Each Web service is independent. A business process, however, can invoke several Web services during the process.

The Banner Cardholder Event Publisher also uses Web services, in a different way, to integrate Banner with external systems. Rather than receive requests from external systems, the Banner Cardholder Event Publisher pushes data from Banner to external systems when specified data elements change. The Publisher provides near real-time publication of data for which Banner is authoritative.

Example uses of Banner Web services

Banner Web services connect external systems with Banner. These connections enforce vital business processes and allow data to be centrally maintained. Data from one system can be used as input to the other. Data updates in one system can initiate data update in the other.

Banner Web services support integration with housing systems and campus card systems. The available services, however, are not limited to these systems and can be used for other business needs.

Integration with campus card systems

Campus cards are electronic or magnetic cards that are issued to students, staff, faculty, and other constituents. Institutions use campus card systems to identify individuals, allow physical access to institution facilities, provide purchasing capabilities at campus point-of-purchase venues, and provide access to other institutional offerings such as libraries.

Web services can be used to provide real-time interaction between third-party campus card systems and Banner. Banner is the authoritative source for constituent information. Campus card systems are the authoritative source for card-access related data.

Consider the following scenario. A student visits the campus card office to request a campus card. The administrative user cannot find the student in the database of the campus card system and takes the following action:

1. Requests a search of Banner to locate a record for the student (via the GetPersonIdentity Web service).

2. Once the record is located, requests additional information about the student (via the GetEligibleCardholder Web service). This information can be used to create a record in the campus card system for the student.

Integration with housing systems

Institutions use housing systems to accept applications for housing, assign applicants to residence halls and related services, and manage campus residence facilities.

Web services can be used to provide real-time interaction between third-party housing systems and Banner. Banner is the authoritative source for constituent information. Housing systems are the authoritative source for housing and resident-related data.

Consider the following scenario. A student visits the residential housing office to apply for housing. The administrative user cannot find the student in the database of the housing system and takes the following action:

1. Requests a search of Banner to locate a record for the student (via the GetPersonIdentity Web service).

2. Once the record is located, requests information to determine if the student is eligible for residential housing (via the GetHousingApplicantEligibility Web service).

3. If the student is eligible, requests additional information that can be used to create a record in the housing system for the student (via the GetHousingApplicantProfile Web service).

Supporting components

The following components work together to provide Web services-based integration with Banner:

- Banner Translation Service
- Banner Web Services Adapters
- Banner Cardholder Event Publisher

Banner Translation Service

Data in Banner is often constrained by lists of valid values. These valid values are stored in support tables and can be configured by your institution. As a result, the data might not be appropriate or usable by external systems. Banner Translation Service converts institution-specific data values in Banner to standard values that external systems can recognize and use.

Refer to the *Banner Translation Service Installation and Administration Guide* for more details.

Banner Web Services Adapters

The Banner Web Services Adapters expose Banner functions as Web services. An adapter can be configured to expose any number of defined Web services. Two sample configurations are provided as J2EE compatible enterprise archive files. One configuration exposes Web services that external campus card systems need to integrate with Banner. Another configuration exposes Web services that external housing systems need to integrate with Banner. These adapters are referred to as the Banner Web Services Adapter for Campus Card Systems and the Banner Web Services Adapter for Housing Systems, respectively.

The adapters refer to an XML document for configuration information. This in-memory singleton class is built from a preconfigured XML file (`process-config.xml`). Configuration information includes the following elements:

- Data source that identifies the Banner database for performing database transactions
- Banner Translation Service lookup, regular expression, and delimiters that perform static and dynamic value translations
- Mappings for message types identified by the incoming root XML element to a PL/SQL packaged procedure to process the request
- List of XSL transformations used to convert UDC schema document instances to Banner schema document instances, and vice versa

Banner Cardholder Event Publisher

The Banner Cardholder Event Publisher publishes data from Banner to external systems when cardholder data changes in Banner tables. The following processing occurs:

1. Oracle Streams captures the table changes and publishes a corresponding Banner Identity event to the Campus Card Event Topic.
2. The Banner Cardholder Event Publisher reads events posted to this topic, retrieves Banner cardholder data, transforms the retrieved data to the proper format, and

publishes SyncEligibleCardholder messages to a campus card system's exposed Web service endpoint that supports the SyncEligibleCardholder interface and SOAP binding.

The Banner Cardholder Event Publisher is delivered as a J2EE compatible enterprise archive file and works with the Banner Web Services Adapter for Campus Card Systems.

Person identifiers

Some Banner Web services contain person data exchange messages. These Web services use three distinct person identifiers. Request messages can contain any of these identifiers. Response and synchronization messages contain all three identifiers.

Person identifiers are represented in XML using the HR-XML EntityIDType structure (see http://ns.hr-xml.org/2_5/HR-XML-2_5/CPO/EntityIdentifiers.html). This structure contains the `/IdValue/@name` attribute, which specifies the identifier's type.

The following person identifiers are exposed:

@name Attribute Value	Banner Data Value
Banner UID	`SPRIDEN_ID`
IMSID	`GOBSRID_SOURCED_ID`
LogonID	`SPRIDEN_ID` or `GOBTPAC_EXTERNAL_USER`

The Banner UID and IMSID person identifiers are statically mapped. The LogonID, however, depends on a setting on the Crosswalk Validation Form (GTVSDAX).

Note

Banner Intcomp previously used the GTVSDAX setting to determine what data was sent to the Luminis Platform as a logon ID. Banner Intcomp now uses a setting on the Integration Configuration Settings Form (GORICCR). Banner Web services, however, still use the GTVSDAX setting to determine the proper value to populate for the LogonID.

This GTVSDAX setting is not delivered as seed data. You must manually create the following row on GTVSDAX. It is recommended that you define this value to match the corresponding value in GORICCR.

Internal Code:	*USERSOURCE*
Internal Group:	*INTCOMP*
External Code:	Source of the LogonID: *I* Current ID from `SPRIDEN_ID` *U* Third-party ID from GOATPAD
Code Description:	Description for this rule.
Limit on GTVSDAX:	One
Message Limit:	One per `PersonIdentity` element

Customization

Your institution can customize rules and settings for individual Banner Web services to meet your specific needs:

- Use the Business Rules Form (GORRSQL) to define roles for campus cards and campus housing.
- Use the Crosswalk Validation Form (GTVSDAX) to identify what data should be selected from the Banner database.

Customizations are described in “Setup requirements” for each Banner Web service.

Translations

There are two ways that external values are translated to valid Banner values:

- *Dynamic* translations are stored in a database and retrieved by the Banner Translation Service when it processes incoming and outgoing messages. The translations must be customized by your institution to match the values stored in Banner validation tables.
- *Static* translations are coded in XSL style sheets to provide translations to standard Banner values that cannot be customized. For example, translating an external value *Yes* to the value *Y* needed by Banner APIs is a static translation. These translations should not be changed.

Dynamic and static translations that occur for each Banner Web service are described in “Translations” for each Banner Web service.

Available documentation

This handbook describes the messages, message mapping to Banner, intended usage, setup requirements, and translations for the Banner Web services that support integration with campus card systems and housing systems.

The following documents provide more information on Banner Web services:

- The *Banner Web Services Installation Guide* provides detailed information on installing Banner Web Services Adapters and the Banner Cardholder Event Publisher.
- The *Banner Translation Service Installation and Administration Guide* provides information on installing and administering the Banner Translation Service, a prerequisite component for exposing the Banner Web services.

2 AddEntityAddress

The AddEntityAddress Web service allows external systems to request the creation of an entity address in Banner®. An “entity address” is an address that is related to a person or organization in Banner. The `EntityAddress` XML structure contains address information and a reference to the entity that owns the address.

Message exchange

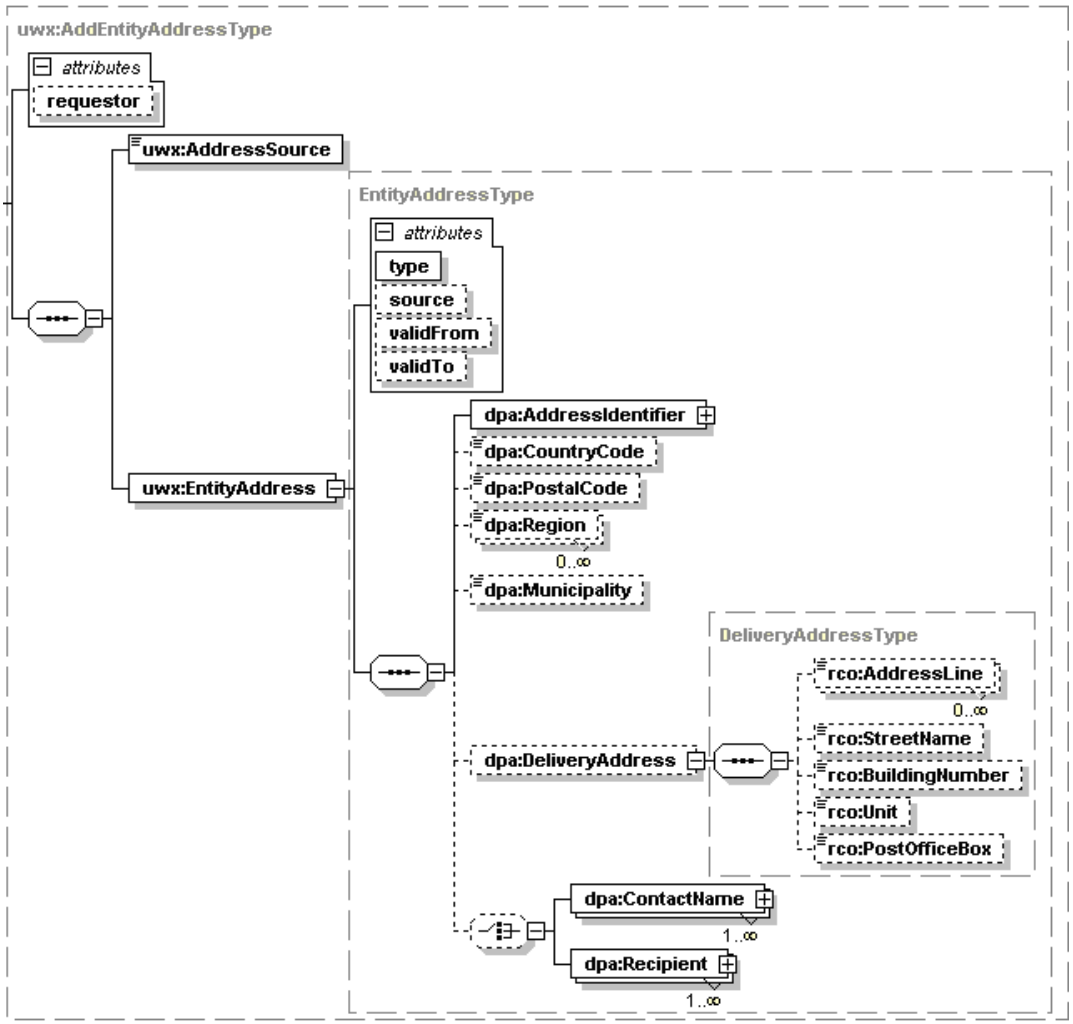
The AddEntityAddress Web service exchanges the following messages:

- AddEntityAddress
- ConfirmAddEntityAddress

AddEntityAddress

An external system uses the AddEntityAddress message to request the creation of an address in Banner. Only one address is permitted in each message.

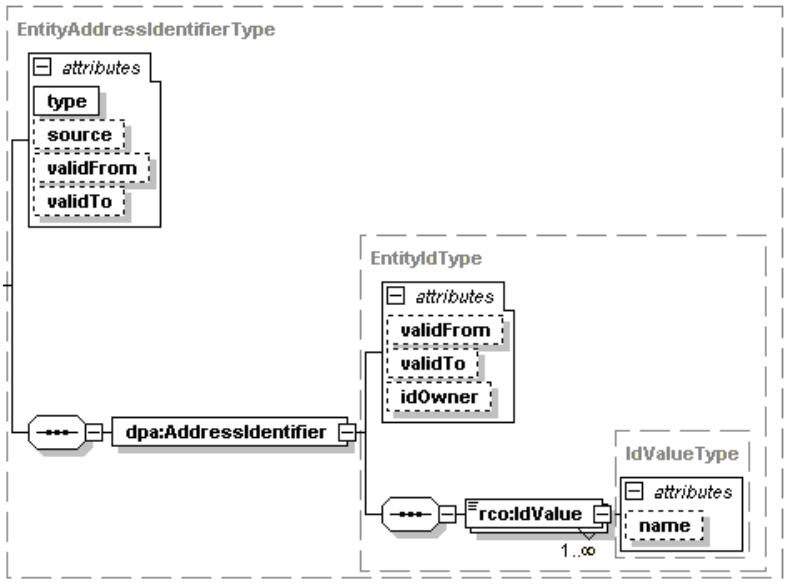
The following diagram shows the structure of the AddEntityAddress message schema.



ConfirmAddEntityAddress

If the entity address is successfully added to Banner, the AddEntityAddress Web service responds with a ConfirmAddEntityAddress message. This message contains information that uniquely identifies the address in Banner so that it can be referenced in subsequent messages (see Chapter 5, “ExpireEntityAddress”).

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



SOAP fault messages

If a valid response cannot be created as a ConfirmAddEntityAddress message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `AddressIdentifier` provided in the AddEntityAddress message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- The address specified in the AddEntityAddress message already exists in Banner for the entity.
- The address type specified on GTVSDAX is not a valid address type, or no GTVSDAX rule exists.
- Completion of the request violates Banner business rules required by the `gb_address` business entity API. For more information about the API, refer to the *Banner General Technical Reference Manual*.
- A network, database, or other technical issue occurs.
- The applicable GTVSDAX rows for this Banner Web service have the value *UPDATE ME* in the **External Code** field.

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.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

AddEntityAddress

Element/Attribute	Database Mapping
AddEntityAddress	
@requestor	NA
AddressSource	GTVSDAX_TRANSLATION_CODE to derive SPRADDR_ASRC_CODE
EntityAddress	
@source	SPRADDR_DATA_ORIGIN
@validFrom	SPRADDR_FROM_DATE
@type	GTVSDAX_TRANSLATION_CODE to derive SPRADDR_ATYP_CODE
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
CountryCode	SPRADDR_NATN_CODE
PostalCode	SPRADDR_ZIP

Element/Attribute	Database Mapping
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
DeliveryAddress	
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA

ConfirmAddEntityAddress

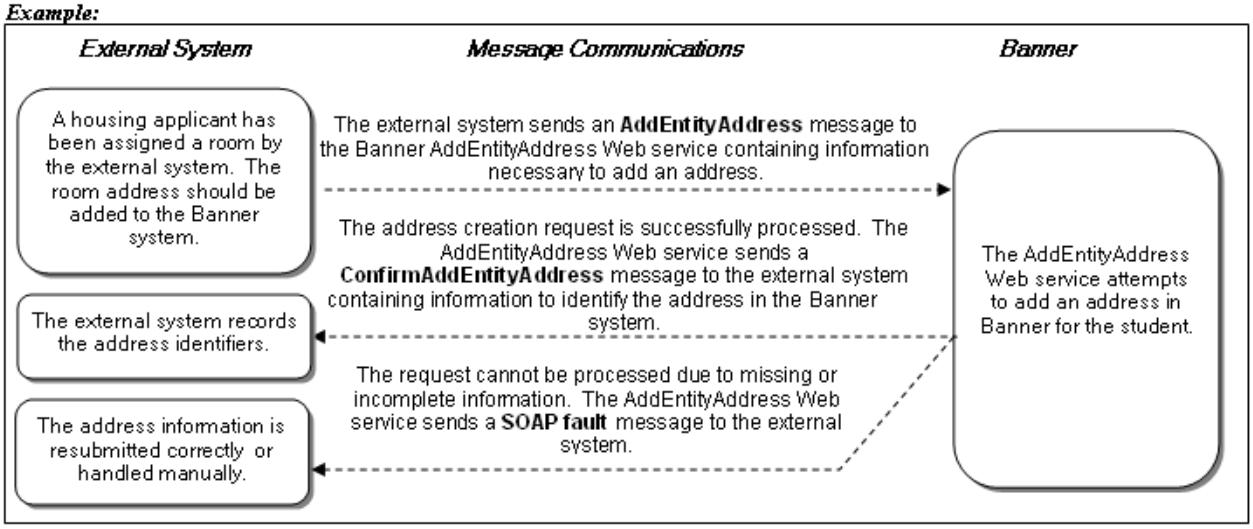
Element/Attribute	Database Mapping
ConfirmAddEntityAddress	
EntityAddressIdentifier	

Element/Attribute	Database Mapping
@source	SPRADDR_DATA_ORIGIN
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA

Intended usage

External systems can use the AddEntityAddress Web service to create a new address in Banner.

Example:



Setup requirements

Two settings on the Crosswalk Validation Form (GTVSDAX) specify Banner-specific values to use when creating a new entity address. Initial GTVSDAX records provided with this Web service should be installed in Banner with the other components. You can access and modify these rules. After completing the setup activities, your technical staff should run the `housing_check.sql` utility script to verify that no unused GTVSDAX records remain. If the utility finds GTVSDAX records with an **External Code** set to *UPDATE ME*, you should remove or update these records before using this Web service.

Address type code

This GTVSDAX setting specifies the `SPRADDR_ATYP_CODE` that the AddEntityAddress Web service uses to create new addresses in Banner.

The `@type` attribute in the AddEntityAddress message contains a value provided by the external system describing the type of address. The AddEntityAddress Web service searches for a GTVSDAX rule where the provided address type matches the translation value and uses the external code as the Banner address type code. If the provided address type is not found as a translation value on GTVSDAX, no address is created.

Use the following information to create or modify rules on GTVSDAX. You should establish one rule for each address type provided by an external system. An unlimited number of address type rules can be set up, but only one is allowed for each external address type value.

Internal Code:	*INTEG*
Internal Group:	*ADDRTYPE*
Translation Code:	*RESIADDRESS*, etc. The partner system provides this value. *RESIADDRESS* is used for all addresses created as a result of room assignment processing in the external housing system. You can also set up a second address type for residence addresses (*RESIADDRESS2*, for example) to accommodate room changes when the resident has access to more than one room on the same day.
External Code:	Address type assigned to the new address (appropriate value from `STVATYP_CODE`)
Code Description:	Description of this rule
Limit on GTVSDAX:	One rule per external address type
Message Limit:	One address per message

Address source code

This GTVSDAX setting specifies the `SPRADDR_ASRC_CODE` used to create an address. The AddEntityAddress Web service does not require this rule to create new addresses in Banner, but this rule can be used to record the source of the address.

The `AddressSource` element in the AddEntityAddress message contains a value provided by the external system describing the source of the address to be created. The AddEntityAddress Web service searches for a GTVSDAX rule where the provided address source matches the translation value and uses the external code as the Banner address source code. If the provided address source is not found as a translation value on GTVSDAX or is not provided in the message, the address is created without a source code.

Use the following information to create or modify rules on GTVSDAX. You can establish one rule for each address source provided by an external system. An unlimited number of address source rules can be set up, but only one is allowed for each external address source value.

Internal Code:	*INTEG*
Internal Group:	*ADDRSRCE*
Translation Code:	*ADIRONDACK*, *RMS*, etc. The partner system provides this value.
External Code:	Address source assigned to the new address (appropriate value from `STVASRC_CODE`)
Code Description:	Description of this rule
Limit on GTVSDAX:	One rule per external address source
Message Limit:	One address per message

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

AddEntityAddress

The following dynamic translation must be customized:

Element	Enterprise Value	Transformed to Banner Value
`dpa:CountryCode`	Value provided by external system	Value from `STVNATN_CODE`

There are no static translations for the AddEntityAddress message.

ConfirmAddEntityAddress

There are no dynamic or static translations for the ConfirmAddEntityAddress message.

3 AddStudentAccount

Three Web services are available for creating student Accounts Receivable transactions in Banner®. Your institution can determine the level of control exercised over each type of transaction. Calling each Web service results in a different context used to process the transaction:

AddStudentAccountTransaction	Provides full editing by business rules as if data were entered on a Banner form (for example, the Student Account Detail Form (TSADETL)).
AddStudentAcctTransSource	Can be used when the transaction should be created using a system-maintained source code (**Transaction Entry Allowed** is not checked on the Charge/Payment Source Code Validation Form (TTVSRCE)).
AddStudentAcctTransSystem	Processes rules-based calculations that are to be recorded without regard to the following: • AR holds • Inactive detail code • System maintained source code • User profile security • Effective date restrictions • Title IV withdrawal status

Each type of transaction to be created should be evaluated to determine the appropriate Web service to use.

These Web services allow external systems to request that a charge or payment be recorded on a student account in Banner. These Web services can be used when an external system calculates charges or records payments, but the institution prefers to keep a consolidated student accounts receivable history in Banner.

Note

These Web services require Banner Accounts Receivable.

Message exchange

The AddStudentAccount Web services exchange the following messages:

- AddStudentAccountTransaction
- AddStudentAcctTransSource
- AddStudentAcctTransSystem
- ConfirmAddStudentAccountTransaction

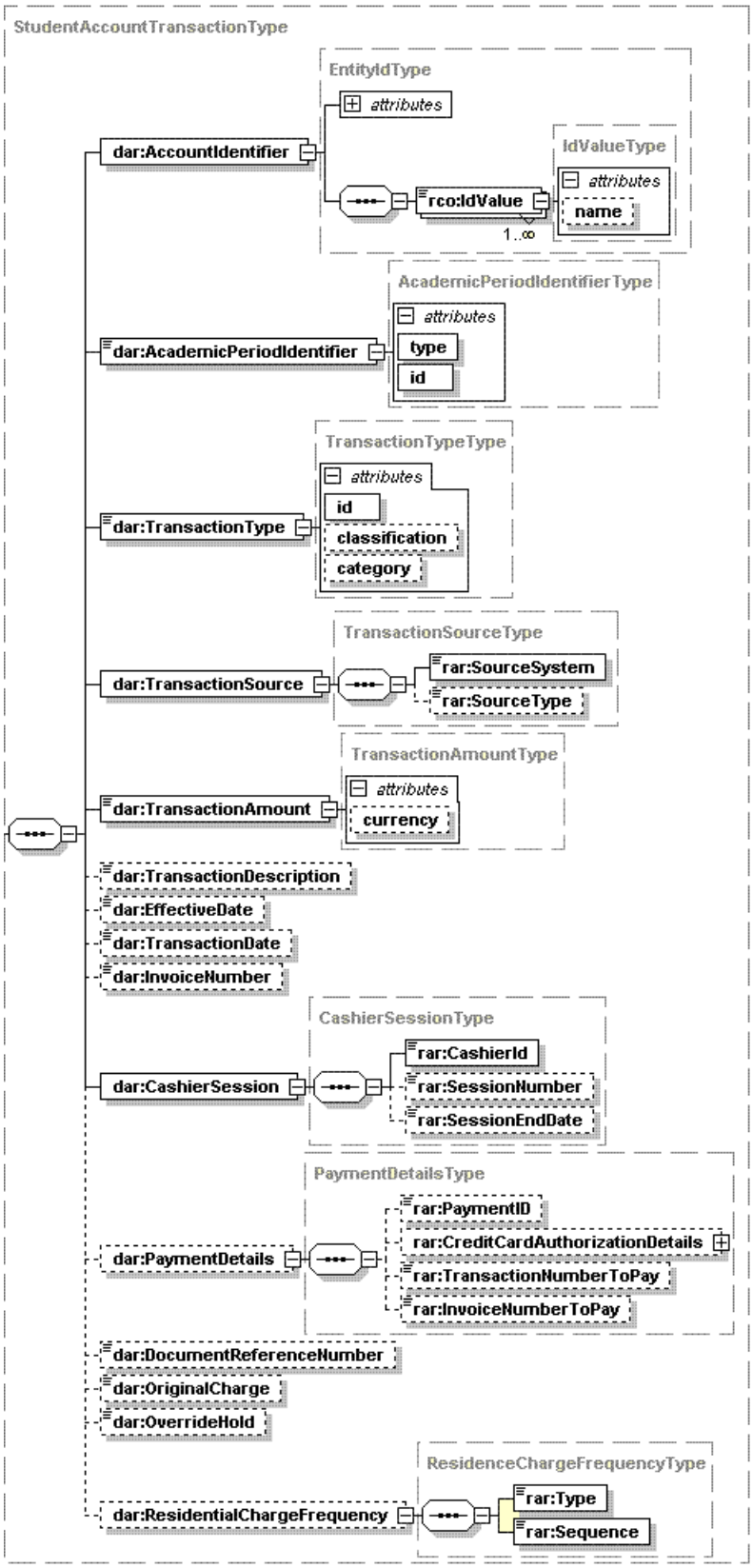
AddStudentAccountTransaction, AddStudentAcctTransSource, and AddStudentAcctTransSystem

An external system uses these messages to request that a transaction record be added to a student's Accounts Receivable history in Banner. The messages include elements required to create this record. Only one transaction is permitted in each message.

Note

Creating a transaction in Banner impacts the financial accounting of the institution. If a transaction is recorded in both the external system and Banner, valid accounting procedures are required to accurately reflect and reconcile the actual amount on record at the institution. ■

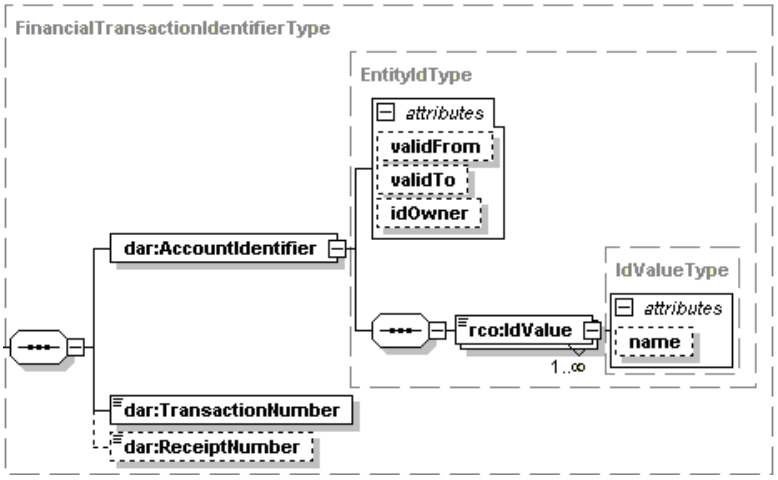
The following diagram shows the structure of the AddStudentAccountTransaction message schema.


StudentAccountTransactionType
EntityIdType
attributes
dar:AccountIdentifier
IdValueType
attributes
rco:IdValue
name
1..∞
AcademicPeriodIdentifierType
attributes
dar:AcademicPeriodIdentifier
type
id
TransactionTypeType
attributes
id
dar:TransactionType
classification
category
TransactionSourceType
dar:TransactionSource
rar:SourceSystem
rar:SourceType
TransactionAmountType
attributes
dar:TransactionAmount
currency
dar:TransactionDescription
dar:EffectiveDate
dar:TransactionDate
dar:InvoiceNumber
CashierSessionType
dar:CashierSession
rar:CashierId
rar:SessionNumber
rar:SessionEndDate
PaymentDetailsType
dar:PaymentDetails
rar:PaymentID
rar:CreditCardAuthorizationDetails
rar:TransactionNumberToPay
rar:InvoiceNumberToPay
dar:DocumentReferenceNumber
dar:OriginalCharge
dar:OverrideHold
ResidenceChargeFrequencyType
dar:ResidentialChargeFrequency
rar:Type
rar:Sequence

ConfirmAddStudentAccountTransaction

If the transaction is successfully added to the student's account in Banner, the AddStudentAccountTransaction, AddStudentAcctTransSource, or AddStudentAcct TransSystem Web service responds with a ConfirmAddStudentAccountTransaction message.

The following diagram shows the structure of the ConfirmAddStudentAccount Transaction message schema.



SOAP fault messages

If a valid response cannot be created as a ConfirmAddStudentAccountTransaction message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `AccountIdentifier` provided in the AddStudentAccountTransaction, AddStudentAcctTransSource, or AddStudentAcctTransSystem message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- The AddStudentAccountTransaction, AddStudentAcctTransSource, or AddStudentAcctTransSystem message contains values that are not recognized by Banner.
- Completion of the request violates Banner business rules required by the `tb_receivable` business entity API. For more information about the API, refer to the *Banner Accounts Receivable TRM Supplement*.
- 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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

AddStudentAccountTransaction, AddStudentAcctTransSource, and AddStudentAcctTransSystem

Element/Attribute	Database Mapping
AddStudentAccountTransaction (AddStudentAcctTransSource, AddStudentAcctTransSystem)	
StudentAccountTransaction	
AccountIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
AcademicPeriodIdentifier	STVTERM_DESC
@id	TBRACCD_TERM_CODE
@type	NA

Element/Attribute	Database Mapping
TransactionType	TBBDETC_DESC
@category	TTVDCAT_DESC
@id	TBRACCD_DETAIL_CODE
@classification	TBBDETC_TYPE_IND
TransactionSource	
SourceSystem	TBRACCD_DATA_ORIGIN
SourceType	TBRACCD_SRCE_CODE
TransactionAmount	TBRACCD_AMOUNT
@currency	GTVCURR_TITLE
TransactionAmount	TBRACCD_FOREIGN_AMOUNT
@currency	GTVCURR_TITLE
TransactionDescription	TBRACCD_DESC
EffectiveDate	TBRACCD_EFFECTIVE_DATE
TransactionDate	TBRACCD_TRANS_DATE
InvoiceNumber	TBRACCD_INVOICE_NUMBER
CashierSession	
CashierId	TBRACCD_USER
SessionNumber	TBRACCD_SESSION_NUMBER
SessionEndDate	TBRACCD_CSHR_END_DATE
PaymentDetails	
PaymentID	TBRACCD_PAYMENT_ID
TransactionNumberToPay	TBRACCD_TRAN_NUMBER_PAID
InvoiceNumberToPay	TBRACCD_INV_NUMBER_PAID
DocumentReferenceNumber	TBRACCD_DOCUMENT_NUMBER
OriginalCharge	TBRACCD_ORIG_CHG_IND
OverrideHold	TBRACCD_OVERRIDE_HOLD

Element/Attribute	Database Mapping
ResidentialChargeFrequency	
Type	TBRACCD_LOC_MDT
Sequence	TBRACCD_LOC_MDT_SEQ

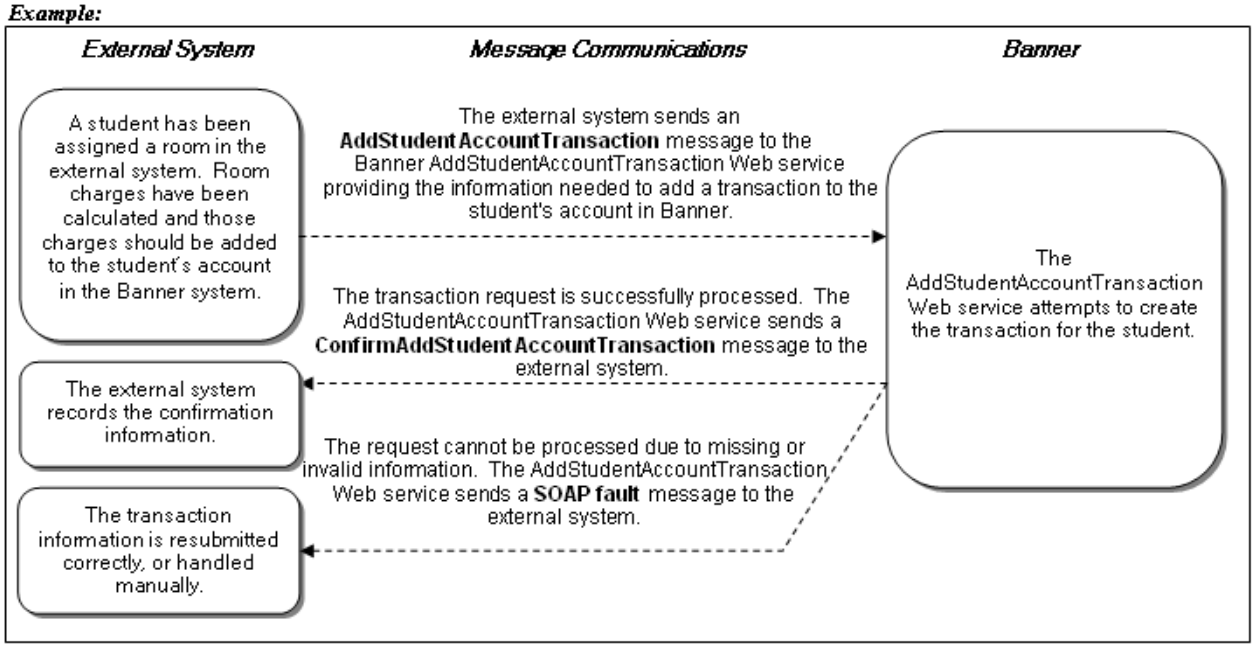
ConfirmAddStudentAccountTransaction

Element/Attribute	Database Mapping
ConfirmAddStudentAccountTransaction	
AccountTransactionIdentifier	
AccountIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	/A
TransactionNumber	TBRACCD_TRAN_NUMBER
ReceiptNumber	TBRACCD_RECEIPT_NUMBER

Intended usage

External systems can use any of the three Web services (AddStudentAccount Transaction, AddStudentAcctTransSource, or AddStudentAcctTransSystem) to record a charge or payment for a student in Banner.

Example:



Setup requirements

Your institution must coordinate the use of Banner-specific values with an external system. The external system must provide the values required by the transaction.

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

AddStudentAccountTransaction, AddStudentAcctTransSource, and AddStudentAcctTransSystem

The following dynamic translations must be customized:

Element	Enterprise Value	Transformed to Banner Value
`rar:CardType`	Value provided by external system	Value from `GTVCCRD_CODE`
`dar:TransactionAmount/ @currency`	Value from `GTVCURR_TITLE` or ISO currency codes	Value from `GTVCURR_CURR_CODE`

The following static translations are translated automatically and should not be changed:

Element	Enterprise Value	Transformed to Banner Value
`OriginalCharge`	*Yes* *No*	*Y* *N*
`OverrideHold`	*Yes* *No*	*Y* *N*
`ResidentialChargeFrequency/ Type`	*Monthly* *Term* *Daily*	*M* *T* *D*

ConfirmAddStudentAccountTransaction

There are no dynamic or static translations for the ConfirmAddStudentAccount Transaction message.

4 AddStudentDeposit

The AddStudentDeposit Web service allows external systems to create a student deposit record in Banner®. A student deposit is a sum of money given as security for an item or service. These funds may be released later to cover damages charged to a student's account or remitted to the student at the end of a specified period.

Note

This Web service requires Banner Accounts Receivable.

Message exchange

The AddStudentDeposit Web service exchanges the following messages:

- AddStudentDeposit
- ConfirmAddStudentDeposit

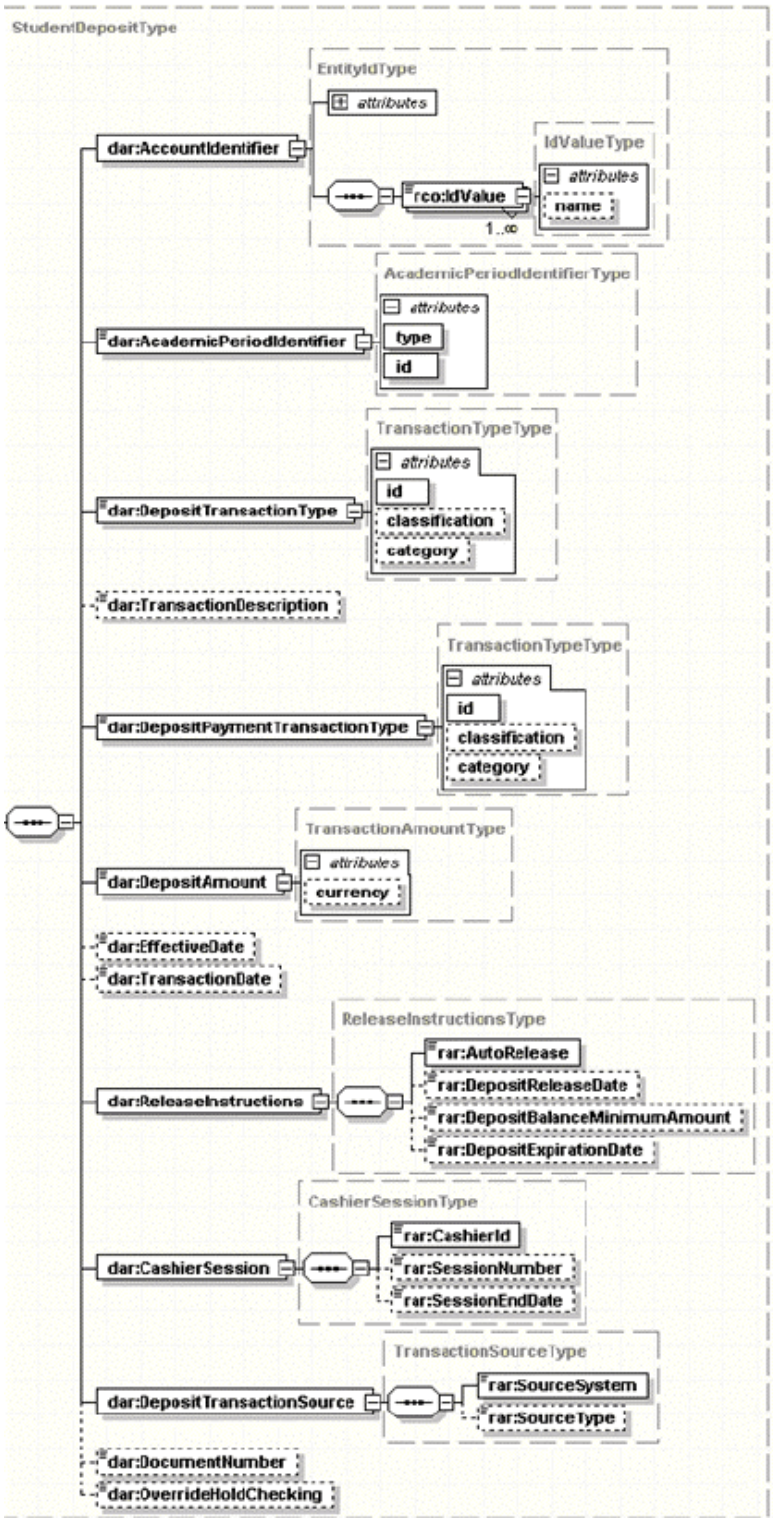
AddStudentDeposit

An external system uses the AddStudentDeposit message to request the creation of a student deposit in Banner. You must provide valid Banner values to create a deposit. These values must be provided to the external system as part of the installation and configuration of the integration solution.

Note

Creating a transaction in Banner impacts the financial accounting of the institution. If a deposit is recorded in both the external system and Banner, valid accounting procedures are required to accurately reflect and reconcile the actual amount on record at an institution.

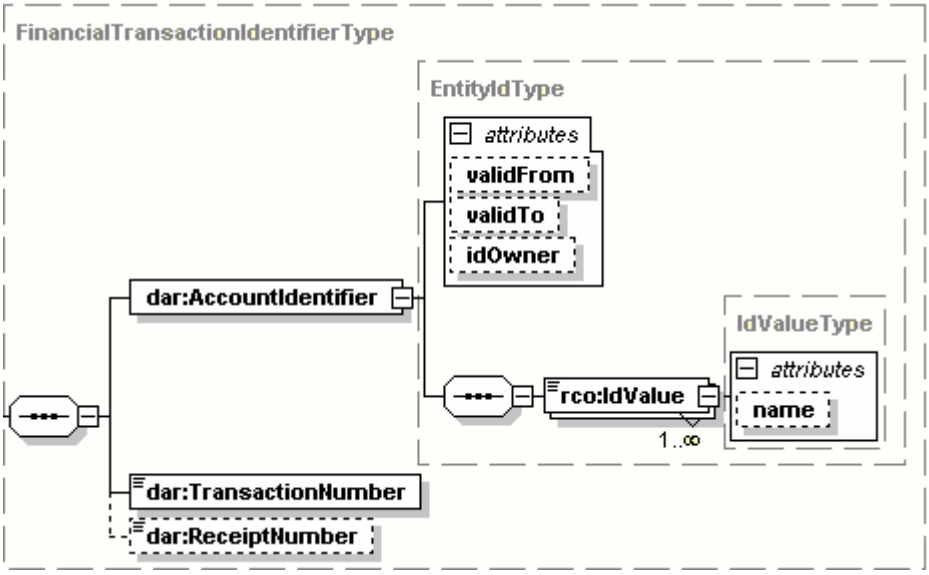
The following diagram shows the structure of the AddStudentDeposit message schema.


StudentDepositType
EntityIdType
attributes
dar:AccountIdentifier
IdValueType
attributes
rco:IdValue
name
1..∞
AcademicPeriodIdentifierType
attributes
dar:AcademicPeriodIdentifier
type
id
TransactionTypeType
attributes
dar:DepositTransactionType
id
classification
category
dar:TransactionDescription
TransactionTypeType
attributes
dar:DepositPaymentTransactionType
id
classification
category
TransactionAmountType
attributes
dar:DepositAmount
currency
dar:EffectiveDate
dar:TransactionDate
ReleaseInstructionsType
dar:ReleaseInstructions
rar:AutoRelease
rar:DepositReleaseDate
rar:DepositBalanceMinimumAmount
rar:DepositExpirationDate
CashierSessionType
dar:CashierSession
rar:CashierId
rar:SessionNumber
rar:SessionEndDate
TransactionSourceType
dar:DepositTransactionSource
rar:SourceSystem
rar:SourceType
dar:DocumentNumber
dar:OverrideHoldChecking

ConfirmAddStudentDeposit

If the deposit is successfully added to the student's account in Banner, the AddStudentDeposit Web service responds with a ConfirmAddStudentDeposit message.

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



SOAP fault messages

If a valid response cannot be created as a ConfirmAddStudentDeposit message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `AccountIdentifier` provided in the AddStudentDeposit message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- The AddStudentDeposit message contains values that are not recognized by Banner.
- Completion of the request violates Banner business rules required by the `tb_deposit` business entity API. For more information about the API, refer to the *Banner Accounts Receivable TRM Supplement*.
- 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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

AddStudentDeposit

Element/Attribute	Database Mapping
AddStudentDeposit	
StudentDeposit	
AccountIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
AcademicPeriodIdentifier	STVTERM_DESC
@id	TBRDEPO_TERM_CODE
@type	NA
DepositTransactionType	TBBDETC_DESC
@category	TTVDCAT_DESC
@id	TBRDEPO_DETAIL_CODE_DEPOSIT

Element/Attribute	Database Mapping
@classification	TBBDETC_TYPE_IND
TransactionDescription	TBRDEPO_DESC
DepositPaymentTransactionType	TBBDETC_DESC
@category	TTVDCAT_DESC
@id	TBRDEPO_DETAIL_CODE_PAYMENT
@classification	TBBDETC_TYPE_IND
DepositAmount	TBRDEPO_AMOUNT
@currency	GTVCURR_TITLE
DepositAmount	TBRDEPO_FOREIGN_AMOUNT
@currency	GTVCURR_TITLE
EffectiveDate	TBRDEPO_EFFECTIVE_DATE
TransactionDate	TBRDEPO_TRANS_DATE
ReleaseInstructions	
AutoRelease	TBRDEPO_AUTO_RELEASE_IND
DepositReleaseDate	TBRDEPO_RELEASE_DATE
DepositBalanceMinimumAmount	TBRDEPO_MIN_AMOUNT
DepositExpirationDate	TBRDEPO_EXPIRATION_DATE
CashierSession	
CashierId	TBRDEPO_USER
SessionNumber	TBRDEPO_SESSION_NUMBER
SessionEndDate	TBRDEPO_CSHR_END_DATE
DepositTransactionSource	
SourceSystem	TBRDEPO_DATA_ORIGIN
SourceType	NA
DocumentNumber	TRBDEPO_DOCUMENT_NUMBER
OverrideHoldChecking	NA

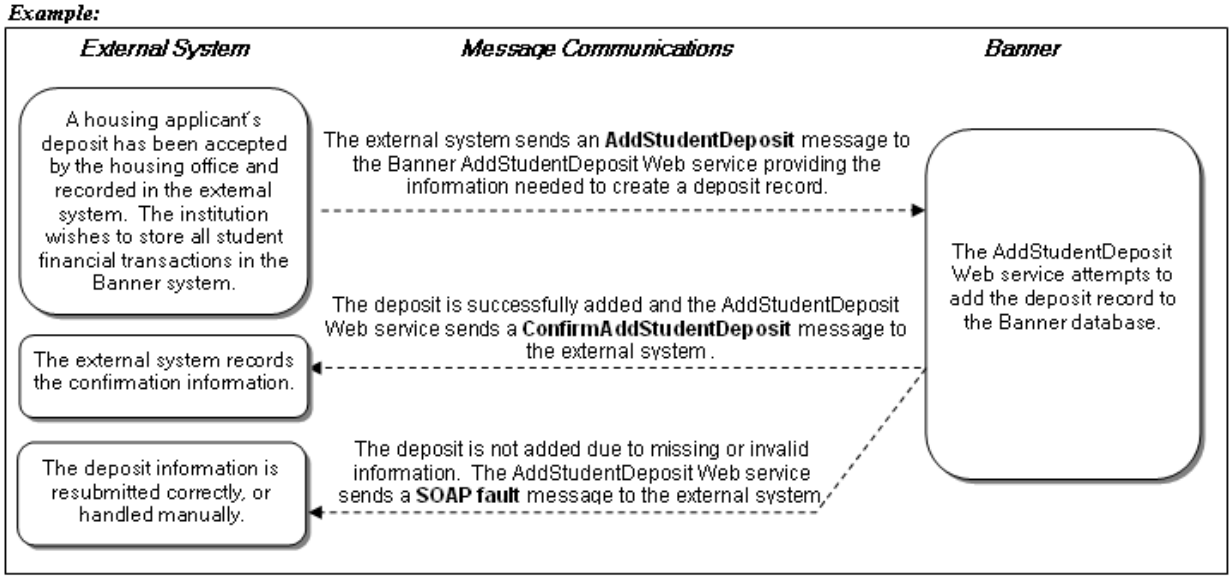
ConfirmAddStudentDeposit

Element/Attribute	Database Mapping
ConfirmAddStudentDeposit	
StudentDepositIdentifier	
AccountIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
TransactionNumber	TBRDEPO_TRAN_NUMBER
ReceiptNumber	TBRDEPO_RECEIPT_NUMBER

Intended usage

External systems can use the AddStudentDeposit Web service to create a deposit record for a student in Banner.

Example:



Setup requirements

Your institution must coordinate the use of Banner-specific values with an external system. The external system must provide the values required by the transaction.

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

AddStudentDeposit

The following dynamic translation must be customized:

Element	Enterprise Value	Transformed to Banner Value
`dar:DepositAmount/@currency`	Value from `GTVCURR_TITLE` or ISO currency codes	Value from `GTVCURR_CURR_CODE`

The following static translations are translated automatically and should not be changed:

Element	Enterprise Value	Transformed to Banner Value
`AutoRelease`	*Yes*	*Y*
	No	*N*
`OverrideHoldChecking`	*Yes*	*Y*
	No	*N*

ConfirmAddStudentDeposit

There are no dynamic or static translations for the ConfirmAddStudentDeposit message.

5 ExpireEntityAddress

The ExpireEntityAddress Web service allows external systems to request the expiration of an existing address in Banner®. An “entity address” is an address that is related to a person or organization in Banner. The `EntityAddress` XML structure contains address information and a reference to the entity that owns the address.

Addresses are expired in Banner by updating the address end date. If the new end date (`dpa:ExpirationDate`) is the same as the start date for the address in Banner, the address is also inactivated. This processing permits the creation of a new address for the same date.

Message exchange

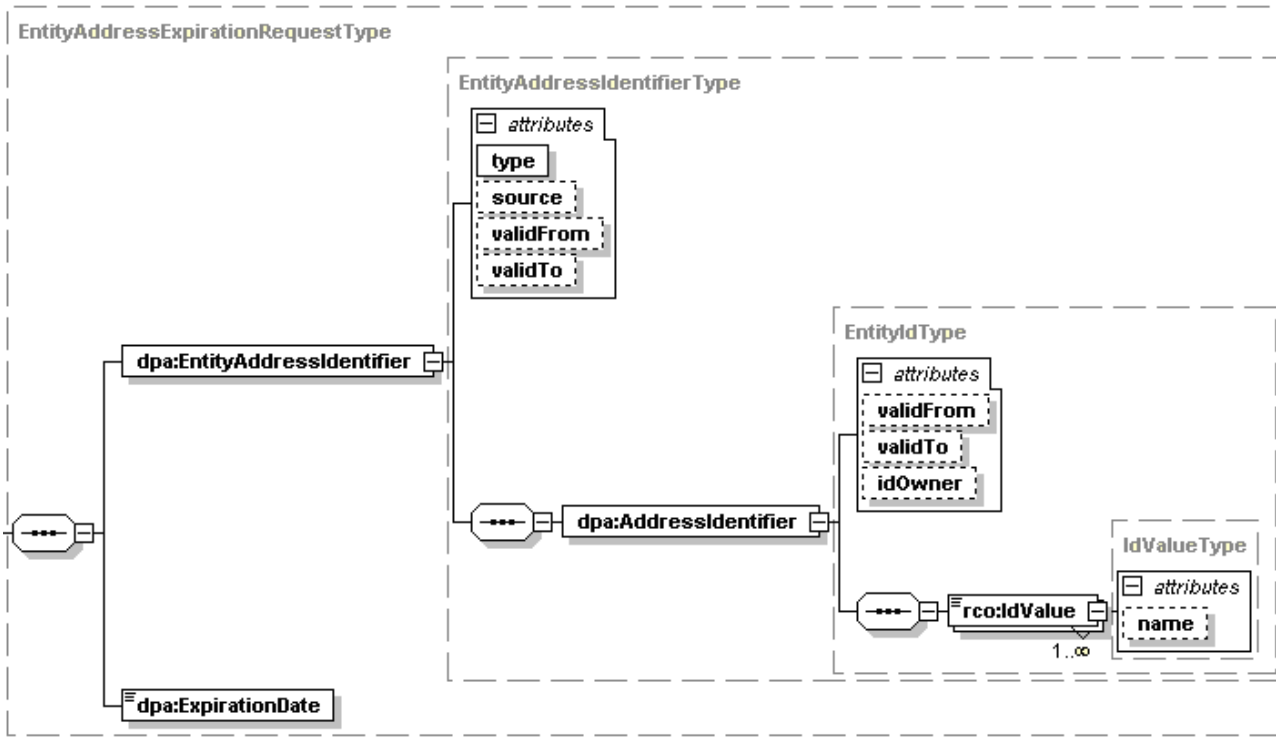
The ExpireEntityAddress Web service exchanges the following messages:

- ExpireEntityAddress
- ConfirmExpireEntityAddress

ExpireEntityAddress

An external system uses the ExpireEntityAddress message to request the expiration of an address in Banner. You can expire only one address with each message. The external system must provide the unique identifier for the address, which includes the Banner ID (or other accepted identifier), the address type code, and the address sequence number.

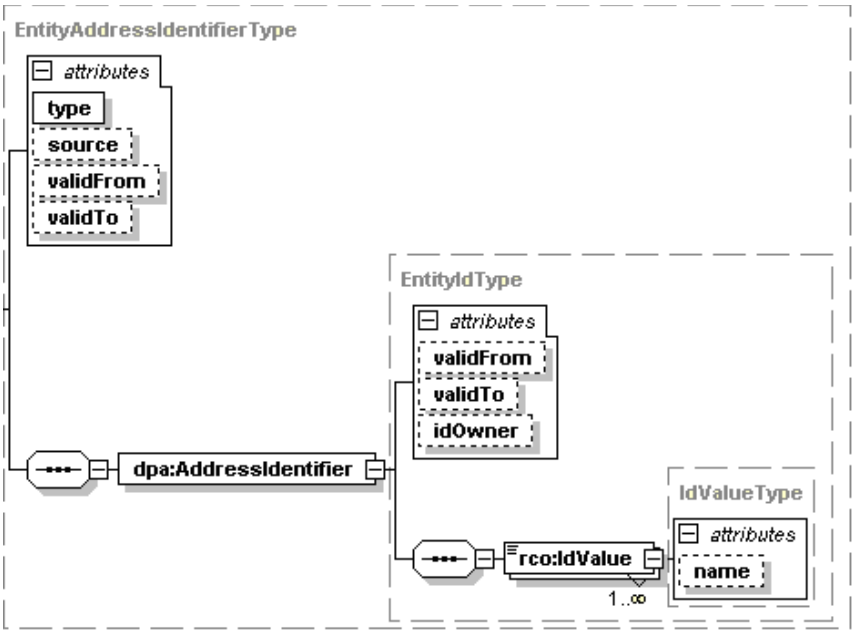
The following diagram shows the structure of the ExpireEntityAddress message schema.



ConfirmExpireEntityAddress

If the entity address is successfully expired in Banner, the ExpireEntityAddress Web service responds with a ConfirmExpireEntityAddress message. This message contains information that uniquely identifies the address and updated address dates.

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



SOAP fault messages

If a valid response cannot be created as a ConfirmExpireEntityAddress message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `AddressIdentifier` provided in the ExpireEntityAddress message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- The address cannot be identified in Banner with the information provided in the ExpireEntityAddress message.
- The address identified in the ExpireEntityAddress message is marked inactive in Banner.
- Completion of the request violates Banner business rules required by the `gb_address` business entity API. For more information about the API, refer to the *Banner General Technical Reference Manual*.
- 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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

ExpireEntityAddress

Element/Attribute	Database Mapping
ExpireEntityAddress	
EntityAddressExpirationRequest	
EntityAddressIdentifier	
@source	SPRADDR_DATA_ORIGIN
@validFrom	NA
@type	STVATYP_DESC
@validTo	NA
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
ExpirationDate	SPRADDR_TO_DATE

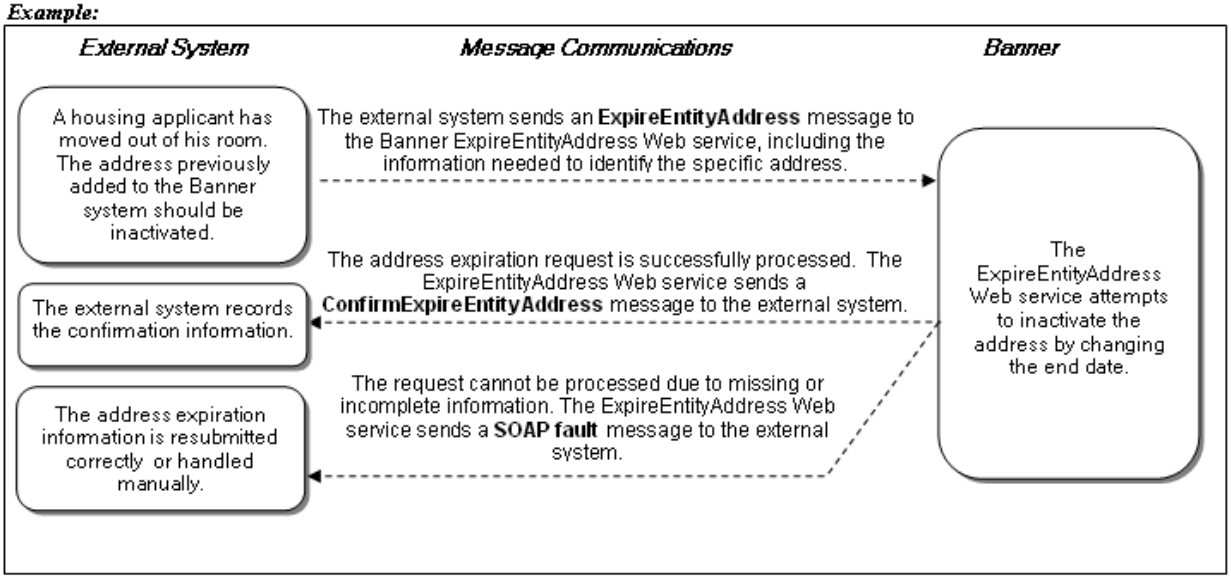
ConfirmExpireEntityAddress

Element/Attribute	Database Mapping
ConfirmExpireEntityAddress	
EntityAddressIdentifier	
@source	SPRADDR_DATA_ORIGIN
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA

Intended usage

External systems can use the ExpireEntityAddress Web service to change the end date for an address in Banner.

Example:



Setup requirements

The ExpireEntityAddress Web service does not require rules on the Crosswalk Validation Form (GTVSDAX) or roles on the Business Rules Form (GORRSQL). However, it does require valid address identifier information used to locate the specific address in Banner. This information is returned to the external system in the ConfirmAddEntityAddress message when that system uses the AddEntityAddress Web service to create a new address in Banner. It is expected that the external system will store this information upon receipt of the ConfirmAddEntity Address message for use when calling the ExpireEntityAddress Web service.

Translations

There are no dynamic or static translations for the ExpireEntityAddress Web service.

6 GetAcademicPeriods

The GetAcademicPeriods Web service allows external systems to request information about Banner® academic terms. The Web service can be called during the external system implementation process or can be used as a real-time request when Banner term information is needed. The Web service is designed for use with Banner Student.

Information about an academic period (term) is provided in the `AcademicPeriod` element in the ShowAcademicPeriods message.

An *academic period* is a period of time used by institutions to bind class schedules, student matriculation, and dormitory residency. In Banner, academic periods are referred to as *terms*. An external system communicating with Banner might be required to provide academic period information as part of another request. If that information is not known by the external system, the GetAcademicPeriods Web service can be used to provide it.

Note

This Web service requires Banner Student.

Message exchange

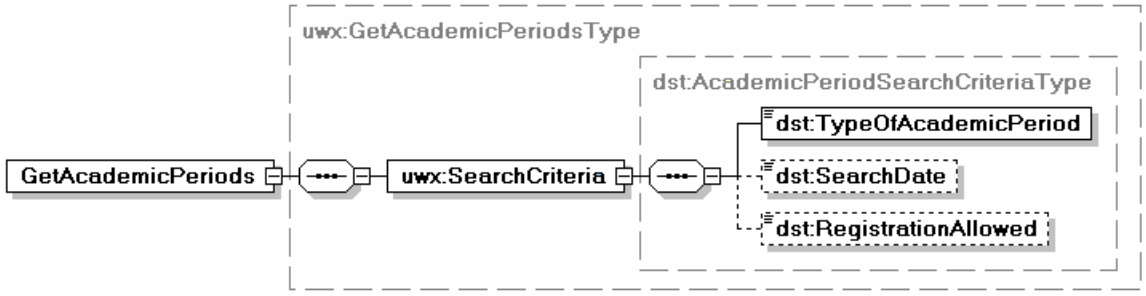
The GetAcademicPeriods Web service exchanges the following messages:

- GetAcademicPeriods
- ShowAcademicPeriods

GetAcademicPeriods

An external system uses the GetAcademicPeriods message to request information from Banner about the academic periods that match a given search criteria. Search criteria include a search date and the ability to select only academic periods for which registration is open.

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



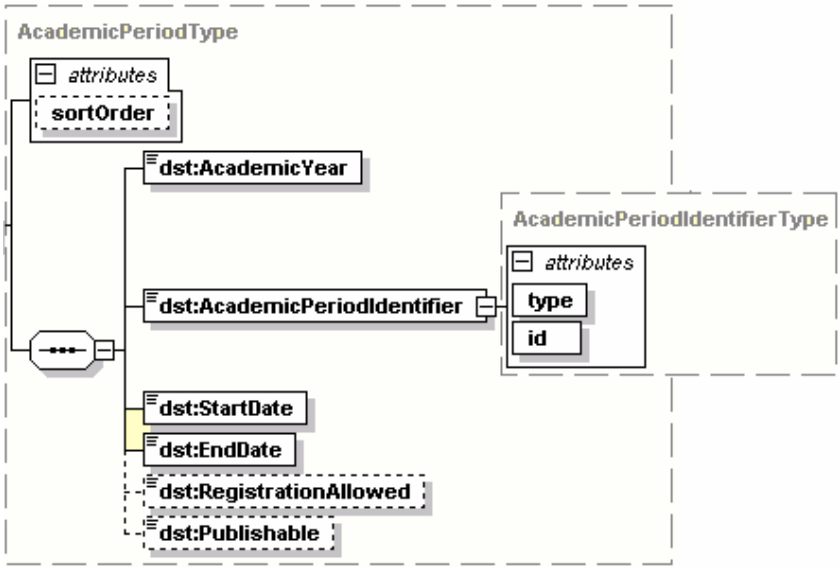
Note

The value in the `TypeOfAcademicPeriod` element should always be *Term*.

ShowAcademicPeriods

The GetAcademicPeriods Web service queries the Banner database and returns any matching academic periods in a ShowAcademicPeriods message. If a `SearchDate` is provided and that date falls between the start and end dates of more than one Banner term, multiple `AcademicPeriod` elements are returned.

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



SOAP fault messages

If a valid response cannot be created as a ShowAcademicPeriods message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The combination of search criteria provided in the GetAcademicPeriods message results in the selection of no terms in Banner.
- 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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

GetAcademicPeriods

Element/Attribute	Database Mapping
GetAcademicPeriods	
SearchCriteria	
TypeOfAcademicPeriod	NA
SearchDate	NA
RegistrationAllowed	SOBTERM_REG_ALLOWED

ShowAcademicPeriods

Element/Attribute	Database Mapping
ShowAcademicPeriods	
AcademicPeriod	
@sortOrder	NA

Element/Attribute	Database Mapping
AcademicYear	STVACYR_DESC
AcademicPeriodIdentifier	STVTERM_DESC
@id	STVTERM_CODE
@type	NA
StartDate	STVTERM_START_DATE
EndDate	STVTERM_END_DATE
RegistrationAllowed	SOBTERM_REG_ALLOWED
Publishable	SOBTERM_PROFILE_SEND_IND

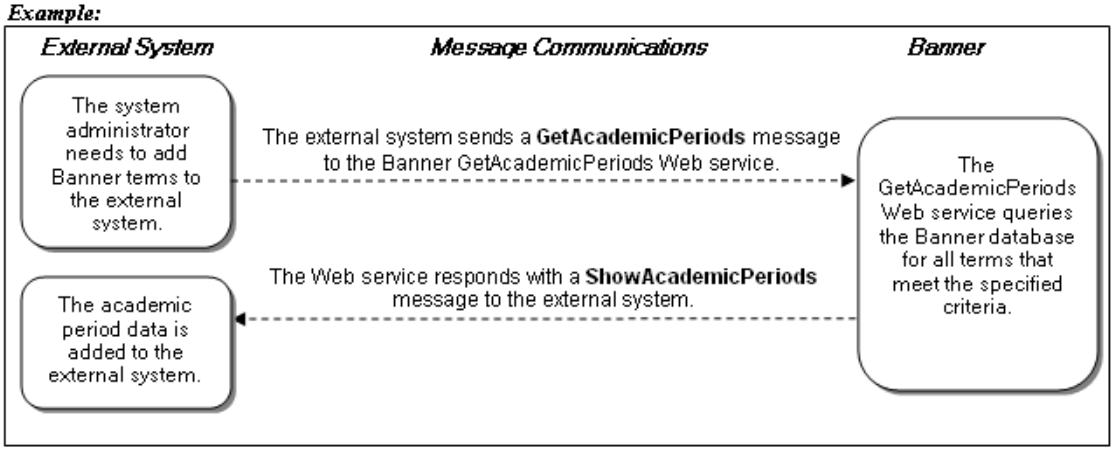
Intended usage

External systems can use the GetAcademicPeriods Web service to synchronize academic periods or search for specific terms in Banner.

Synchronize academic periods

The GetAcademicPeriods Web service can be used to synchronize academic periods from Banner with an external system. For example, during the implementation of the external system, the Web service can be called with no search criteria to provide information about all existing Banner terms.

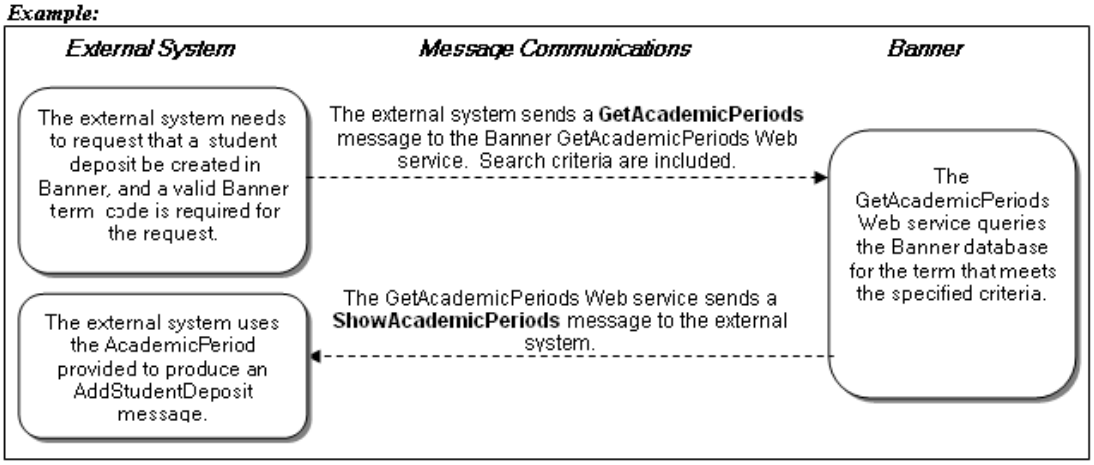
Example:



Search for academic periods

The GetAcademicPeriods Web service can be used to look up a specific term or terms in Banner when the external system needs that term to call another Banner Web service. The search criteria can be used to limit the number of academic periods returned.

Example:



Setup requirements

The GetAcademicPeriods Web service does not require rules on the Crosswalk Validation Form (GTVSDAX) or roles on the Business Rules Form (GORRSQL).This Web service requires that terms are set up on the Term Control Form (SOATERM).

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

GetAcademicPeriods

There are no dynamic translations for the GetAcademicPeriods message.

The following static translation is translated automatically and should not be changed:

Element	Enterprise Value	Transformed to Banner Value
RegistrationAllowed	*Yes*	*Y*

ShowAcademicPeriods

There are no dynamic translations for the ShowAcademicPeriods message.

The following static translations are translated automatically and should not be changed:

Element	Banner Value	Transformed to Enterprise Value
RegistrationAllowed	*Y*	*Yes*
Publishable	*Y*	*Yes*

7 GetEligibleCardholder

The GetEligibleCardholder Web service allows external systems to acquire information from Banner® about an individual who qualifies for a campus card at the institution. External systems can use the provided data to create or update a cardholder record.

This Web service can accept the following user IDs:

- Current Banner ID (Banner UID)
- `GOBSRID_SOURCED_ID` used for Banner Integration for eLearning (IMSID)
- Luminis Platform logon identifier (LogonID)

Message exchange

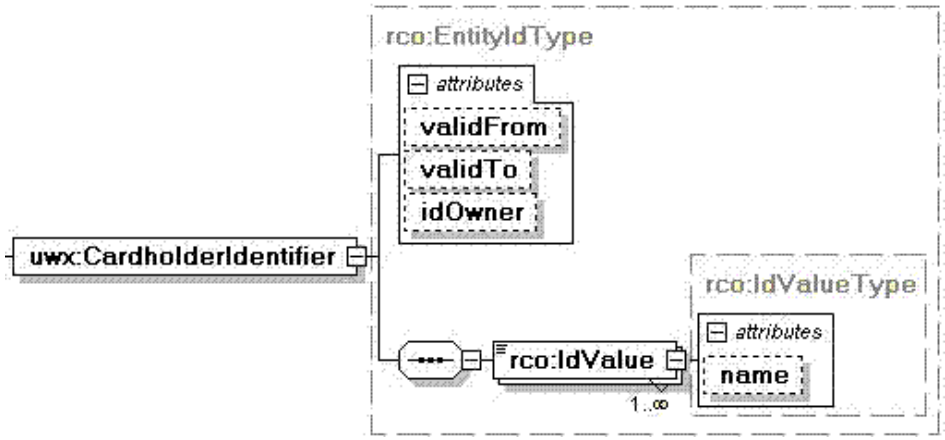
The GetEligibleCardholder Web service exchanges the following messages:

- GetEligibleCardholder
- ShowEligibleCardholder

GetEligibleCardholder

An external system uses the GetEligibleCardholder message to pass an accepted identifier for a person to Banner. The Web service uses the identifier to locate the person in Banner and collect information that makes up the `EligibleCardholder` profile.

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



ShowEligibleCardholder

The ShowEligibleCardholder message is produced in response to a GetEligible Cardholder message. It contains one `EligibleCardholder` element with the following profile information:

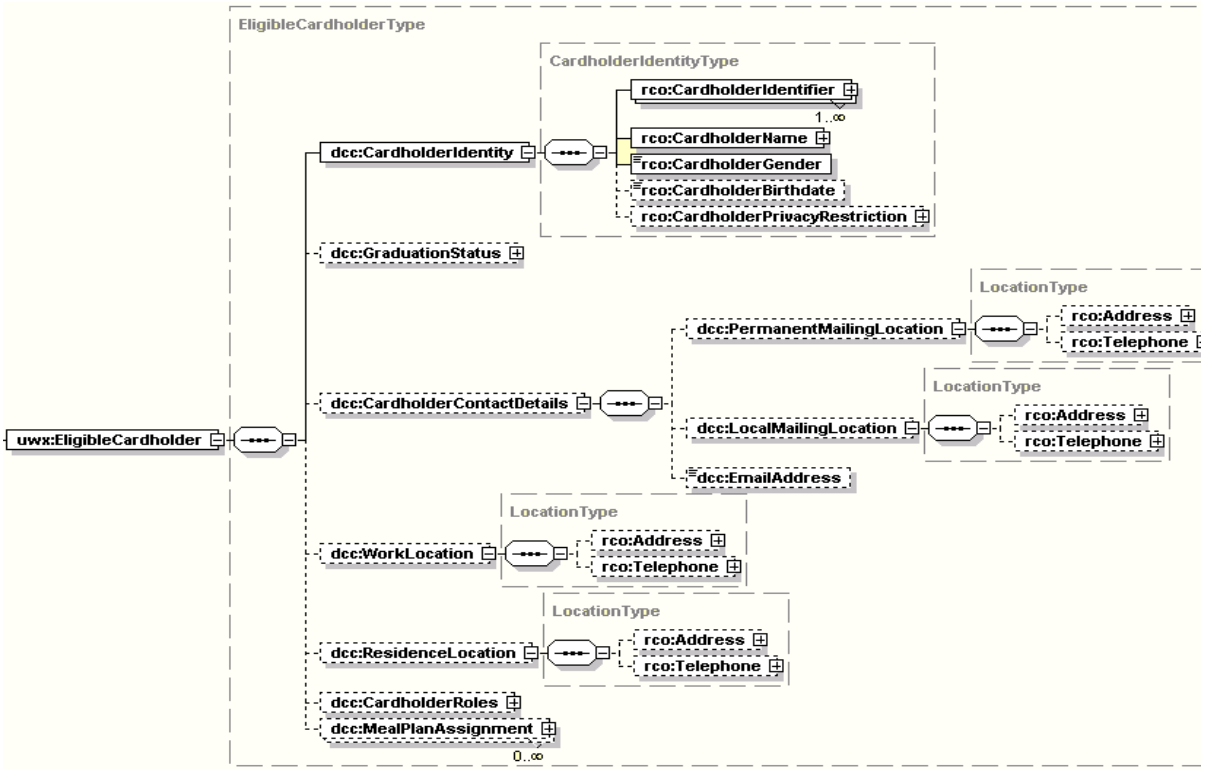
- Name
- Biographical information
- Permanent and local contact information
- Campus residence location
- Campus work location
- Cardholder roles
- Meal assignments

Cardholder roles can be used to determine if a person is eligible for a card and which privileges that card should allow. Cardholder roles are used to categorize potential cardholders or apply a custom set of criteria, based on information stored in Banner, to determine eligibility. For example, if only students who are enrolled in the current term are allowed to own a campus card, a role can be set up on the Business Rules Form (GORRSQL) to support this requirement. A person can qualify for multiple cardholder roles. All roles for which the person qualifies are included in the message.

The GetEligibleCardholder Web service does not require cardholder roles. If either of the following scenarios occurs, the ShowEligibleCardholder message does not include the `CardholderRoles` element:

- There are no active rules for cardholder roles on GORRSQL.
- The requested person does not meet the criteria for any active role.

The following diagram shows the structure of the `EligibleCardholder` element used in the ShowEligibleCardholder message.



SOAP fault messages

If a valid response cannot be created as a ShowEligibleCardholder message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `CardholderIdentifier` provided in the GetEligibleCardholder message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- A current term is not available. A current term is defined as a term for which the current date (SYSDATE) falls between the term start date (`STVTERM_START_DATE`) and end date (`STVTERM_END_DATE`).
- 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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

GetEligibleCardholder

Element/Attribute	Database Mapping
GetEligibleCardholder	
CardholderIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA

ShowEligibleCardholder

Element/Attribute	Database Mapping
ShowEligibleCardholder	
EligibleCardholder	
CardholderIdentity	
CardholderIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	N/A

Element/Attribute	Database Mapping
IdValue	SPRIDEN_ID
@name	NA
CardholderName	
FormattedName	SPBPERS_NAME_PREFIX SPRIDEN_FIRST_NAME SPRIDEN_MI SPRIDEN_LAST_NAME SPBPERS_NAME_SUFFIX
LegalName	SPBPERS_LEGAL_NAME
GivenName	SPRIDEN_FIRST_NAME
PreferredGivenName	SPBPERS_PREF_FIRST_NAME
MiddleName	SPRIDEN_MI
FamilyName	SPRIDEN_LAST_NAME
Affix	SPBPERS_NAME_PREFIX
@type	NA
Affix	SPBPERS_NAME_SUFFIX
@type	NA
CardholderGender	SPBPERS_SEX
CardholderBirthdate	SPBPERS_BIRTH_DATE
CardholderPrivacyRestriction	
PrivacyRestrictionDate	NA
PrivacyRestrictionLevel	SPBPERS_CONFID_IND
GraduationStatus	
ExpectedGraduationDate	SGBSTDN_EXP_GRAD_DATE
ExpectedGraduationPeriod	STVTERM_DESC
@id	SGBSTDN_TERM_CODE_GRAD
@type	NA

Element/Attribute	Database Mapping
ExpectedGraduationYear	SGBSTDN_ACYR_CODE
CardholderContactDetails	
PermanentMailingLocation	
Address	
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
DeliveryAddress	
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3

Element/Attribute	Database Mapping
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
LocalMailingLocation	
Address	
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE

Element/Attribute	Database Mapping
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
DeliveryAddress	
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA

Element/Attribute	Database Mapping
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
EmailAddress	GOREMAL_EMAIL_ADDRESS
WorkLocation	
Address	
@validFrom	NA
@type	NA
@validTo	NA
AddressIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	NA
@name	NA

Element/Attribute	Database Mapping
CountryCode	STVNATN_NATION
PostalCode	PTRJBLN_ZIPC_CODE
Region	PTRJBLN_STAT_CODE
Municipality	PTRJBLN_CITY
DeliveryAddress	
AddressLine	PTRJBLN_DESC
AddressLine	PTRJBLN_ADDRESS1
AddressLine	PTRJBLN_ADDRESS2
StreetName	NA
BuildingNumber	NA
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA

Element/Attribute	Database Mapping
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
ResidenceLocation	
Address	
@validFrom	SPRADDR_FROM_DATE
@validFrom	SLRRASG_BEGIN_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
@validTo	SLRRASG_END_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP or SLBBLDG_ZIP
Region	SPRADDR_STAT_CODE or SLBBLDG_STAT_CODE
Municipality	SPRADDR_CITY or SLBBLDG_CITY
DeliveryAddress	

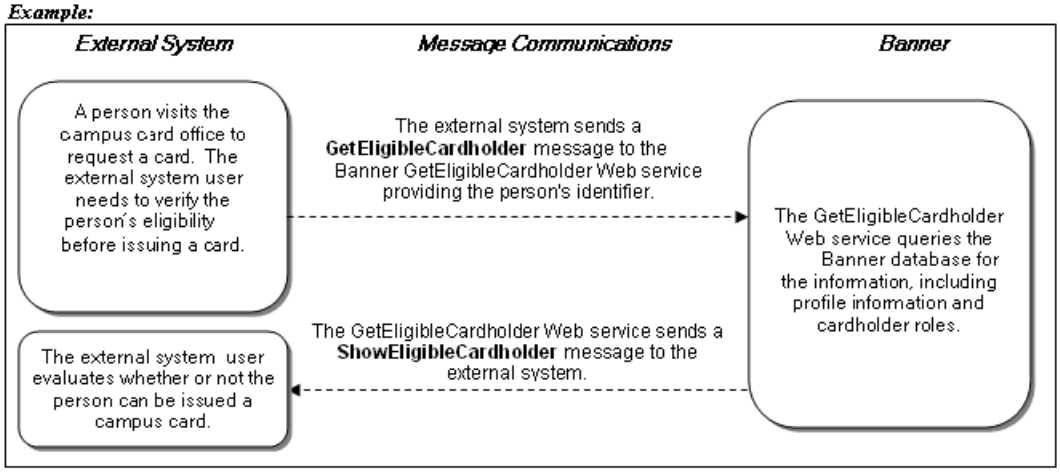
Element/Attribute	Database Mapping
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA or SLRPASG_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER or SLRPASG_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT or SLRPASG_PHONE_EXT

Element/Attribute	Database Mapping
CardholderRoles	
Role	GORRSQL_SQRU_CODE
MealPlanAssignment	
@status	STVMSCD_DESC
MealPlanDescription	STVMRCD_DESC
@id	SLRMASG_MRCD_CODE
BeginDate	SLRMASG_BEGIN_DATE
EndDate	SLRMASG_END_DATE

Intended usage

External systems can use the GetEligibleCardholder Web service to ensure that a person is eligible to receive a campus card, and, if so, retrieve data from Banner required to create a cardholder record in the external system.

Example:



Setup requirements

Initial roles and rules provided with this Web service should be installed in Banner with the other components. You can modify these roles and rules on the Business Rules Form (GORRSQL) and Crosswalk Validation Form (GTVSDAX). After completing the setup

activities, your technical staff should run the `campuscard_check.sql` utility script to verify that the rules exist and that no unused GTVSDAX records remain. If the utility finds GTVSDAX records with an **External Code** set to *UPDATE ME*, you should remove or update these records before using this Web service.

GORRSQL setup

The GetEligibleCardholder Web service provides cardholder roles as part of the information that can be used to determine if a person is eligible for a card and which privileges that card should allow. Cardholder roles are used to categorize potential cardholders or apply a custom set of criteria, based on information stored in Banner, to determine eligibility. For example, a rule can be set up on GORRSQL to support the requirement that only students who are enrolled in the current term are allowed to own a campus card.

Each GORRSQL rule for the GetEligibleCardholder Web service must use process code *CARDHOLDER_ROLES* and a unique rule code that identifies the rule. Sample roles are provided. To meet your institution's needs, you can inactivate the sample roles and add new ones. The role code value (`GORRSQL_SQRU_CODE`**)** is provided as the `Role` element in the ShowEligibleCardholder message, so the value should be descriptive enough to be understood by an external system user.

Note

GORRSQL is used to define rules for various Banner processes. The GetEligibleCardholder Web service uses only those rules that have process code *CARDHOLDER_ROLES*.

The GetEligibleCardholder Web service does not require cardholder roles. If there are no active rules on GORRSQL for cardholder roles, the ShowEligibleCardholder message does not include the `CardholderRoles` element.

Use the following steps to establish a new role for eligible cardholders:

1. Create a rule code on the Business Rule Code Validation Form (GTVSQRU).
2. Access the Business Rules Form (GORRSQL).
3. Enter *CARDHOLDER_ROLES* as the **Process** in the key block.
4. Enter the new rule code in the **Rule** field.
5. In the Rule Data block, enter an SQL SELECT statement that will select the population of persons from Banner who should be given the role. Consider the following when entering your statement:
    - The SELECT clause must select a PIDM, and only a PIDM, from any Banner table that contains a PIDM column.

- The PIDMs you select must be unique. A SELECT statement that selects the same PIDM more than once might create an error when the Web service is called.
- If you want to exclude deceased persons from the eligible cardholder populations, enter a SELECT statement that excludes them.
- The FROM clause can contain more than one table (that is, a join), but the SELECT clause can select a PIDM column from only one table.
- A `:TERM` parameter is available for use in the SELECT statement. When used with the *CARDHOLDER_ROLES* process code, the current term is defined as the highest term code in the set of terms having a start date less than or equal to today's date and an end date greater than or equal to today's date.

This is an example SELECT statement:

```
SELECT SGBSTDN_PIDM
FROM SGBSTDN A,STVSTST
WHERE A.SGBSTDN_STST_CODE = STVSTST_CODE
   AND STVSTST_REQ_IND = 'Y'
   AND A.SGBSTDN_TERM_CODE_EFF =
      (SELECT MAX (B.SGBSTDN_TERM_CODE_EFF)
      FROM SGBSTDN B
      WHERE B.SGBSTDN_PIDM = A.SGBSTDN_PIDM
         AND B.SGBSTDN_TERM_CODE_EFF <= :TERM)
```

6. Use the **Validate** button to validate the SQL statement.

7. Select the **Active** check box to activate the rule.

Once the rule is validated and activated, the GetEligibleCardholder Web service automatically uses the rule when determining a cardholder's roles.

You can modify the rules defined on GORRSQL at any time to change the definition of cardholder roles; however, the roles assigned to cardholders already known to the external campus card system are not re-evaluated until the campus card system requests such information.

Rules that are delivered as seed data cannot be deleted nor modified, but they can be inactivated. If a delivered rule is not needed for a certain role, clear the **Active** check box on the rule and insert a new rule record with a higher sequence number. If more than one active SELECT statement for a given process/rule (each with a unique sequence number) exists, a person in Banner need only be selected by one SELECT statement to be given that role.

Refer to the *Banner General User Guide* for more information on establishing roles on GORRSQL.

GTVSDAX setup

Settings on the Crosswalk Validation Form (GTVSDAX) specify what data is used for cardholder information:

- Contact details
- Residence location
- Work location
- Meal plan assignment status

Contact details

Staff who manage the campus card system use contact details to contact the cardholder. The following GTVSDAX settings specify what contact information is provided in the response message:

- E-mail address
- Permanent location and address type
- Permanent location and telephone type
- Local location and address type
- Local location and telephone type

Note

These settings are not required. If a setting does not exist, the related information is not included in the message.

E-mail address

This GTVSDAX setting determines the e-mail address type codes that the GetEligibleCardholder Web service uses to select the cardholder's e-mail address.

There is no limit to the number of e-mail address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an e-mail address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching e-mail address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_EMAIL*
Internal Group:	*EMAIL*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`GOREMAL_EMAL_CODE` value for the e-mail type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Permanent location and address type

This GTVSDAX setting specifies the address type codes that the GetEligibleCardholder Web service uses to select the cardholder's permanent mailing address.

There is no limit to the number of address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_PERM*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRADDR_ATYP_CODE` value for the address type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Permanent location and telephone type

This GTVSDAX setting specifies the telephone type codes that the GetEligibleCardholder Web service uses to select the cardholder's permanent telephone number.

There is no limit to the number of telephone type codes that can be included. The Web service uses GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the cardholder.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If no matching telephone number is found, the telephone number linked to the permanent address specified in the previous GTVSDAX rule is selected for the cardholder.
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the permanent address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_PERM*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Local location and address type

This GTVSDAX setting specifies the address type codes that the GetEligibleCardholder Web service uses to select the cardholder's local mailing address.

There is no limit to the number of address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_LOCAL*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRADDR_ATYP_CODE` value for the address type code.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Local location and telephone type

This GTVSDAX setting specifies the telephone type codes that the GetEligibleCardholder Web service uses to select the cardholder's local telephone number.

There is no limit to the number of telephone type codes that can be included. The Web service uses GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the cardholder.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If no matching telephone number is found, the telephone number linked to the local address specified in the previous GTVSDAX rule is selected for the cardholder.
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the local address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_LOCAL*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Residence location

The residence location information provides the external campus card system with a primary physical campus residence location for the cardholder and a way to contact the cardholder at that location. The following GTVSDAX settings specify the appropriate residence location information to select for the cardholder and the source of this information in Banner:

- Residence location address source
- Residence address type
- Room assignment status
- Residence location telephone source
- Residence telephone type
- Telephone assignment status

Note

These settings are not required. If a setting does not exist, the related information is not included in the message.

Residence location address source

Campus residence location addresses come from one of two places in Banner:

- Address Table (SPRADDR)
- Room Assignment Table (SLRRASG)

This GTVSDAX setting determines which source the GetEligibleCardholder Web service uses to get the residence location. Use the following information to create or modify a rule on GTVSDAX:

Internal Code:	*CC_RESADDR*
Internal Group:	*DATASOURCE*
External Code:	*P* = Address Table (SPRADDR) *L* = Room Assignment Table (SLRRASG)
Code Description:	Description of this rule
Limit on GTVSDAX:	One
Message Limit:	NA

Residence address type

If the residence location address source setting on GTVSDAX is *P* (SPRADDR), this GTVSDAX setting specifies the address type code that the GetEligibleCardholder Web service uses to select the cardholder's campus residence address.

There is no limit to the number of address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_RESIDE*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRADDR_ATYP_CODE` value for the address type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Room assignment status

If the residence location address source setting on GTVSDAX is *L* (SLRRASG), this GTVSDAX setting specifies the room assignment status codes that the GetEligibleCardholder Web service uses to select active room assignments in Banner Location Management.

Multiple GTVSDAX rules are allowed, but only one active room assignment is selected for each cardholder. The Web service uses the current date (SYSDATE) to identify current active room assignments. If more than one room assignment is active for that date, the assignment with the lowest assignment end date is selected.

Only SLRRASG records with `SLRRASG_ASCD_CODE` are used for the GetEligibleCardholder Web service.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*ACTIVEROOM*
Internal Group:	*ASSIGNMENT STATUS*
External Code:	`SLRRASG_ASCD_CODE` value for the room assignment status code that indicates an active room assignment
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Residence location telephone source

Campus residence location telephone numbers come from one of two places in Banner:

- Telephone Table (SPRTELE)
- Phone Assignment Table (SLRPASG)

This GTVSDAX setting determines which source the GetEligibleCardholder Web service uses to get the residence location telephone number. Use the following information to create or modify a rule on GTVSDAX:

Internal Code:	*CC_RESPHON*
Internal Group:	*DATASOURCE*
External Code:	*P* = Telephone Table (SPRTELE)
	L = Phone Assignments Table (SLRPASG)
Code Description:	Description of this rule
Limit on GTVSDAX:	One
Message Limit:	NA

Residence telephone type

If the residence location telephone source setting on GTVSDAX is *P* (SPRTELE), this GTVSDAX setting specifies the telephone type codes that the GetEligibleCardholder Web service uses to select the cardholder's campus residence telephone number.

There is no limit to the number of telephone type codes that can be included. The Web service uses GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the cardholder.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If no matching telephone number is found and if the residence address source code rule on GTVSDAX is *P*, the telephone number linked to the selected residence address is selected for the cardholder.
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the selected residence address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_RESIDE*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Telephone assignment status

If the residence location telephone source setting on GTVSDAX is *L* (SLRPASG), this GTVSDAX setting specifies which phone assignment status codes the GetEligibleCardholder Web service uses to select active telephone assignments in Banner Location Management.

Multiple GTVSDAX rules are allowed, but only one residence location telephone number is selected for each cardholder. The Web service uses the current date (SYSDATE) to identify current active telephone assignments. If more than one telephone assignment is active for that date, the assignment with the lowest assignment end date is selected.

Only SLRPASG records with `SLRPASG_PSCD_CODE` are used for the GetEligibleCardholder Web service.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*ACTIVEPHON*
Internal Group:	*ASSIGNMENTSTATUS*
External Code:	`SLRPASG_PSCD_CODE` value for the phone assignment status code that indicates an active phone assignment.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Work location

Work location information provides the external campus card system with a primary physical work location for the cardholder and a way to contact the cardholder during work hours.

The physical work location does not require a GTVSDAX setting because it is derived from the job location (`NBRJOBS_JBLN_CODE`) that is associated with the person's primary job for the current date. Details about the address are retrieved from the Job Location Rules Form (PTRJBLN). To include work location details in the ShowEligibleCardholder messages, you should define job locations on PTRJBLN and ensure that the **Job Location** field on the Position Definition Form (NBAPOSN) or Employee Jobs Form (NBAJOBS) references a job location defined on PTRJBLN.

A GTVSDAX setting specifies the telephone type codes that the GetEligibleCardholder Web service uses to select the cardholder's work telephone number. There is no limit to the number of telephone type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an active telephone number matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching telephone number is found or no GTVSDAX rules are left to evaluate. It does not select an unlisted telephone number unless no active, listed numbers match any of the rules on GTVSDAX. If an unlisted telephone number is selected, the term *Unlisted* is provided in the `SubscriberNumber` element.

Note

This GTVSDAX setting is not required. If the setting does not exist, the work telephone number is not included in the message. ■

Use the following information to create or modify the rules on GTVSDAX:

Internal Code:	*CC_WORK*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Meal plan assignment status

This GTVSDAX setting specifies the meal plan assignment status codes that the GetEligibleCardholder Web service uses to identify active meal plan assignments for cardholders.

Note

This setting is not required. If a setting does not exist, the related information is not included in the message.

Multiple rules are allowed. The Web service selects all meal plan assignments with a status code found in any of the GTVSDAX meal plan assignment status rules, if the current date (SYSDATE) falls between the assignment start and end dates.

Only SLRMASG records with this `SLRMASG_MSCD_CODE` are used by the GetEligible Cardholder Web service.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*ACTIVEMEAL*
Internal Group:	*ASSIGNMENTSTATUS*
External Code:	`SLRMASG_MSCD_CODE` value for the meal plan assignment status code that indicates an active meal plan assignment.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

GetEligibleCardholder

There are no dynamic or static translations for the GetEligibleCardholder message.

ShowEligibleCardholder

The following dynamic translation must be customized:

Element	Banner Value	Transformed to Enterprise Value
`CardholderGender`	*M*	*Male*
	F	*Female*
	N	*Unknown*

There are no static translations for the ShowEligibleCardholder message.

8 GetHousingApplicant Eligibility

The GetHousingApplicantEligibility Web service allows external systems to request housing applicant information to verify eligibility for campus housing or assignment to a specific room.

A *housing applicant* is an individual who is interested in residential accommodations at the institution. Persons currently in the Banner® system are identified as potential housing applicants by using one or more housing eligibility roles established on the Business Rules Form (GORRSQL). Refer to "Setup requirements" on page 8-9 for more information.

Note

This Web service requires Banner Student.

Message exchange

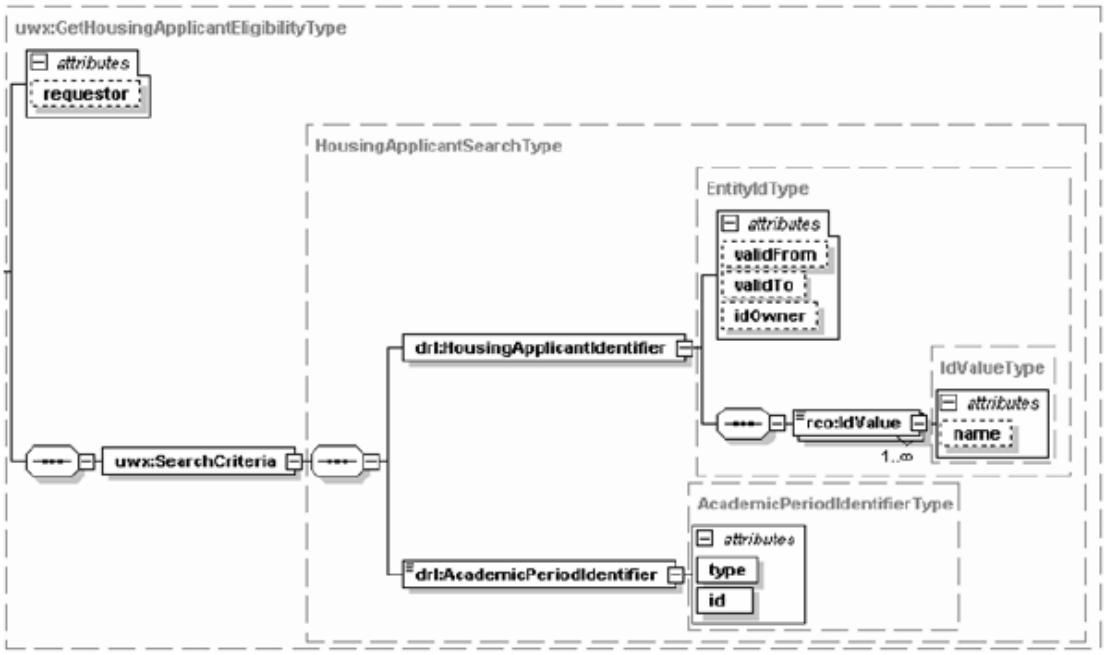
The GetHousingApplicantEligibility Web service exchanges the following messages:

- GetHousingApplicantEligibility
- ShowHousingApplicantEligibility

GetHousingApplicantEligibility

An external system uses the GetHousingApplicantEligibility message to request housing applicant information. The message must include one valid identifier (current Banner ID) and the term for which information is needed.

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



ShowHousingApplicantEligibility

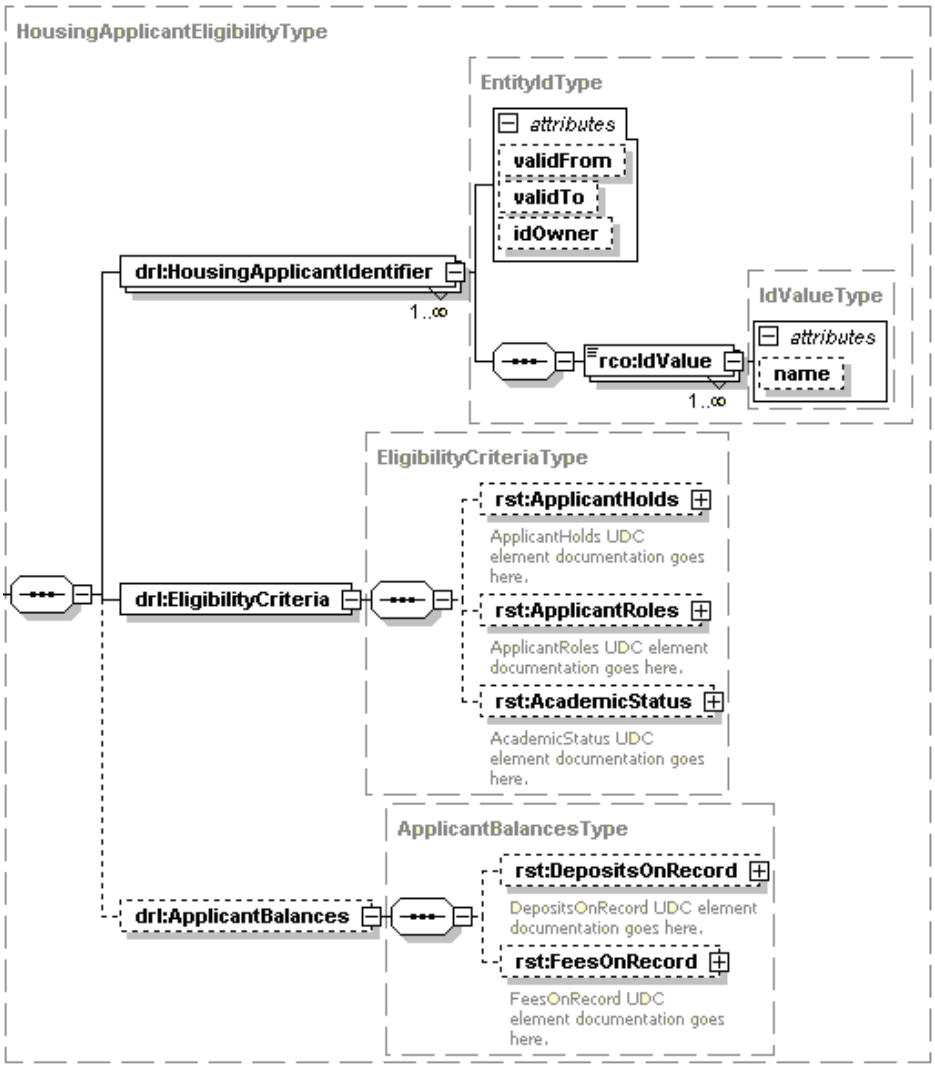
The GetHousingApplicantEligibility Web service returns the ShowHousing ApplicantEligibility message. This message contains information about a person's holds, academic standing, housing eligibility roles, and fees and deposits related to residential housing charges. GTVSDAX rules establish the detail codes and deposit type codes that should be included.

Applicant roles can be used to determine if a person is eligible for housing. Roles are used to categorize applicants or apply a custom set of criteria, based on information stored in Banner, to determine eligibility. For example, if only students who are enrolled in the current term are allowed to apply for housing, a role can be set up on the Business Rules Form (GORRSQL) to support this requirement. A person can qualify for multiple roles. All roles for which the person qualifies are included in the message.

The GetHousingApplicantEligibility Web service does not require roles. If either of the following scenarios occurs, the ShowHousingApplicantEligibility message does not include the `ApplicantRoles` element:

- There are no active rules for applicant roles on GORRSQL.
- The requested person does not meet the criteria for any active role.

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



SOAP fault messages

If a valid response cannot be created as a ShowHousingApplicantEligibility message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `HousingApplicantIdentifier` provided in the GetHousingApplicant Eligibility message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- A network, database, or other technical issue occurs.
- The applicable GTVSDAX rows for this Banner Web service have the value *UPDATE ME* in the **External Code** field.

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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

GetHousingApplicantEligibility

Element/Attribute	Database Mapping
GetHousingApplicantEligibility	
@requestor	
SearchCriteria	
HousingApplicantIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
AcademicPeriodIdentifier	STVTERM_CODE
@id	STVTERM_DESC
@type	NA

ShowHousingApplicantEligibility

Element/Attribute	Database Mapping
ShowHousingApplicantEligibility	
HousingApplicantEligibility	
HousingApplicantIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
EligibilityCriteria	
ApplicantHolds	
ApplicantHold	
HoldType	STVHLDD_DESC
@codeValue	SPRHOLD_HLDD_CODE
FromDate	SPRHOLD_FROM_DATE
ToDate	SPRHOLD_TO_DATE
Restriction	STVHLDD_REG_HOLD_IND
Reason	SPRHOLD_REASON
ApplicantRoles	
Role	GORRSQL_SQRU_CODE
AcademicStatus	

Element/Attribute	Database Mapping
AcademicStanding	
AcademicPeriod	STVTERM_DESC
@id	SHRTTRM_TERM_CODE
@id	SGBSTDN_TERM_CODE_EFF
@type	NA
Assessment	STVASTD_DESC
ProgressEvaluation	
AcademicPeriod	STVTERM_DESC
@id	SHRTTRM_TERM_CODE
@id	SGBSTDN_TERM_CODE_EFF
@type	NA
Assessment	STVPREV_DESC
CombinedAcademicStanding	
AcademicPeriod	STVTERM_DESC
@id	SHRTTRM_TERM_CODE
@id	SGBSTDN_TERM_CODE_EFF
@type	NA
Assessment	STVCAST_DESC
ApplicantBalances	
DepositsOnRecord	
Deposit	
AcademicPeriodIdentifier	STVTERM_DESC
@id	TBRDEPO_TERM_CODE
@type	NA
DepositType	TTVDTYP_DESC
@codeValue	TBBDEPC_DTYP_CODE

Element/Attribute	Database Mapping
InitialDeposit-Amount	Sum(TBRDEPO_AMOUNT)
@currency	NA
DepositBalance	TBRDEPO_AMOUNT
@currency	NA
FeesOnRecord	
Fee	
AcademicPeriodIdentifier	STVTERM_DESC
@id	TBRACCD_TERM_CODE
@type	NA
TransactionType	TBBDETC_DESC
@category	TTVDCAT_DESC
@id	TBRACCD_DETAIL_CODE
@classification	TBRACCD_TYPE_IND
NetAmountPosted	TBRACCD_AMOUNT
@currency	NA

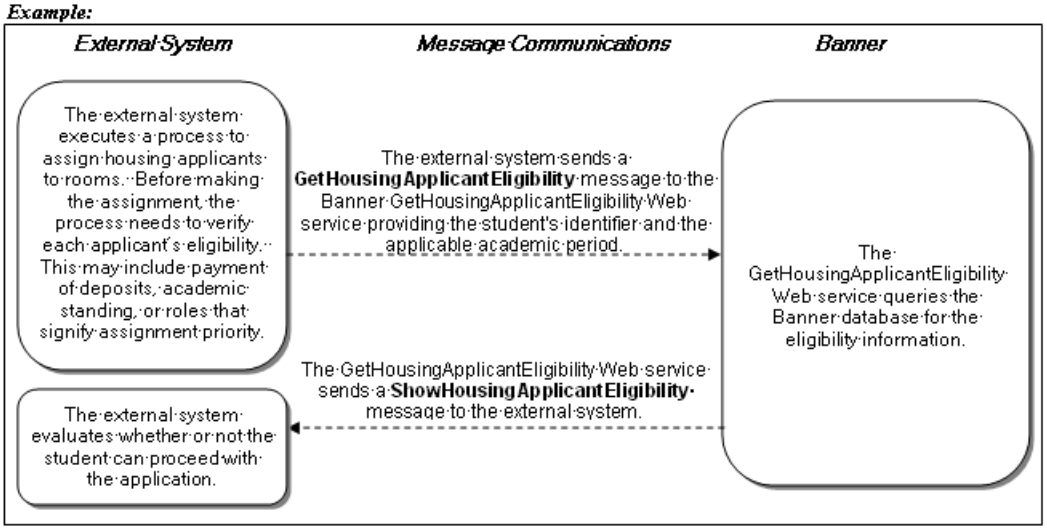
Intended usage

External systems can use the GetHousingApplicantEligibility Web service to ensure that a person is eligible for housing.

Online housing application

External systems can use the GetHousingApplicantEligibility Web service to obtain data to verify eligibility when a housing application is attempted.

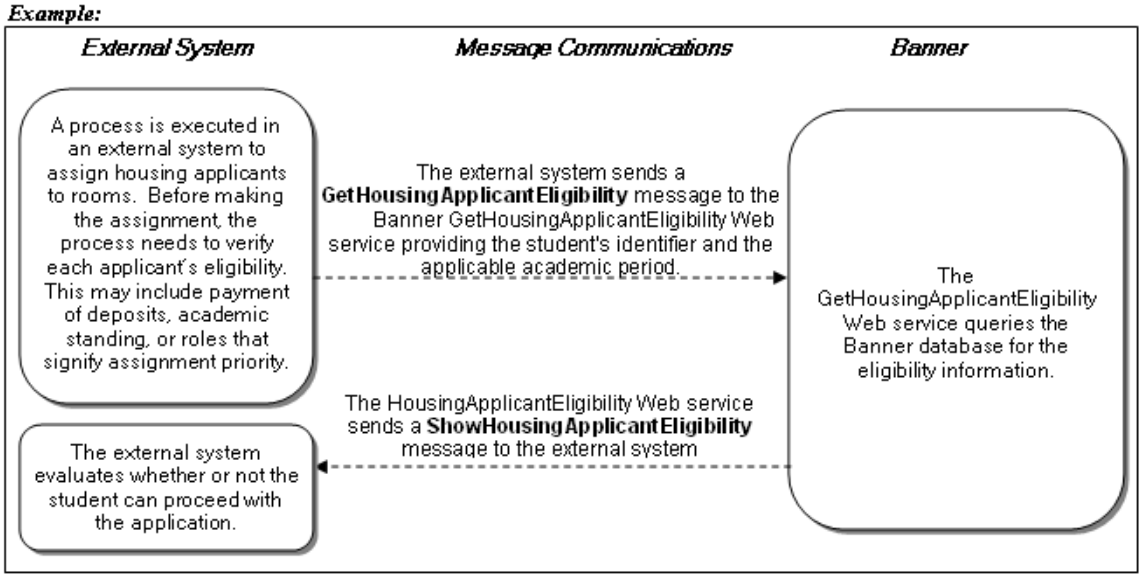
Example:



Room assignment process

External systems can use the GetHousingApplicantEligibility Web service to obtain data that can be used to verify eligibility during the room assignment process.

Example:



Setup requirements

Initial roles and rules provided with this Web service should be installed in Banner with the other components. You can modify these roles and rules on the Business Rules Form (GORRSQL) and Crosswalk Validation Form (GTVSDAX). After completing the setup activities, your technical staff should run the `housing_check.sql` utility script to verify that the rules exist and that no unused GTVSDAX records remain. If the utility finds GTVSDAX records with an **External Code** set to *UPDATE ME*, you should remove or update these records before using this Web service.

GORRSQL setup

The GetHousingApplicantEligibility Web service provides applicant roles as part of the information that can be used to determine the eligibility of the housing applicant. Applicant roles are used to categorize applicants or apply a custom set of criteria, based on information stored in Banner, to determine eligibility. For example, a rule can be set up on GORRSQL to support the requirement that only students who are enrolled in the current term are allowed to apply for housing.

Each GORRSQL rule for the GetHousingApplicantEligibility Web service must use process code *HOUSING_ELIGIBILITY* and a unique rule code that identifies the rule. A sample role is provided. To meet your institution's needs, you can inactivate the sample role and add a new one. The role code value (`GORRSQL_SQRU_CODE`) is provided as the `Role` element in the ShowHousing ApplicantEligibility message, so the value should be descriptive enough to be understood by an external system user.

Note

GORRSQL is used to define rules for various Banner processes. The GetHousingApplicantEligibility Web service uses only those rules that have process code *HOUSING_ELIGIBILITY*.

The GetHousingApplicantEligibility Web service does not require applicant roles. If there are no active rules on GORRSQL for applicant roles, the ShowHousingApplicant Eligibility message does not include the `ApplicantRoles` element.

Use the following steps to establish a new role for housing applicants:

1. Create a rule code on the Business Rule Code Validation Form (GTVSQRU).
2. Access the Business Rules Form (GORRSQL).
3. Enter *HOUSING_ELIGIBILITY* as the **Process** in the key block.
4. Enter the new rule code in the **Rule** field.

5. In the Rule Data block, enter an SQL SELECT statement that will select the population of persons from Banner who should be given the role. Consider the following when entering your statement:

   - The SELECT clause must select a PIDM, and only a PIDM, from any Banner table that contains a PIDM column.
   - The PIDMs you select must be unique. A SELECT statement that selects the same PIDM more than once might create an error when the Web service is called.
   - If you want to exclude deceased persons from the eligible applicant populations, enter a SELECT statement that excludes them.
   - The FROM clause can contain more than one table (that is, a join), but the SELECT clause can select a PIDM column from only one table.
   - The term code (AcademicPeriod) provided in the GetHousingApplicant Eligibility message can be included using the `:TERM` parameter. TERM must also be a valid value on the Business Rule Parameter Code Validation Form (GTVSQPA).

   This is an example SELECT statement:

```
SELECT SFBETRM_PIDM
FROM STVESTS, SFBETRM
WHERE SFBETRM_TERM_CODE = :TERM
   AND STVESTS_CODE = SFBETRM_ESTS_CODE
   AND STVESTS_WD_IND = 'N'
```

6. Use the **Validate** button to validate the SQL statement.

7. Select the **Active** checkbox to activate the rule.

Once the rule is validated and activated, the GetHousingApplicantEligibility Web service automatically uses the rule when determining a housing applicant's roles.

You can modify the rules defined on GORRSQL at any time to change the definition of housing applicant roles; however, the roles assigned to housing applicants already known to the external housing system are not re-evaluated until the housing system requests such information.

Rules that are delivered as seed data cannot be deleted nor modified, but they can be inactivated. If a delivered rule is not needed for a certain role, clear the **Active** check box on the rule and insert a new rule record with a higher sequence number. If more than one active SELECT statement for a given process/rule (each with a unique sequence number) exists, a person in Banner need only be selected by one SELECT statements to be given that role.

Refer to the *Banner General User Guide* for more information on establishing roles on GORRSQL.

GTVSDAX setup

Settings on the Crosswalk Validation Form (GTVSDAX) control the types of deposit and fee summary information included in the ShowHousingApplicantEligibility message.

Note

These settings are not required. If a setting does not exist, the summarization group is not included in the message. Any combination of summarization groups can be used in the messages.

Deposit information

Three kinds of summarization groups are available for deposit information:

- All-term
- Show-term
- Specified-term

Deposits are shown on the Student Account Detail Form (TSADETL). Deposit type codes are established on the Deposit Type Code Validation Form (TTVDTYP) and linked to deposit detail codes on the Deposit Detail Control Form (TGADEPC). Detail codes are established on the Detail Code Control Form (TSADETC).

All-term deposit summarization

The all-term deposit summarization groups deposit information for the student by deposit type code. Term codes are not included in the information.

This GTVSDAX setting specifies which deposit type codes are used to select the deposit records that are summarized for the GetHousingApplicantEligibility Web service. There is no limit to the number of deposit type codes that can be included. All deposits associated with the deposit type codes included in the GTVSDAX setting are summarized in the ShowHousingApplicantEligibility message.

This rule is not required, but if present, the all-term deposit summarization exists in the ShowHousingApplicantEligibility message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*DEPOSITALLTERM*
Translation Code:	NA
External Code:	Deposit type codes from `TTVDTYP_CODE` required for all-term deposit summarizations
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Show-term deposit summarization

The show-term deposit summarization groups deposit information for the student by deposit type code and term code. Term codes are included in the information.

This GTVSDAX setting specifies which deposit type codes are used to select the deposit records that are summarized for the GetHousingApplicantEligibility Web service. There is no limit to the number of deposit type codes that can be included. All deposits associated with the deposit type codes included in the GTVSDAX setting are summarized in the ShowHousingApplicantEligibility message.

This rule is not required, but if present, the show-term deposit summarization exists in the ShowHousingApplicantEligibility message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*DEPOSITSHOWTERM*
Translation Code:	NA
External Code:	Deposit type codes from `TTVDTYP_CODE` required for show-term deposit summarizations
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Specified-term deposit summarization

The specified-term deposit summarization groups deposit information for the student by deposit type codes for the term specified in the GetHousingApplicantEligibility message.

This GTVSDAX setting specifies which deposit type codes are used to select the deposit records that are summarized for the GetHousingApplicantEligibility Web service. There is no limit to the number of deposit type codes that can be included. All deposits associated with the deposit type codes included in the GTVSDAX setting and the specified term are summarized in the ShowHousingApplicantEligibility message.

This rule is not required, but if present, the specified-term deposit summarization exists in the ShowHousingApplicantEligibility message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*DEPOSITSPECIFICTERM*
Translation Code:	NA
External Code:	Deposit type codes from `TTVDTYP_CODE` required for specified-term deposit summarizations
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Fee information

Three kinds of summarization groups are available for fee information:

- All-term
- Show-term
- Specified-term

Fees represent charges or payments shown on the Student Account Detail Form (TSADETL). Detail codes are established on the Detail Code Control Form (TSADETC).

All-term fee summarization

The all-term fee summarization groups fee information for the student by detail code. No term codes are included in the information.

This GTVSDAX setting specifies which detail codes are used to select the transaction records shown on TSADETL (TBRACCD) for housing-related fees and summarized for the GetHousingApplicantEligibility Web service. There is no limit to the number of detail codes that can be included. All transactions associated with detail codes included in the GTVSDAX setting are summarized in the ShowHousingApplicantEligibility message.

This rule is not required, but if present, the all-term fee summarization exists in the ShowHousingApplicantEligibility message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*FEEALLTERM*
Translation Code:	NA
External Code:	Detail codes from `TBBDETC_DETAIL_CODE` required for all-term fee summarizations
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Show-term fee summarization

The show-term fee summarization groups fee information for the student by detail code and term code. All term codes are included in the information.

This GTVSDAX setting specifies which detail codes are used to select the transaction records shown on TSADETL (TBRACCD) for housing-related fees and summarized for the GetHousingApplicantEligibility Web service. There is no limit to the number of detail codes that can be included. All transactions associated with detail codes included in the GTVSDAX setting are summarized in the ShowHousingApplicantEligibility message.

This rule is not required, but if present, the show-term fee summarization exists in the ShowHousingApplicantEligibility message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*FEESHOWTERM*
Translation Code:	NA
External Code:	Detail codes from `TBBDETC_DETAIL_CODE` required for show-term fee summarizations
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Specified-term fee summarization

The specified-term fee summarization groups fee information for the student by detail codes for the term specified in the GetHousingApplicantEligibility message.

This GTVSDAX setting specifies which detail codes are used to select the transaction records shown on TSADETL (TBRACCD) for housing-related fees and summarized for the GetHousingApplicantEligibility Web service. There is no limit to the number of detail codes that can be included. All transactions associated with the detail codes included in the GTVSDAX setting and the specified term are summarized in the message.

This rule is not required, but if present, the specified-term fee summarization exists in the ShowHousingApplicantEligibility message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*FEESPECIFICTERM*
Translation Code:	NA
External Code:	Detail codes from `TBBDETC_DETAIL_CODE` required for specified-term fee summarizations
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

GetHousingApplicantEligibility

There are no dynamic or static translations for the GetHousingApplicantEligibility message.

ShowHousingApplicantEligibility

There are no dynamic translations for the ShowHousingApplicantEligibility message.

The following static translation is translated automatically and should not be changed:

Element	Enterprise Value	Transformed to Banner Value
`@classification`	*C* *P*	*Charge* *Payment*

9 GetHousingApplicant Profile

The GetHousingApplicantProfile Web service allows external systems to request detailed information about a housing applicant. This information can affect the applicant's room assignment selection and priority. This information is requested after a student's eligibility is verified.

The housing applicant profile includes the following information:

- Name and other identifying information
- Demographic information from the General Person Form (SPAPERS)
- Contact address, telephone number, and e-mail address
- Medical information and disabilities
- Emergency contact information
- General learner information, including curricula and academic standing
- Participation in activities that could affect room assignments

Note

This Web service requires Banner® Student.

Message exchange

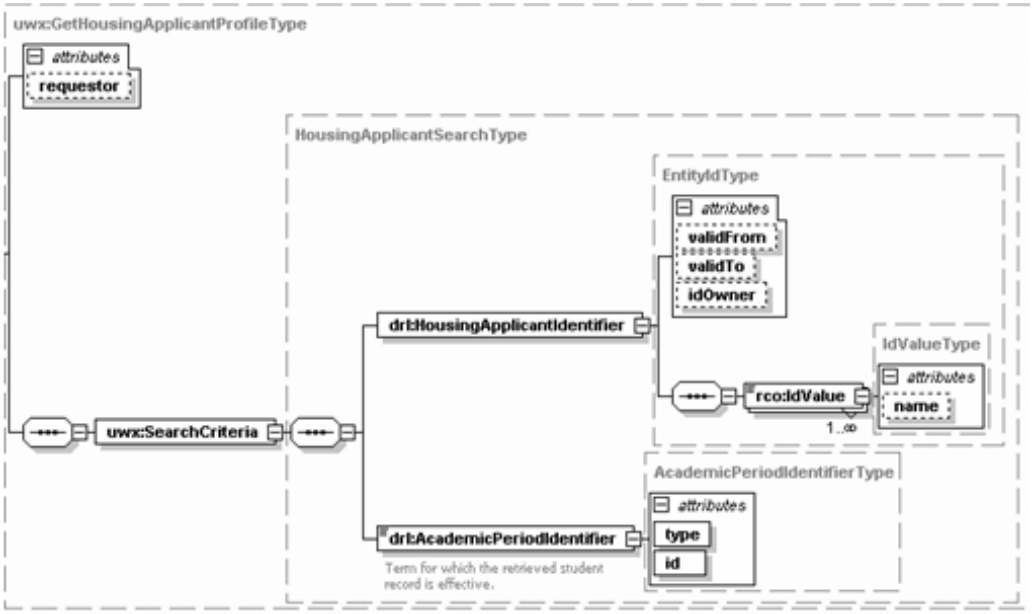
The GetHousingApplicantProfile Web service exchanges the following messages:

- GetHousingApplicantProfile
- ShowHousingApplicantProfile

GetHousingApplicantProfile

An external system uses the GetHousingApplicantProfile message to request detailed information about a housing applicant. The message must include one valid identifier (for example, current Banner ID) and a valid term.

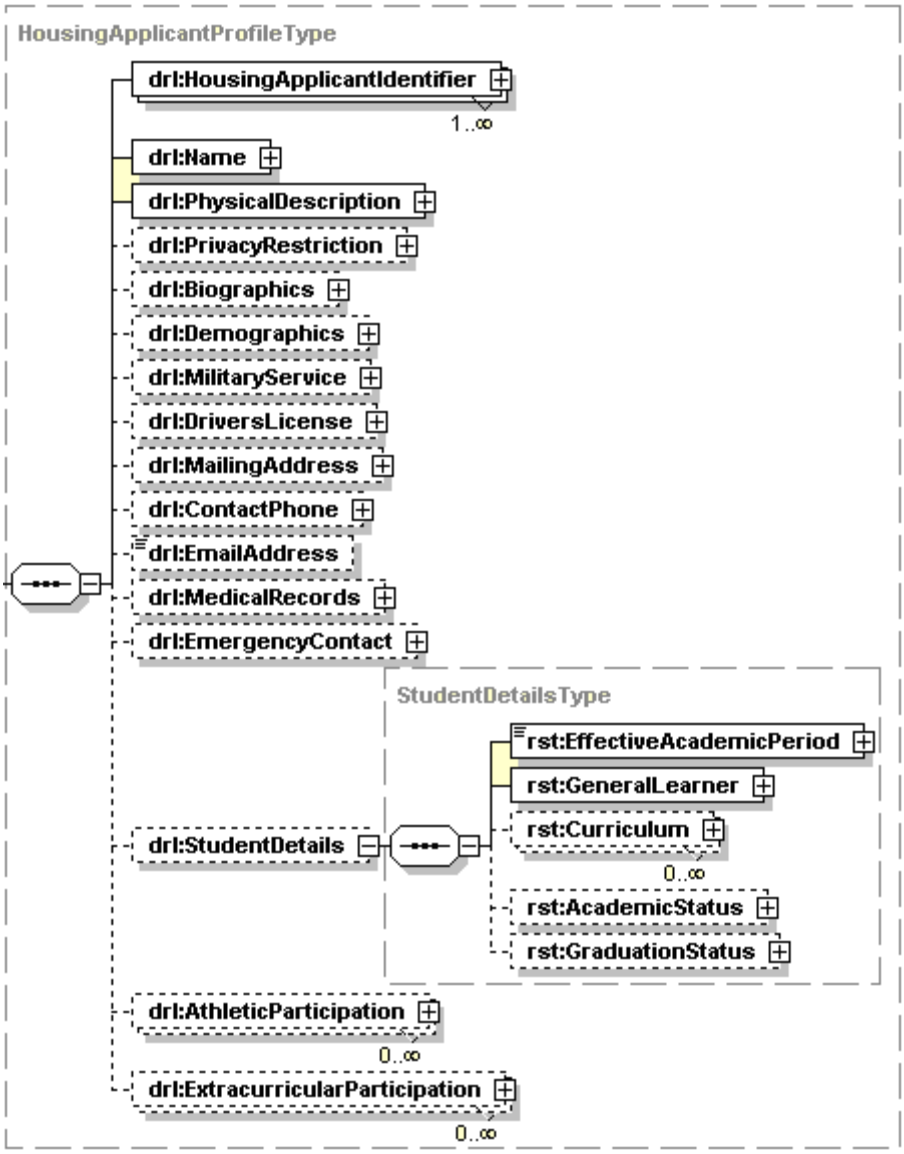
The following diagram shows the structure of the GetHousingApplicantProfile message schema.



ShowHousingApplicantProfile

The GetHousingApplicantProfile Web service returns the ShowHousingApplicantProfile message. This message contains detailed information that the external system can use to make appropriate room assignments or create an applicant record. GTVSDAX rules are needed for the underlying API to include all relevant information in this message.

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



SOAP fault messages

If a valid response cannot be created as a ShowHousingApplicantProfile message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `HousingApplicantIdentifier` provided in the GetHousingApplicantProfile message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- A valid Banner term is not provided in the GetHousingApplicantProfile message.
- A network, database, or other technical issue occurs.

- The applicable GTVSDAX rows for this service have the value *UPDATE ME* in the **External Code** field.

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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

GetHousingApplicantProfile

Element/Attribute	Database Mapping
GetHousingApplicantProfile	
@requestor	NA
SearchCriteria	
HousingApplicantIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
AcademicPeriodIdentifier	STVTERM_DESC

Element/Attribute	Database Mapping
@id	STVTERM_CODE
@type	NA

ShowHousingApplicantProfile

Element/Attribute	Database Mapping
ShowHousingApplicantProfile	
HousingApplicantProfile	
HousingApplicantIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
Name	
FormattedName	SPBPERS_NAME_PREFIX || ' ' || SPRIDEN_FIRST_NAME || ' ' || SPRIDEN_MI || ' '|| SPRIDEN_LAST_NAME || ', ' || SPBPERS_NAME_SUFFIX
LegalName	SPBPERS_LEGAL_NAME
GivenName	SPRIDEN_FIRST_NAME
PreferredGivenName	SPBPERS_PREF_FIRST_NAME
MiddleName	SPRIDEN_MI

Element/Attribute	Database Mapping
FamilyName	SPRIDEN_LAST_NAME
Affix	SPBPERS_NAME_PREFIX
@type	NA
Affix	SPBPERS_NAME_SUFFIX
@type	NA
PhysicalDescription	
Gender	SPBPERS_SEX
Hair	SPBPERS_HAIR_CODE
Eyes	SPBPERS_EYES_CODE
Height	SPBPERS_HEIGHT
@unitOfMeasure	GTVUOMS_DESC
Weight	SPBPERS_WEIGHT
@unitOfMeasure	GTVUOMS_DESC
PrivacyRestriction	
PrivacyRestrictionDate	NA
PrivacyRestrictionLevel	SPBPERS_CONFID_IND
Biographics	
DateOfBirth	SPBPERS_BIRTH_DATE
PlaceOfBirth	SPBPERS_CITY_BIRTH
@placeType	NA
PlaceOfBirth	SPBPERS_STAT_CODE_BIRTH
@placeType	NA
Deceased	SPBPERS_DEAD_IND
DeceasedDate	SPBPERS_DEAD_DATE
Demographics	
Legacy	STVLGCY_DESC

Element/Attribute	Database Mapping
Ethnicity	STVETHN_DESC
MaritalStatus	STVMRTL_DESC
Religion	STVRELG_DESC
CitizenIndicator	STVCITZ_CITIZEN_IND
CitizenType	STVCITZ_DESC
Incarceration	SPBPERS_INCAR_IND
MilitaryService	
FileNumber	SPBPERS_VETC_FILE_NUMBER
ActiveDutySeparationDate	SPBPERS_ACTIVE_DUTY_SEPR_DATE
Category	SPBPERS_VERA_IND
SpecialDisabledVeteran	SPBPERS_SDVET_IND
DriversLicense	
LicenseNumber	SPBPERS_DRIVER_LICENSE
IssuingJurisdiction	SPBPERS_STAT_CODE_DRIVER
@typeOfJurisdiction	N/A
IssuingJurisdiction	STVNATN_NATION
@typeOfJurisdiction	NA
DateOfIssue	NA
ExpirationDate	NA
MailingAddress	
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA

Element/Attribute	Database Mapping
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
DeliveryAddress	
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA

Element/Attribute	Database Mapping
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
ContactPhone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE _PHONE_AREA
SubscriberNumber	SPRTELE _PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
EmailAddress	GOREMAL_EMAIL_ADDRESS
MedicalRecords	
MedicalRecord	
Description	STVMEDI_DESC
Disability	STVDISA_DESC
@primary	SPRMEDI_DISB_IND
SpecialNeeds	STVMDEQ_DESC
ServiceProvider	STVSPSR _DESC
ImmunizationDate	NA
AdditionalInformation	SPRMEDI_COMMENT
EmergencyContact	
Priority	SPREMRG_PRIORITY
Contact	
@validFrom	NA
@type	STVATYP_DESC

Element/Attribute	Database Mapping
@validTo	N/A
AddressIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	NA
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPREMRG _ZIP
Region	SPREMRG _STAT_CODE
Municipality	SPREMRG _CITY
DeliveryAddress	
AddressLine	SPREMRG_ STREET_LINE1
AddressLine	SPREMRG _STREET_LINE2
AddressLine	SPREMRG _STREET_LINE3
StreetName	NA
BuildingNumber	NA
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	SPREMRG_FIRST_NAME
PreferredGivenName	NA
MiddleName	SPREMRG _MI

Element/Attribute	Database Mapping
FamilyName	SPREMRG _LAST_NAME
Affix	NA
@type	NA
ContactPhone	
InternationalCountryCode	NA
NationalNumber	NA
AreaCityCode	SPREMRG _PHONE_AREA
SubscriberNumber	SPREMRG_PHONE_NUMBER
Extension	SPREMRG _PHONE_EXT
Relation	STVRELT_DESC
StudentDetails	
EffectiveAcademicPeriod	STVTERM_DESC
@id	SGBSTDN_TERM_CODE_EFF
@type	NA
GeneralLearner	
Status	STVSTST_DESC
@codeValue	SGBSTDN_STST_CODE
StudentType	STVSTYP_DESC
@codeValue	SGBSTDN_STYP_CODE
ResidencyStatus	STVRESD_DESC
@codeValue	SGBSTDN_RESD_CODE
FullPartTime	SGBSTDN_FULL_PART_IND
@codeValue	NA
Site	STVSITE_DESC
@codeValue	SGBSTDN_SITE_CODE
Session	STVSESS_DESC

Element/Attribute	Database Mapping
@codeValue	SGBSTDN_SESS_CODE
LeaveOfAbsence	
LeaveReason	STVLEAV_DESC
@codeValue	SGBSTDN_LEAV_CODE
LeaveFromDate	SGBSTDN_LEAV_FROM_DATE
LeaveToDate	SGBSTDN_LEAV_TO_DATE
EducationLevel	STVLEVL_DESC
@codeValue	SORLCUR_LEVL_CODE
AdmissionType	STVADMT_DESC
@codeValue	SORLCUR_ADMT_CODE
Curriculum	
@id	SORLCUR_TERM_CODE
PriorityNumber	SORLCUR_PRIORITY_NO
College	STVCOLL_DESC
@codeValue	SORLCUR_COLL_CODE
Degree	STVDEGC_DESC
@codeValue	SORLCUR_DEGC_CODE
Campus	STVCAMP_DESC
@codeValue	SORLCUR_CAMP_CODE
Program	SMRPRLE_PROGRAM_DESC
@codeValue	SORLCUR_PROGRAM
FieldofStudy	
@id	SORLFOS_SEQNO
FieldOfStudyType	GTVLFST_DESC
@codeValue	SORLFOS_LFST_CODE
CurriculumStatus	STVCSTS_DESC

Element/Attribute	Database Mapping
@codeValue	SORLFOS_CSTS_CODE
CurriculumActivityStatus	STVCACT_DESC
@codeValue	SORLFOS_CACT_CODE
Major	STVMAJR_DESC
@codeValue	SORLFOS_MAJR_CODE
Department	STVDEPT_DESC
@codeValue	SORLFOS_DEPT_CODE
MajorAttached	STVMAJR_DESC
@codeValue	SORLFOS_MAJR_CODE_ATTACH
AcademicStatus	
AcademicStanding	
AcademicPeriod	STVTERM_DESC
@id	SHRTTRM_TERM_CODE
@id	SGBSTDN_TERM_CODE_EFF
@type	NA
Assessment	STVASTD_DESC
ProgressEvaluation	
AcademicPeriod	STVTERM_DESC
@id	SHRTTRM_TERM_CODE
@id	SGBSTDN_TERM_CODE_EFF
@type	NA
Assessment	STVPREV_DESC
CombinedAcademicStanding	
AcademicPeriod	STVTERM_DESC
@id	SHRTTRM_TERM_CODE
@id	SGBSTDN_TERM_CODE_EFF

Element/Attribute	Database Mapping
@type	NA
Assessment	STVCAST_DESC
GraduationStatus	
ExpectedGraduationDate	SGBSTDN_EXP_GRAD_DATE
ExpectedGraduationPeriod	STVTERM_DESC
@id	SGBSTDN_TERM_CODE_GRAD
@type	NA
ExpectedGraduationYear	SGBSTDN_ACYR_CODE
AthleticParticipation	
Activity	STVSACT_ACTC_DESC
@codeValue	SGRSPRT_ACTC_CODE
InitialParticipationPeriod	STVTERM_DESC
@id	SGRSPRT_TERM_CODE
@type	NA
EndParticipationPeriod	STVTERM_DESC
@id	SGRSPRT_TERM_CODE
@type	NA
EligibilityReason	STVELIG_DESC
ExtracurricularParticipation	
Activity	STVACTC_DESC
@codeValue	STVSACT_ACTC_CODE
InitialParticipationPeriod	STVTERM_DESC
@id	SGRSACT_TERM_CODE
@type	NA
EndParticipationPeriod	STVTERM_DESC
@id	SGRSACT_TERM_CODE

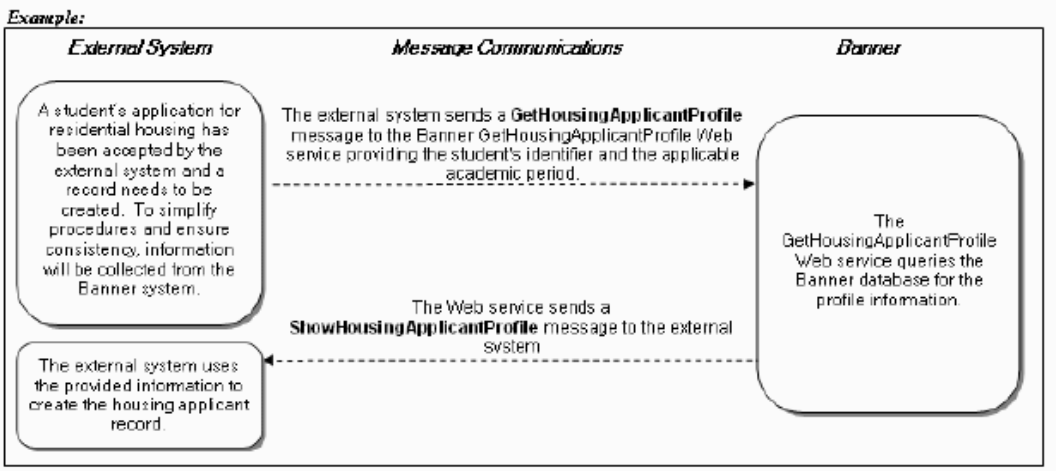
Element/Attribute	Database Mapping
@type	NA
EligibilityReason	NA

Intended usage

External systems can use the GetHousingApplicantProfile Web service to request detailed information about a housing applicant. This information can be used to create a housing applicant record or make appropriate room assignments.

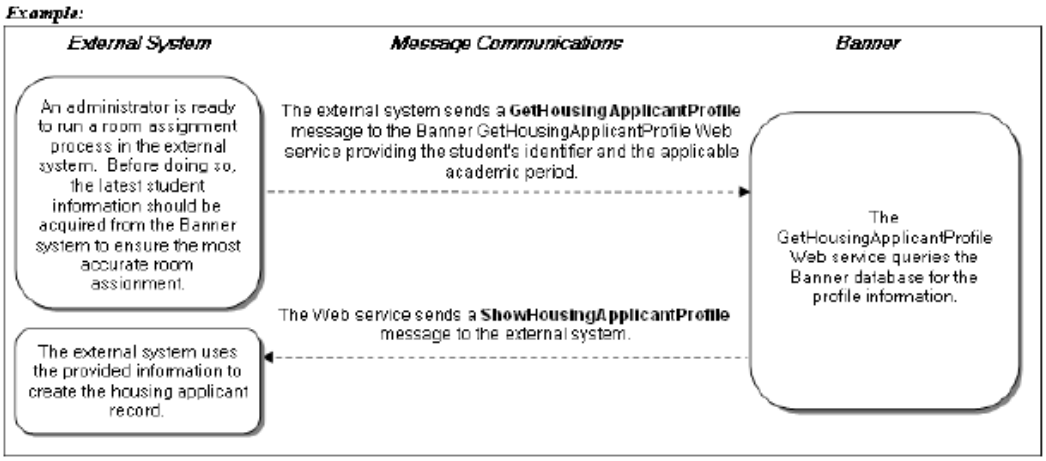
Creation of housing applicant record

External systems can use the GetHousingApplicantProfile Web service to create a housing applicant record.



Room assignment process

External systems can use the GetHousingApplicantProfile Web service to obtain additional information that can affect an applicant's actual room assignment.



Setup requirements

Settings on the Crosswalk Validation Form (GTVSDAX) specify the Banner-specific values that the GetHousingApplicantProfile Web service uses to select contact information and participation information for housing applicants.

Note

These settings are not required. If a setting does not exist, the related information is not included in the ShowHousingApplicantProfile message. ■

Contact information

GTVSDAX settings determine the following contact information provided for the housing applicant:

- Mailing address
- Telephone number
- E-mail address

Mailing address

This GTVSDAX setting specifies the address type codes that the GetHousingApplicant Profile Web service uses to select the applicant's mailing address. There is no limit to the number of address type codes that can be included.

The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an address matching that rule is not found for the applicant, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	Appropriate value from `STVATYP_CODE`

Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One address per applicant

Telephone number

This GTVSDAX setting specifies the telephone type codes that the GetHousingApplicant Profile Web service uses to select the applicant's telephone number. There is no limit to the number of telephone type codes that can be included.

The Web service uses the GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the applicant.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the applicant.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the applicant.
- If no matching telephone number is found, the telephone number linked to the mailing address specified in the Mailing Address GTVSDAX rule is selected for the applicant.
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the mailing address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	Appropriate value from `STVTELE_CODE`
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One contact telephone per applicant

E-mail address

This GTVSDAX setting specifies the e-mail address type codes that the GetHousingApplicantProfile Web service uses to select the applicant's e-mail address. There is no limit to the number of e-mail address type codes that can be included.

The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Web service uses the rule with the lowest sequence number first. If an e-mail address matching that rule is not found for the applicant, it evaluates the next lowest sequence number until a matching e-mail address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*EMAIL*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	Appropriate value from `GTVEMAL_CODE`
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One contact e-mail address per applicant

Participation information

GTVSDAX settings specify the following activity participation that can potentially affect an applicant's housing assignment:

- Athletic participation
- Extracurricular activity participation

For example, if members of athletic teams or fraternities are housed together, these rules can be used to provide the information needed to assign applicants to specific buildings or rooms.

Athletic participation

This GTVSDAX setting specifies the activity codes that the GetHousingApplicant Profile Web service uses to select the applicant's athletic participation information. There is no limit to the number of activity codes that can be included. All athletic participation

records matching the codes set up on GTVSDAX are included in the `AthleticParticipation` elements of the ShowHousingApplicantProfile message.

Athletic team participation is recorded on the Student Sport Form (SGASPRT).

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*SPORT*
Sequence:	Not required
External Code:	Appropriate value from `STVACTC_CODE` (must have `STVACTC_ACTP_CODE` value of *SPRTS*)
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Extracurricular activity participation

This GTVSDAX setting specifies the activity codes that the GetHousingApplicant Profile Web service uses to select the applicant's extracurricular activity participation information There is no limit to the number of activity codes that can be included. All extracurricular activity participation records matching the codes set up on GTVSDAX are included in the ShowHousingApplicantProfile message.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*HOUSINGINT*
Internal Group:	*ACTIVITY*
Sequence:	Not required
External Code:	Appropriate value from `STVACTC_CODE`
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

GetHousingApplicantProfile

There are no dynamic or static translations for the GetHousingApplicantProfile message.

ShowHousingApplicantProfile

The following dynamic translation must be customized:

Element	Banner Value	Transformed to Enterprise Value
`rco:Gender`	*M*	*Male*
	F	*Female*
	N	*NotReported*
`rco:CountryCode`	*STVNATN_CODE*	*STVNATN_NATION*

The following static translations are translated automatically and should not be changed:

Element	Banner Value	Transformed to Enterprise Value
`Deceased`	*Y*	*Yes*
	N	*No*
`CitizenIndicator`	*Y*	*Yes*
	N	*No*
`Incarceration`	*Y*	*Yes*
	N	*No*

Element	Banner Value	Transformed to Enterprise Value
SpecialDisabledVeteran	*Y*	*Yes*
	N	*No*
Disability/@primary	*Y*	*Yes*
	N	*No*

10 GetPersonIdentity

The GetPersonIdentity Web service allows external systems to retrieve the unique identifier for a person from Banner®. The external system can query Banner with known details about a person to obtain the unique identifier and corresponding person details for one or multiple matching records.

This service can locate a person in Banner in two ways:

- The service can accept an inactive Banner ID, a LoginID (used for Luminis Platform access), or an IMSID (used for Banner Integration for eLearning) and return the current Banner ID.
- The service can accept a set of criteria describing a person, and apply a set of Common Matching rules to identify persons in the Banner database who match the submitted criteria.

Message exchange

The GetPersonIdentity Web service exchanges the following messages:

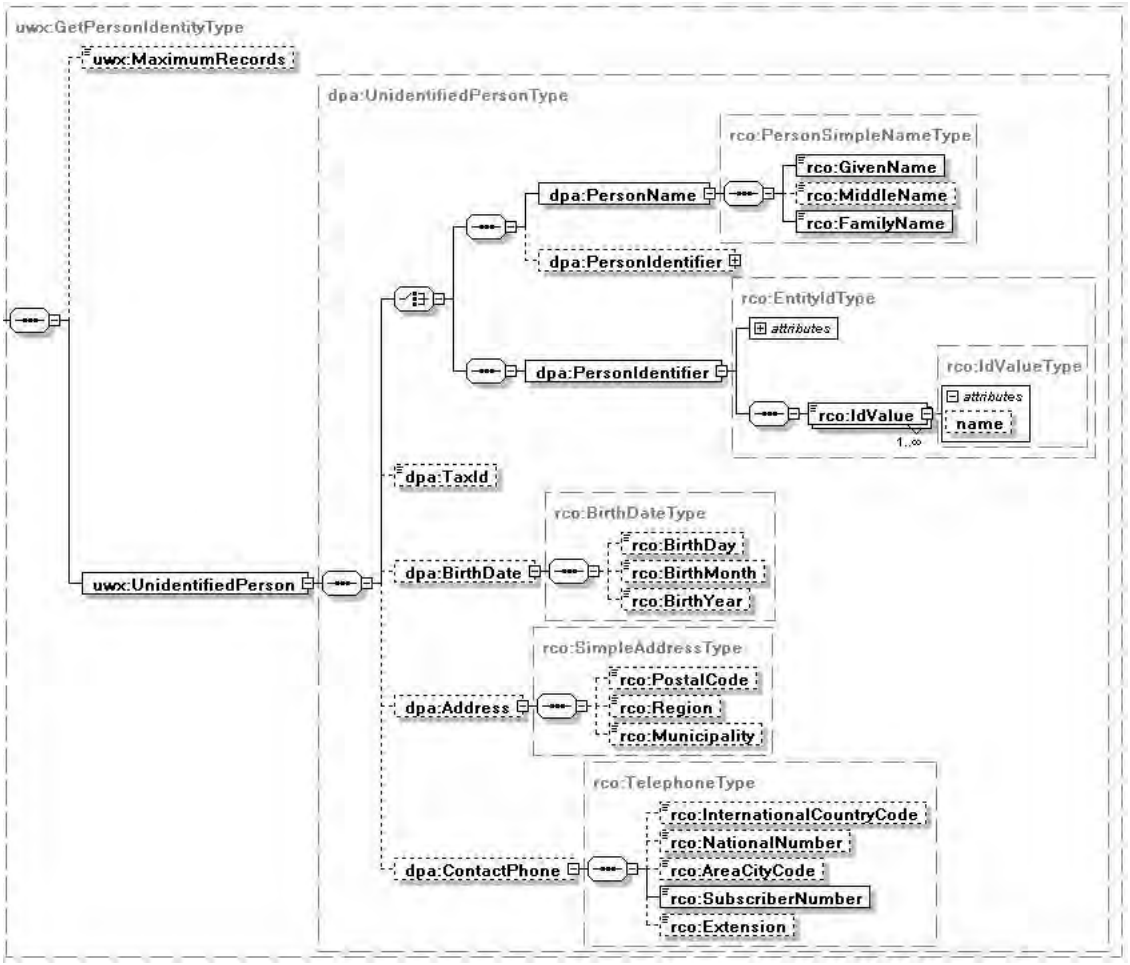
- GetPersonIdentity
- ShowPersonIdentity

GetPersonIdentity

An external system uses the GetPersonIdentity message to pass identifying information about a person to Banner. This information can either be an accepted identifier or elements established in a set of Common Matching rules. The Web service uses this information to identify one or more persons who match the criteria. This message also allows the requestor to specify a limit to the number of possible matches that are returned. If no `MaximumRecords` value is specified, a maximum of 20 possible matches is returned.

The GetPersonIdentity Web service first determines if a `PersonName` element is present. If present, it applies Common Matching rule processing to identify a person or persons matching the rule criteria. If no `PersonName` element exists, the Web service uses the `PersonIdentifier` value to identify a person in Banner.

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



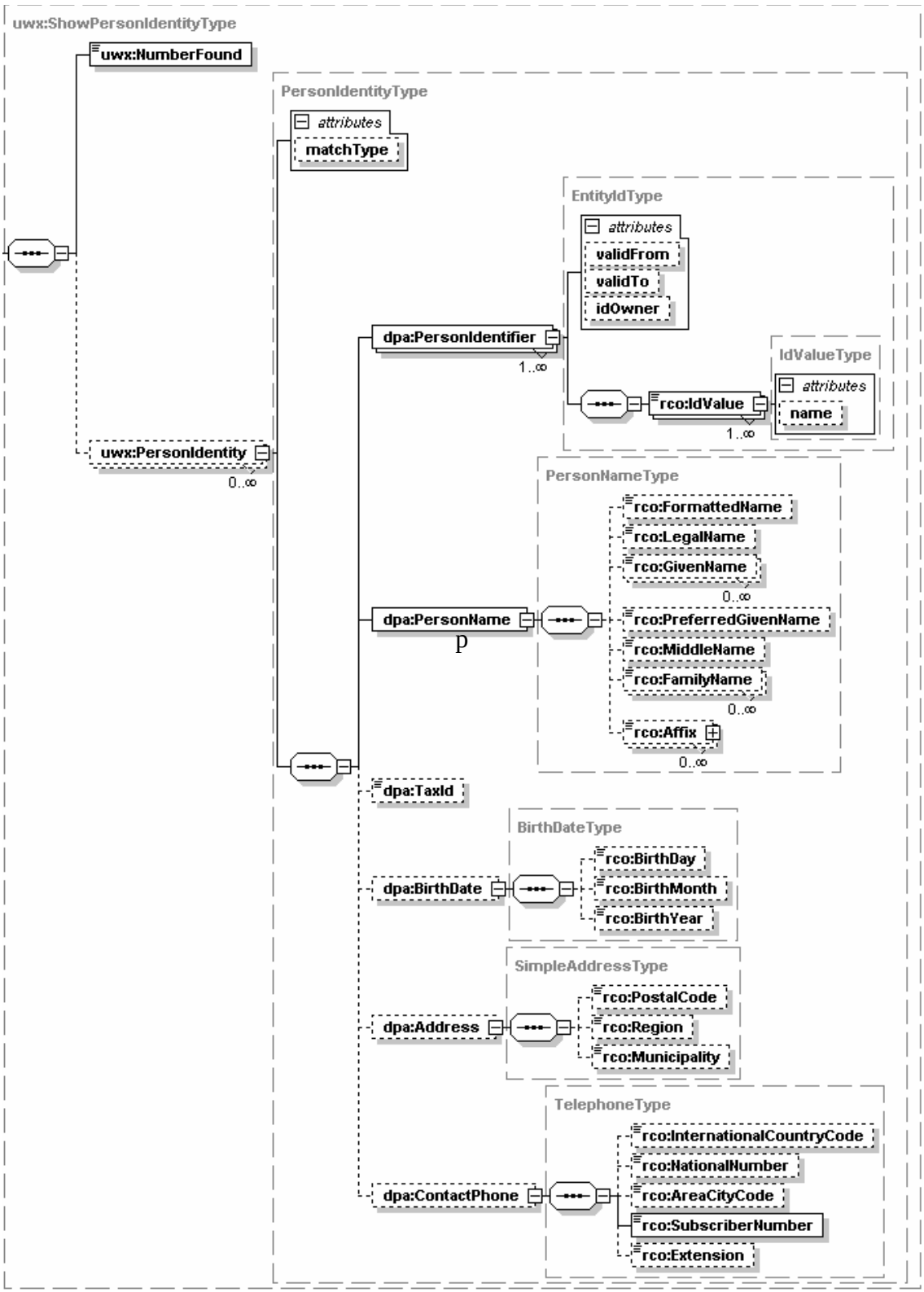
ShowPersonIdentity

If the `PersonName` element is present in the GetPersonIdentity message and Common Matching rule processing is used, the ShowPersonIdentity message returns one of the following:

- One `PersonIdentity` element if a unique match is found
- Multiple `PersonIdentity` elements if several persons match the criteria
- An empty `PersonIdentity` element if no person matches the criteria

Each match or potential match contains an attribute specifying if it is a match or a possible match. If the Web service can determine the address and phone number that resulted in a match or potential match, that information is also returned.

If a `PersonIdentifier` is provided instead of a `PersonName`, the ShowPersonIdentity message returns one `PersonIdentity` element if a match is found in Banner. The following diagram shows the structure of the response message schema.


uwx:ShowPersonIdentityType
uwx:NumberFound
PersonIdentityType
attributes
matchType
EntityIdType
attributes
validFrom
validTo
idOwner
dpa:PersonIdentifier
1..∞
IdValueType
attributes
name
rco:IdValue
1..∞
uwx:PersonIdentity
0..∞
PersonNameType
rco:FormattedName
rco:LegalName
rco:GivenName
0..∞
dpa:PersonName
rco:PreferredGivenName
rco:MiddleName
rco:FamilyName
0..∞
rco:Affix
0..∞
dpa:TaxId
BirthDateType
dpa:BirthDate
rco:BirthDay
rco:BirthMonth
rco:BirthYear
SimpleAddressType
dpa:Address
rco:PostalCode
rco:Region
rco:Municipality
TelephoneType
dpa:ContactPhone
rco:InternationalCountryCode
rco:NationalNumber
rco:AreaCityCode
rco:SubscriberNumber
rco:Extension

SOAP fault messages

If a valid response cannot be created as a ShowPersonIdentity message, a SOAP fault message is returned instead. Situations that might cause a SOAP fault message include the following:

- The `PersonIdentifier` provided in the GetPersonIdentity message is not found in Banner.
- The Common Matching source code that identifies the appropriate set of rules to use cannot be identified.
- A network, database, or other technical issue occurs.
- The applicable GTVSDAX rows for this service have the value *UPDATE ME* in the **External Code** field.

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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

GetPersonIdentity

Element/Attribute	Database Mapping
`GetPersonIdentity`	
`MaximumRecords`	NA
`UnidentifiedPerson`	
`PersonName`	
`GivenName`	`SPRIDEN_FIRST_NAME`
`MiddleName`	`SPRIDEN_MI`
`FamilyName`	`SPRIDEN_LAST_NAME`
`PersonIdentifier`	

Element/Attribute	Database Mapping
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
TaxId	SPBPERS_SSN
BirthDate	
BirthDay	SPBPERS_BIRTH_DATE
BirthMonth	SPBPERS_BIRTH_DATE
BirthYear	SPBPERS_BIRTH_DATE
Address	
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
ContactPhone	
InternationalCountryCode	NA
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	NA

ShowPersonIdentity

Element/Attribute	Database Mapping
ShowPersonIdentity	
NumberFound	NA
PersonIdentity	
@matchType	GOTCMRT_RESULT_IND
PersonIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
PersonName	
FormattedName	SPBPERS_NAME_PREFIX || ' ' || SPRIDEN_FIRST_NAME || ' ' || SPRIDEN_MI || ' '|| SPRIDEN_LAST_NAME || ', ' || SPBPERS_NAME_SUFFIX
LegalName	SPBPERS_LEGAL_NAME
GivenName	SPRIDEN_FIRST_NAME
PreferredGivenName	SPBPERS_PREF_FIRST_NAME
MiddleName	SPRIDEN_MI
FamilyName	SPRIDEN_LAST_NAME
Affix	SPBPERS_NAME_PREFIX

Element/Attribute	Database Mapping
@type	NA
Affix	SPBPERS_NAME_SUFFIX
@type	NA
TaxId	SPBPERS_SSN
BirthDate	
BirthDay	SPBPERS_BIRTH_DATE
BirthMonth	SPBPERS_BIRTH_DATE
BirthYear	SPBPERS_BIRTH_DATE
Address	
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
ContactPhone	
InternationalCountryCode	NA
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	NA

Intended usage

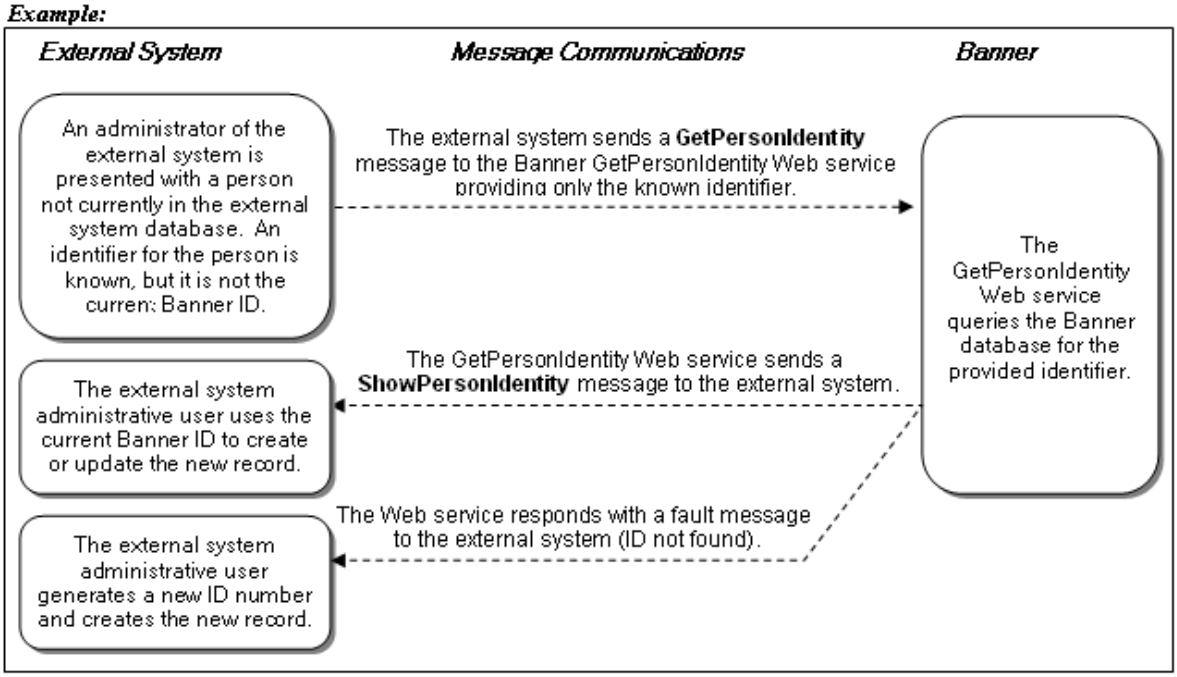
External systems can use the GetPersonIdentity Web service to retrieve the unique person identifier from Banner.

Current Banner ID not known

The GetPersonIdentity Web service can be used when the external system does not know the current Banner ID but the person in question is known by another identifier. The current Banner ID may be required by other Banner Web services.

The external system can provide an inactive BannerUID, a LogonID (used for Luminis Platform access), or an IMSID (used for Banner Integration for eLearning) in the GetPersonIdentity message. The GetPersonIdentity Web service uses the provided identifier to search for the current BannerUID (Banner ID).

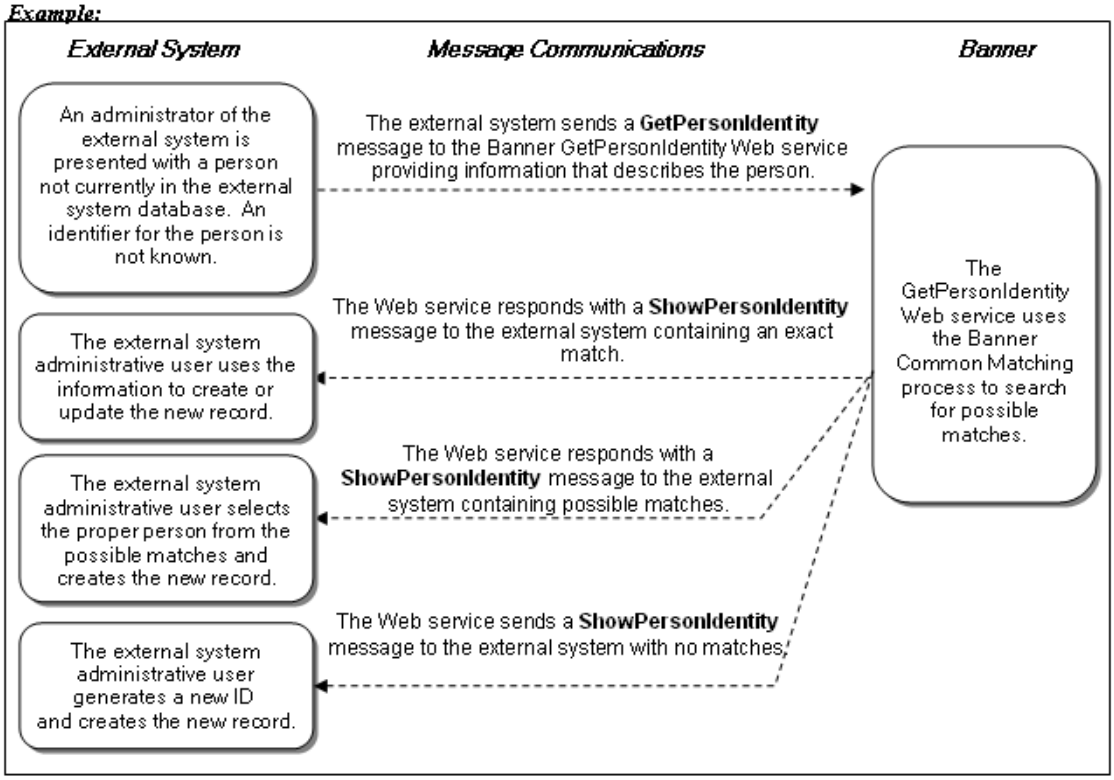
Example:



No Banner ID known

If no identifier is known, the GetPersonIdentity Web service can still be used to locate a person in the Banner database using Common Matching functionality. The external system produces a GetPersonIdentity message without a `PersonIdentifier` element but with a name and other descriptive information. Common Matching rules must be created in Banner to process these queries.

Example:



Setup requirements

The GetPersonIdentity Web service has two setups:

- Setting on the Crosswalk Validation Form (GTVSDAX) used by Common Matching processing
- Maximum number of matching records

GTVXDAX setup

The GetPersonIdentity Web service uses Common Matching rules to locate persons in the Banner database. The Web service determines which set of Common Matching rules to use as follows:

- If user INTEGMGR has a default Common Matching source code defined on the Common Matching User Setup Form (GORCMUS), this source code is used to specify which set of Common Matching rules is used. (INTEGMGR is the userID that Ellucian integration software uses to connect to the Banner database.)

- If user INTEGMGR does not have a default Common Matching source code defined on GORCMUS, a GTVSDAX setting is used to specify which set of Common Matching rules is used.

Note

This GTVSDAX setting is not required if user INTEGMGR has a default Common Matching source code defined on GORCMUS.

Internal Code:	*INTEG*
Internal Group:	*CM_SOURCE_CODE*
External Code:	Common Matching source (appropriate value from `GTVCMSC_CMSC_CODE`)
Code Description:	Description of this rule
Limit on GTVSDAX:	One
Message Limit:	NA

The Common Matching source code associated with the GTVSDAX rule or the user default should be set up on the Common Matching Source Code Validation Form (GTVCMSC). Common Matching rules should be set up for that source code on the Common Matching Source Rules Form (GORCMSC) and the Common Matching Rules Form (GORCMRL). Refer to the *Banner General Common Matching Handbook* for more information about Common Matching processing.

Maximum matching records

The `MaximumRecords` element in the GetPersonIdentity message can be used to specify the maximum number of possible matching records that your institution wants returned in each ShowPersonIdentity message. If no maximum number is provided, a maximum of 20 possible matches is returned in each ShowPersonIdentity response message.

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

GetPersonIdentity

There are no dynamic or static translations for the GetPersonIdentity message.

ShowPersonIdentity

There are no dynamic translations for the ShowPersonIdentity message.

The following static translation is translated automatically and should not be changed:

Element	Banner Value	Transformed to Enterprise Value
`@matchType`	*M* S	*Match* *Possible Match*

11 ReleaseStudentDeposit

The ReleaseStudentDeposit Web service allows external systems to request the release of funds held on deposit on a student's account in Banner®. These funds may be released to cover damages charged to a student's account or they may be remitted to the student at the end of a specified period.

Note

This Web service requires Banner Accounts Receivable.

Message exchange

The ReleaseStudentDeposit Web service exchanges the following messages:

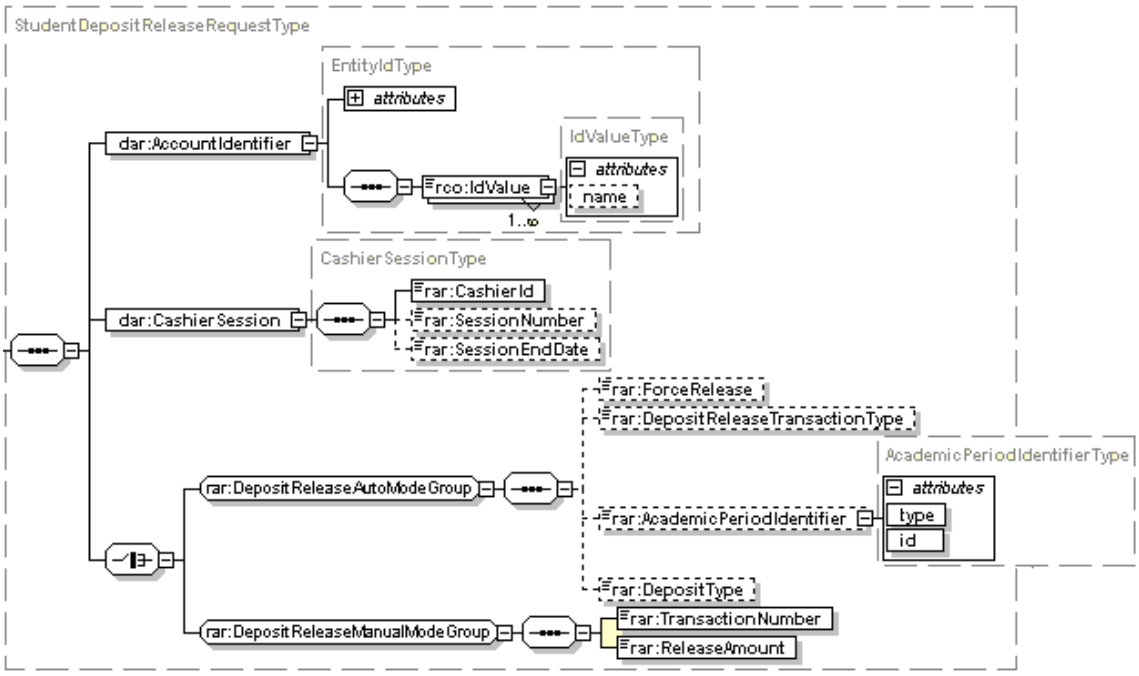
- ReleaseStudentDeposit
- ConfirmReleaseStudentDeposit

ReleaseStudentDeposit

An external system uses the ReleaseStudentDeposit message to request the release of deposit funds for a student. The message contains elements that describe how funds should be released. The Web service can do one of the following:

- Process the release of an individual deposit.
- Process the release of multiple deposits according to the auto release flag and date.
- Force the release of multiple deposits meeting specified criteria.

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



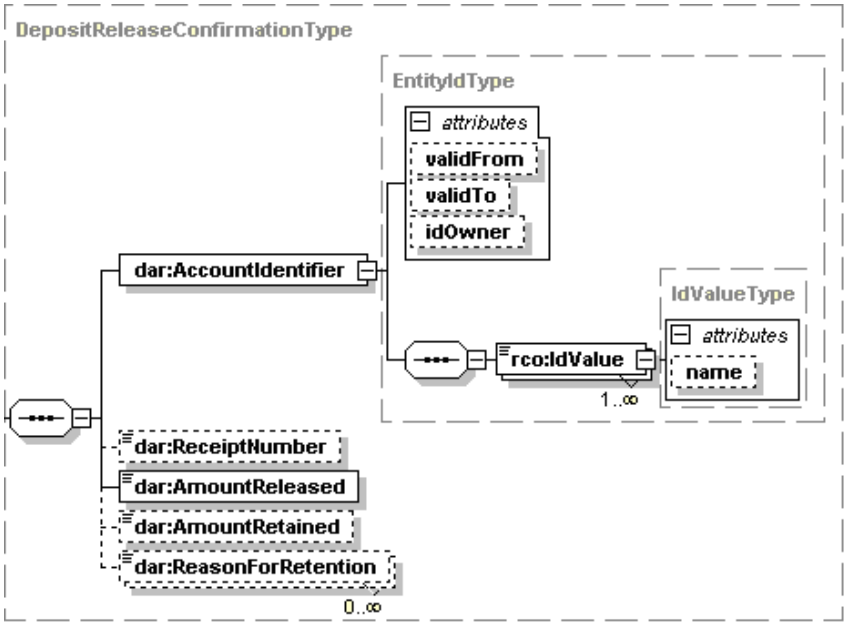
ConfirmReleaseStudentDeposit

If the request is processed successfully, the ReleaseStudentDeposit Web service responds with a ConfirmReleaseStudentDeposit message that contains the following information:

- Elements identifying the transaction
- Amount actually released
- Amount retained on record

Because it is possible to successfully process the message without releasing funds, this information informs the calling system of the final disposition of the request.

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



SOAP fault messages

If a valid response cannot be created as a ConfirmReleaseStudentDeposit message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `AccountIdentifier` provided in the ReleaseStudentDeposit message is not a valid LogonID, IMSID, or current Banner ID (BannerUID).
- The ReleaseStudentDeposit message contains values that are not recognized by Banner.
- Completion of the request would violate Banner business rules required by the `tb_deposit` business entity API. For more information about the API, refer to the *Banner Accounts Receivable TRM Supplement*.
- The same ReleaseStudentDeposit request is executed more than once using manual mode.
- 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.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

ReleaseStudentDeposit

Element/Attribute	Database Mapping
ReleaseStudentDeposit	
StudentDepositReleaseRequest	
AccountIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
CashierSession	
CashierId	TBRDEPO_USER
SessionNumber	TBRDEPO_SESSION_NUMBER
SessionEndDate	TBRDEPO_CSHR_END_DATE
ForceRelease	NA
DepositReleaseTransactionType	TBBDEPC_DETAIL_CODE
AcademicPeriodIdentifier	STVTERM_DESC
@id	TBRDEPO_TERM_CODE
@type	NA

Element/Attribute	Database Mapping
DepositType	TBBDEPC_DTYP_CODE
TransactionNumber	TRBDEPO_TRAN_NUMBER
ReleaseAmount	NA

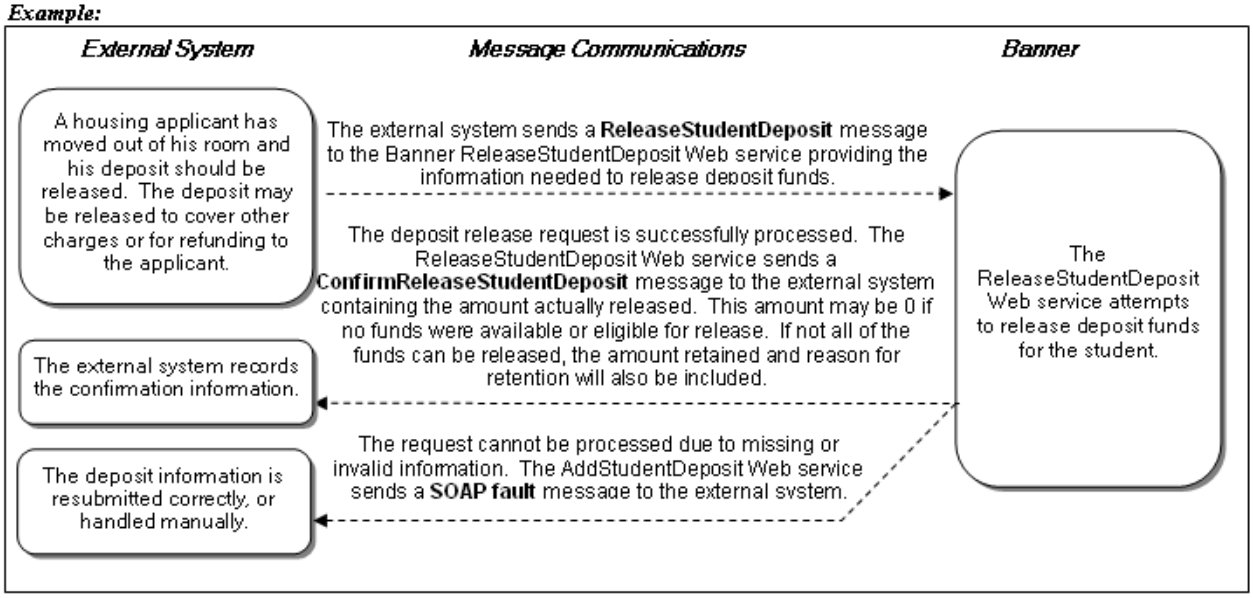
ConfirmReleaseStudentDeposit

Element/Attribute	Database Mapping
ConfirmReleaseStudentDeposit	
StudentDepositReleaseConfirmation	
AccountIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	GOBSRID_SOURCED_ID
@name	NA
IdValue	GOBTPAC_EXTERNAL_USER
@name	NA
ReceiptNumber	TBRDEPO_RECEIPT_NUMBER
AmountReleased	NA
AmountRetained	NA
ReasonForRetention	NA

Intended usage

External systems can use the ReleaseStudentDeposit Web service to release funds held on deposit for a student in Banner.

Example:



Setup requirements

Your institution must coordinate the use of Banner-specific values with an external system. The external system must provide the values required by the transaction.

Translations

The contents of some elements require translation between enterprise values and Banner values.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service.

Static translations are translated automatically and should not be changed.

ReleaseStudentDeposit

There are no dynamic translations for the ReleaseStudentDeposit message.

The following static translation is translated automatically and should not be changed:

Element	Enterprise Value	Transformed to Banner Value
`ForceRelease`	*Yes*	*Y*
	No	*N*

ConfirmReleaseStudentDeposit

There are no dynamic or static translations for the ConfirmReleaseStudentDeposit message.

12 SyncEligibleCardholder

Unlike the other Web services described in this handbook, SyncEligibleCardholder is not a request/reply Web service that Banner® exposes. Rather, SyncEligibleCardholder is a Web service interface (portType) that external campus card systems can implement to facilitate integration with Banner. By exposing an endpoint that adheres to this interface and its related SOAP binding, the external system can accept SyncEligibleCardholder messages that are posted by the Banner Cardholder Event Publisher when cardholder information in Banner changes.

The `EligibleCardholder` XML structure aggregates data from several Banner tables that are used by campus card systems. Campus card systems can use this data to create or update a cardholder record and grant associated privileges.

As an interface, SyncEligibleCardholder acts as a template or blueprint for systems that want to receive information from Banner in a defined, standardized format. The interface defines the structure of the SyncEligibleCardholder message and the message exchange pattern, but nothing more.

SyncEligibleCardholder message

The SyncEligibleCardholder interface defines one message: SyncEligibleCardholder. This is a “fire-and-forget” message. The Banner Cardholder Event Publisher sends the message to the external campus card system and does not wait for a response.

The Banner Cardholder Event Publisher produces the SyncEligibleCardholder message when data in the `EligiblelCardholder` XML structure changes. This includes changes to the following data in Banner:

- Name
- Biographical information
- Permanent and local contact information
- Campus residence location
- Campus work location
- Meal assignments
- Cardholder roles

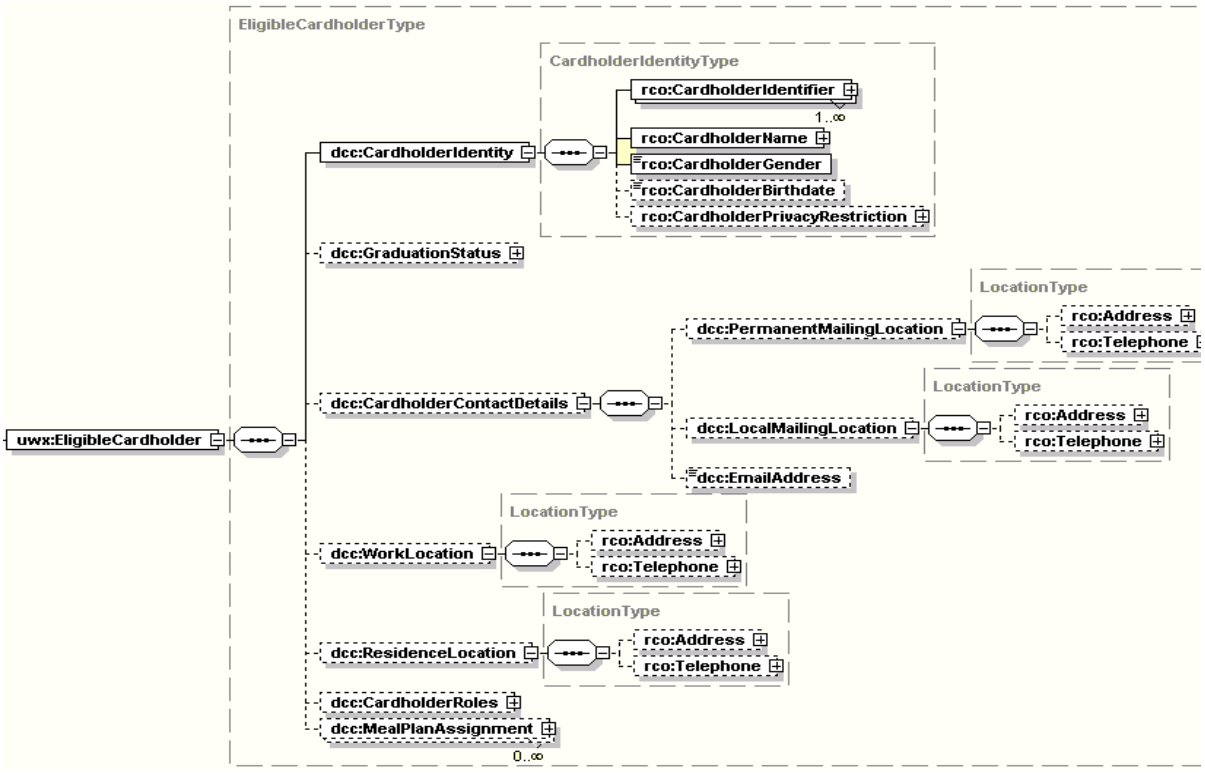
Cardholder roles can be used to determine if a person is eligible for a card and which privileges that card should allow. Cardholder roles are used to categorize potential cardholders or apply a custom set of criteria, based on information stored in Banner, to

determine eligibility. For example, a role can be set up on the Business Rules Form (GORRSQL) to support a requirement that only students who are enrolled in the current term are allowed to own a campus card. A person can qualify for multiple cardholder roles. All roles for which the person qualifies are included in the message.

The SyncEligibleCardholder Web service interface does not require cardholder roles. If either of the following scenarios occurs, the SyncEligibleCardholder message does not include the `CardholderRoles` element:

- There are no active rules for cardholder roles on GORRSQL.
- The person does not meet the criteria for any active role.

The following diagram shows the structure of the `EligibleCardholder` element used in the SyncEligibleCardholder message.



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 also nest inside other elements.

For a detailed description of the message elements/attributes and their properties, refer to the Ellucian XML Schema documentation, available with the downloaded software in the `\banner_service_repository\html_doc` directory.

Element/Attribute	Database Mapping
SyncEligibleCardholder	
EligibleCardholder	
CardholderIdentity	
CardholderIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
CardholderName	
FormattedName	SPBPERS_NAME_PREFIX SPRIDEN_FIRST_NAME SPRIDEN_MI SPRIDEN_LAST_NAME SPBPERS_NAME_SUFFIX
LegalName	SPBPERS_LEGAL_NAME
GivenName	SPRIDEN_FIRST_NAME
PreferredGivenName	SPBPERS_PREF_FIRST_NAME
MiddleName	SPRIDEN_MI
FamilyName	SPRIDEN_LAST_NAME
Affix	SPBPERS_NAME_PREFIX
@type	NA
Affix	SPBPERS_NAME_SUFFIX
@type	NA

Element/Attribute	Database Mapping
CardholderGender	SPBPERS_SEX
CardholderBirthdate	SPBPERS_BIRTH_DATE
CardholderPrivacyRestriction	
PrivacyRestrictionDate	NA
PrivacyRestrictionLevel	SPBPERS_CONFID_IND
GraduationStatus	
ExpectedGraduationDate	SGBSTDN_EXP_GRAD_DATE
ExpectedGraduationPeriod	STVTERM_DESC
@id	SGBSTDN_TERM_CODE_GRAD
@type	NA
ExpectedGraduationYear	SGBSTDN_ACYR_CODE
CardholderContactDetails	
PermanentMailingLocation	
Address	
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA

Element/Attribute	Database Mapping
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
DeliveryAddress	
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	

Element/Attribute	Database Mapping
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
LocalMailingLocation	
Address	
@validFrom	SPRADDR_FROM_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	ICBIRULE_INTEGRATION_SOURCE
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP
Region	SPRADDR_STAT_CODE
Municipality	SPRADDR_CITY
DeliveryAddress	

Element/Attribute	Database Mapping
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
EmailAddress	GOREMAL_EMAIL_ADDRESS
WorkLocation	

Element/Attribute	Database Mapping
Address	
@validFrom	NA
@type	NA
@validTo	NA
AddressIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	NA
@name	NA
CountryCode	STVNATN_NATION
PostalCode	PTRJBLN_ZIPC_CODE
Region	PTRJBLN_STAT_CODE
Municipality	PTRJBLN_CITY
DeliveryAddress	
AddressLine	PTRJBLN_DESC
AddressLine	PTRJBLN_ADDRESS1
AddressLine	PTRJBLN_ADDRESS2
StreetName	NA
BuildingNumber	NA
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA

Element/Attribute	Database Mapping
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT
ResidenceLocation	
Address	
@validFrom	SPRADDR_FROM_DATE
@validFrom	SLRRASG_BEGIN_DATE
@type	STVATYP_DESC
@validTo	SPRADDR_TO_DATE
@validTo	SLRRASG_END_DATE
AddressIdentifier	
@validFrom	NA
@idOwner	NA
@validTo	NA
IdValue	SPRIDEN_ID
@name	NA

Element/Attribute	Database Mapping
IdValue	SPRADDR_ATYP_CODE
@name	NA
IdValue	SPRADDR_SEQNO
@name	NA
CountryCode	STVNATN_NATION
PostalCode	SPRADDR_ZIP or SLBBLDG_ZIP
Region	SPRADDR_STAT_CODE or SLBBLDG_STAT_CODE
Municipality	SPRADDR_CITY or SLBBLDG_CITY
DeliveryAddress	
AddressLine	SPRADDR_STREET_LINE1
AddressLine	SPRADDR_STREET_LINE2
AddressLine	SPRADDR_STREET_LINE3
AddressLine	SPRADDR_STREET_LINE4
StreetName	NA
BuildingNumber	SPRADDR_HOUSE_NUMBER
Unit	NA
PostOfficeBox	NA
ContactName	
FormattedName	NA
LegalName	NA
GivenName	NA
PreferredGivenName	NA
MiddleName	NA
FamilyName	NA

Element/Attribute	Database Mapping
Affix	NA
@type	NA
Telephone	
InternationalCountryCode	SPRTELE_CTRY_CODE_PHONE
NationalNumber	NA
AreaCityCode	SPRTELE_PHONE_AREA or SLRPASG_PHONE_AREA
SubscriberNumber	SPRTELE_PHONE_NUMBER or SLRPASG_PHONE_NUMBER
Extension	SPRTELE_PHONE_EXT or SLRPASG_PHONE_EXT
CardholderRoles	
Role	GORRSQL_SQRU_CODE
MealPlanAssignment	
@status	STVMSCD_DESC
MealPlanDescription	STVMRCD_DESC
@id	SLRMASG_MRCD_CODE
BeginDate	SLRMASG_BEGIN_DATE
EndDate	SLRMASG_END_DATE

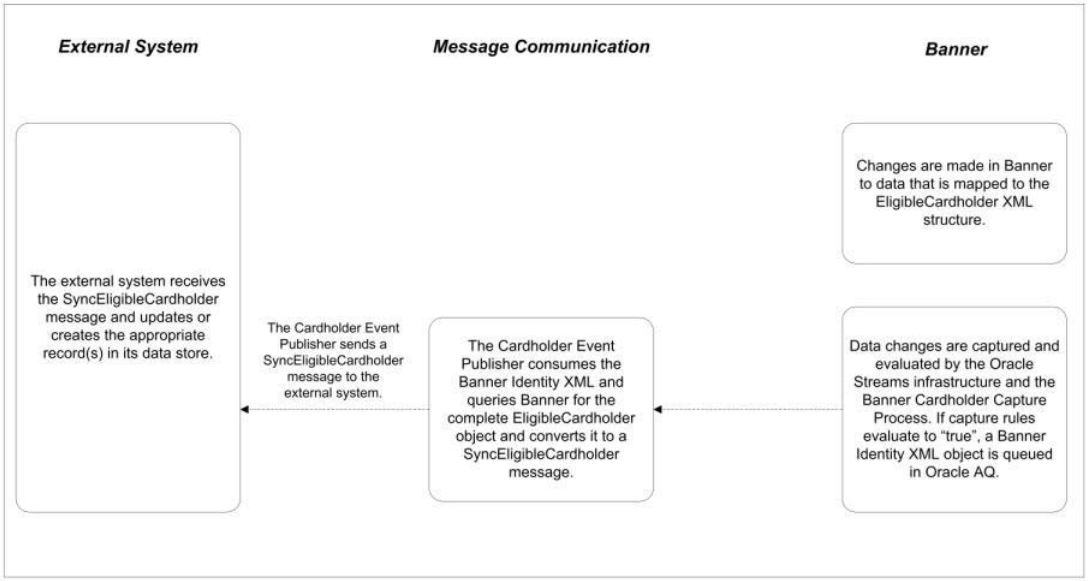
Intended usage

The SyncEligibleCardholder interface facilitates the real-time publication of data from Banner to an external campus card system, keeping the campus card system synchronized with changing Banner information. Data is published by Banner and used to update cardholder or patron records in the campus card system so that contact information, access to buildings, and meal plan authorization are up-to-date.

To ensure communication, both ends of the exchange require software components that comply with the SyncEligibleCardholder interface:

- The campus card system must expose an endpoint that accepts a SyncEligibleCardholder message wrapped in a SOAP envelope transmitted via http.
- Banner requires the following components:
  - Oracle Streams must be enabled in the Banner database.
  - The cardholder capture process must be compiled from rules defined for cardholder eventing on the Streams Rules Configuration Form (GUASADM).
  - The Banner Cardholder Event Publisher must be deployed to an Oracle application server and configured to communicate to the Banner database, the Banner Translation Service, and the external campus system.

With these components in place and properly configured, the external system receives notice of changes to Banner data and can create or update the appropriate cardholder records.



Setup requirements

The following setup requirements ensure the proper publication of SyncEligible Cardholder messages from Banner. These requirements are the same as those for GetEligibleCardholder.

Initial roles and rules are provided for the Banner Cardholder Event Publisher to create SyncEligibleCardholder messages. These roles and rules should be installed in Banner with the other components. You can modify these roles and rules on the Business Rules

Form (GORRSQL) and Crosswalk Validation Form (GTVSDAX). After completing the setup activities, your technical staff should run the `campuscard_check.sql` utility script to verify that the rules exist and that no unused GTVSDAX records remain. If the utility finds GTVSDAX records with an **External Code** set to *UPDATE ME*, you should remove or update these records before using the Banner Cardholder Event Publisher.

GORRSQL setup

The SyncEligibleCardholder message includes cardholder roles as part of the information that is used to determine if a person is eligible for a card and which privileges that card should allow. Cardholder roles are used to categorize potential cardholders or apply a custom set of criteria, based on information stored in Banner, to determine eligibility. For example, a rule can be set up on GORRSQL to support the requirement that only students who are enrolled in the current term are allowed to own a campus card.

Each GORRSQL rule for the Banner Cardholder Event Publisher must use process code *CARDHOLDER_ROLES* and a unique rule code that identifies the rule. Sample roles are provided. To meet your institution's needs, you can inactivate the sample roles and add new ones. The rule code value (`GORRSQL_SQRU_CODE`**)** is provided as the `Role` element in the SyncEligibleCardholder message, so the value should be descriptive enough to be understood by an external system user.

Note

GORRSQL is used to define rules for various Banner processes. The SyncEligibleCardholder message uses only those rules that have process code *CARDHOLDER_ROLES*.

The SyncEligibleCardholder message does not require cardholder roles. If there are no active rules on GORRSQL for cardholder roles, the SyncEligibleCardholder message does not include the `CardholderRoles` element.

Use the following steps to establish a new role for eligible cardholders:

1. Create a rule code on the Business Rule Code Validation Form (GTVSQRU).
2. Access the Business Rules Form (GORRSQL).
3. Enter *CARDHOLDER_ROLES* as the **Process** in the key block.
4. Enter the new rule code in the **Rule** field.
5. In the Rule Data block, enter an SQL SELECT statement that will select the population of persons from Banner who should be given the role. Consider the following when entering your statement:
    - The SELECT clause must select a PIDM, and only a PIDM, from any Banner table that contains a PIDM column.

- The PIDMs you select must be unique. A SELECT statement that selects the same PIDM more than once might result in an error when the Banner Cardholder Event Publisher is called.
- If you want to exclude deceased persons from the eligible cardholder populations, enter a SELECT statement that excludes them.
- The FROM clause can contain more than one table (that is, a join), but the SELECT clause can select a PIDM column from only one table.
- A `:TERM` parameter is available for use in the SELECT statement. When used with the *CARDHOLDER_ROLES* process code, the current term is defined as the highest term code in the set of terms having a start date less than or equal to today's date and an end date greater than or equal to today's date.

This is an example SELECT statement:

```
SELECT SGBSTDN_PIDM
FROM SGBSTDN A,STVSTST
WHERE A.SGBSTDN_STST_CODE = STVSTST_CODE
      AND STVSTST_REQ_IND = 'Y'
      AND A.SGBSTDN_TERM_CODE_EFF =
         (SELECT MAX (B.SGBSTDN_TERM_CODE_EFF)
         FROM SGBSTDN B
         WHERE B.SGBSTDN_PIDM = A.SGBSTDN_PIDM
            AND B.SGBSTDN_TERM_CODE_EFF <= :TERM)
```

6. Use the **Validate** button to validate the SQL statement.

7. Select the **Active** checkbox to activate the rule.

Once the rule is validated and activated, the Banner Cardholder Event Publisher automatically uses the rule when determining a cardholder's roles.

You can modify the rules defined on GORRSQL at any time to change the definition of cardholder roles. However, the roles assigned to cardholders already known to the external campus card system are not re-evaluated until the campus card system requests such information or until a change to EligibleCardholder data results in the reevaluation of cardholder roles for a person.

Rules that are delivered as seed data cannot be deleted nor modified, but they can be inactivated. If a delivered rule is not needed for a certain role, clear the **Active** checkbox on the rule and insert a new rule record with a higher sequence number. If more than one active SELECT statement for a given process/rule (each with a unique sequence number) exists, a person in Banner need only be selected by one SELECT statements to be given that role.

GTVSDAX setup

Settings on the Crosswalk Validation Form (GTVSDAX) specify what data is used for cardholder information:

- Contact details
- Residence location
- Work location
- Meal plan assignment status

Contact details

Staff who manage the campus card system use contact details to contact the cardholder. The following GTVSDAX settings specify what contact information is provided in the response message:

- E-mail address
- Permanent location/address type
- Permanent location/telephone type
- Local location/address type
- Local location/telephone type

Note

These settings are not required. If a setting does not exist, the related information is not included in the message.

E-mail address

This GTVSDAX setting determines the e-mail address type codes that the Banner Cardholder Event Publisher uses to select the cardholder's e-mail address.

There is no limit to the number of e-mail address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Publisher uses the rule with the lowest sequence number first. If an e-mail address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching e-mail address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_EMAIL*
Internal Group:	*EMAIL*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`GOREMAL_EMAL_CODE` value for the e-mail type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Permanent location/address type

This GTVSDAX setting specifies the address type codes that the Banner Cardholder Event Publisher uses to select the cardholder's permanent mailing address.

There is no limit to the number of address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Publisher uses the rule with the lowest sequence number first. If an address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_PERM*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRADDR_ATYP_CODE` value for the address type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Permanent location/telephone type

This GTVSDAX setting specifies the telephone type codes that the Banner Cardholder Event Publisher uses to select the cardholder's permanent telephone number.

There is no limit to the number of telephone type codes that can be included. The Publisher uses GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the cardholder.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If no matching telephone number is found, the telephone number linked to the permanent address specified in the previous GTVSDAX rule is selected for the cardholder
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the permanent address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_PERM*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Local location/address type

This GTVSDAX setting specifies the address type codes that the Banner Cardholder Event Publisher uses to select the cardholder's local mailing address.

There is no limit to the number of address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Publisher uses the rule with the lowest sequence number first. If an address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_LOCAL*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first
External Code:	`SPRADDR_ATYP_CODE` value for the address type code.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Local location/telephone type

This GTVSDAX setting specifies the telephone type codes that the Banner Cardholder Event Publisher uses to select the cardholder's local telephone number.

There is no limit to the number of telephone type codes that can be included. The Publisher uses GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the cardholder.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If no matching telephone number is found, the telephone number linked to the local address specified in the previous GTVSDAX rule is selected for the cardholder.
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the local address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_LOCAL*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Residence location

The residence location information provides the external campus card system with a primary physical campus residence location for the cardholder and a way to contact the cardholder at that location. The following GTVSDAX settings specify the appropriate residence location information to select for the cardholder and the source of this information in Banner:

- Residence location address source
- Residence address type
- Room assignment status
- Residence location telephone source
- Residence telephone type
- Telephone assignment status

Note

These settings are not required. If a setting does not exist, the related information is not included in the message.

Residence location address source

Campus residence location addresses come from one of two places in Banner:

- Address Table (SPRADDR)
- Room Assignment Table (SLRRASG)

This GTVSDAX setting determines which source the Banner Cardholder Event Publisher uses to get the residence location. Use the following information to create or modify records on GTVSDAX:

Internal Code:	*CC_RESADDR*
Internal Group:	*DATASOURCE*
External Code:	*P* = Address Table (SPRADDR) *L* = Room Assignment Table (SLRRASG)
Code Description:	Description of this rule
Limit on GTVSDAX:	One
Message Limit:	NA

Residence address type

If the residence location address source setting on GTVSDAX is *P* (SPRADDR), this GTVSDAX setting specifies the address type code the Banner Cardholder Event Publisher uses to select the cardholder's campus residence address.

There is no limit to the number of address type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Publisher uses the rule with the lowest sequence number first. If an address matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching address is found or no GTVSDAX rules are left to evaluate.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_RESIDE*
Internal Group:	*ADDRESS*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRADDR_ATYP_CODE` value for the address type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Room assignment status

If the residence location address source setting on GTVSDAX is *L* (SLRRASG), this GTVSDAX setting specifies the room assignment status codes that the Banner Cardholder Event Publisher uses to select active room assignments in Banner Location Management.

Multiple GTVSDAX rules are allowed, but only one active room assignment is selected for each cardholder. The Publisher uses the current date (SYSDATE) to identify current active room assignments. If more than one room assignment is active for that date, the assignment with the lowest assignment end date is selected.

Only SLRRASG records with `SLRRASG_ASCD_CODE` are used by the Banner Cardholder Event Publisher.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*ACTIVEROOM*
Internal Group:	*ASSIGNMENT STATUS*
External Code:	`SLRRASG_ASCD_CODE` value for the room assignment status code that indicates an active room assignment
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Residence location telephone source

Campus residence location telephone numbers come from one of two places in Banner:

- Telephone Table (SPRTELE)
- Phone Assignment Table (SLRPASG)

This GTVSDAX setting determines which source the Banner Cardholder Event Publisher uses to get the residence location telephone number. Use the following information to create or modify records on GTVSDAX:

Internal Code:	*CC_RESPHON*
Internal Group:	*DATASOURCE*
External Code:	*P* = Telephone Table (SPRTELE)
	L = Phone Assignments Table (SLRPASG)
Code Description:	Description of this rule
Limit on GTVSDAX:	One
Message Limit:	NA

Residence telephone type

If the residence location telephone type source setting on GTVSDAX is *P* (SPRTELE), this GTVSDAX setting specifies the telephone type codes that the Banner Cardholder Event Publisher uses to select the cardholder's campus residence telephone number.

There is no limit to the number of telephone type codes that can be included. The Publisher uses GTVSDAX rules as follows:

- If one active telephone number matches the rules and is marked as primary, that telephone number is selected for the cardholder.
- If multiple active telephone numbers match the rules and are marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If multiple active telephone numbers match the rules but none is marked as primary, the GTVSDAX rule with the lowest sequence number is used to select the telephone number for the cardholder.
- If no matching telephone number is found and if the residence address source code rule on GTVSDAX is *P*, the telephone number linked to the selected residence address specified is selected for the cardholder.
- An unlisted telephone number is selected only if no active, listed telephone number matches the rules and no telephone number is linked to the selected residence address. The term *Unlisted* is provided in the `SubscriberNumber` element.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*CC_RESIDE*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Telephone assignment status

If the residence location telephone source setting on GTVSDAX is *L* (SLRPASG), this GTVSDAX setting specifies which phone assignment status codes the Banner Cardholder Event Publisher uses to select active telephone assignments in Banner Location Management.

Multiple GTVSDAX rules are allowed, but only one residence location telephone number is selected for each cardholder. The Publisher uses the current date (SYSDATE) to identify current active telephone assignments. If more than one telephone assignment is active for that date, the assignment with the lowest assignment end date is selected.

Only SLRPASG records with this `SLRPASG_PSCD_CODE` are used by the Banner Cardholder Event Publisher.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*ACTIVEPHON*
Internal Group:	*ASSIGNMENTSTATUS*
External Code:	`SLRPASG_PSCD_CODE` value for the phone assignment status code that indicates an active phone assignment.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Work location

Work location information provides the external campus card system with a primary physical work location for the cardholder and a way to contact the cardholder during work hours.

The physical work location does not require a GTVSDAX setting because it is derived from the job location (`NBRJOBS_JBLN_CODE`) that is associated with the person's primary job for the current date. Details about the address are retrieved from the Job Location Rules Form (PTRJBLN). To include work location details in the SyncEligibleCardholder messages, you should define job locations on PTRJBLN and ensure that the **Job Location** field on the Position Definition Form (NBAPOSN) or Employee Jobs Form (NBAJOBS) references a job location defined on PTRJBLN.

A GTVSDAX setting specifies the telephone type codes that the Banner Cardholder Event Publisher uses to select the cardholder's work telephone number. There is no limit to the number of telephone type codes that can be included. The `GTVSDAX_INTERNAL_CODE_SEQNO` is required and is used to determine the order in which the rules are used. The Publisher uses the rule with the lowest sequence number first. If an active telephone number matching that rule is not found for the cardholder, it evaluates the next lowest sequence number until a matching telephone number is found or no GTVSDAX rules are left to evaluate. It does not select an unlisted telephone number unless no active, listed numbers match any of the rules on GTVSDAX. If an unlisted telephone number is selected, the term *Unlisted* is provided in the `SubscriberNumber` element.

Note

This GTVSDAX setting is not required. If the setting does not exist, the work telephone number is not included in the message.

Use the following information to create or modify the rules on GTVSDAX:

Internal Code:	*CC_WORK*
Internal Group:	*TELEPHONE*
Sequence:	Sequence in which the setting is used. The setting with the lowest sequence number is used first.
External Code:	`SPRTELE_TELE_CODE` value for the telephone type code
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	One

Meal plan assignment status

This GTVSDAX setting specifies the meal plan assignment status codes the Banner Cardholder Event Publisher uses to identify active meal plan assignments for cardholders.

Note

This setting is not required. If a setting does not exist, the related information is not included in the message.

Multiple rules are allowed. The Publisher selects all meal plan assignments with a status code found in any of the GTVSDAX meal plan assignment status rules, if the current date (SYSDATE) falls between the assignment start and end dates.

Only SLRMASG records with this `SLRMASG_MSCD_CODE` are used by the Banner Cardholder Event Publisher.

Use the following information to create or modify rules on GTVSDAX:

Internal Code:	*ACTIVEMEAL*
Internal Group:	*ASSIGNMENTSTATUS*
External Code:	`SLRMASG_MSCD_CODE` value for the meal plan assignment status code that indicates an active meal plan assignment.
Code Description:	Description of this rule
Limit on GTVSDAX:	No limit
Message Limit:	No limit

Translations

The contents of some elements of SyncEligibleCardholder messages require translation between enterprise values and Banner values. These translations ensure the proper publication of SyncEligibleCardholder messages from Banner. These translations are the same as those for GetEligibleCardholder.

Dynamic translations must be customized to map enterprise values to Banner values. Refer to the *Banner Translation Service Installation and Administration Guide* for details on editing values in the Banner Translation Service. The following dynamic translation must be customized:

Element	Banner Value	Transformed to Enterprise Value
CardholderGender	*M*	*Male*
	F	*Female*
	N	*Unknown*

Static translations are translated automatically and should not be changed.There are no static translations for the SyncEligibleCardholder message.

A Administering Banner Cardholder Event Publisher

The Banner® Cardholder Event Publisher posts SyncEligibleCardholder messages to an external campus card system when `EligibleCardholder` information in Banner changes. This appendix describes the following administrative tasks associated with the Banner Cardholder Event Publisher:

- Configure the Publisher to communicate with the Banner Translation Service and the external campus card system.
- Administer the underlying Oracle Streams capture and apply processes that notify the Publisher when Banner data changes.
- Produce SyncEligibleCardholder messages for persons having one or more cardholder roles defined in Banner.

These administrative tasks are performed with the Banner Cardholder Event Publisher administrative interface.

Configure Banner Cardholder Event Publisher

Use the following steps to configure the Publisher to communicate with the Banner Translation Service and the SyncEligibleCardholder end point. The end point can be an external campus card system or a test application.

1. Connect to the Banner Cardholder Event Publisher administrative interface:

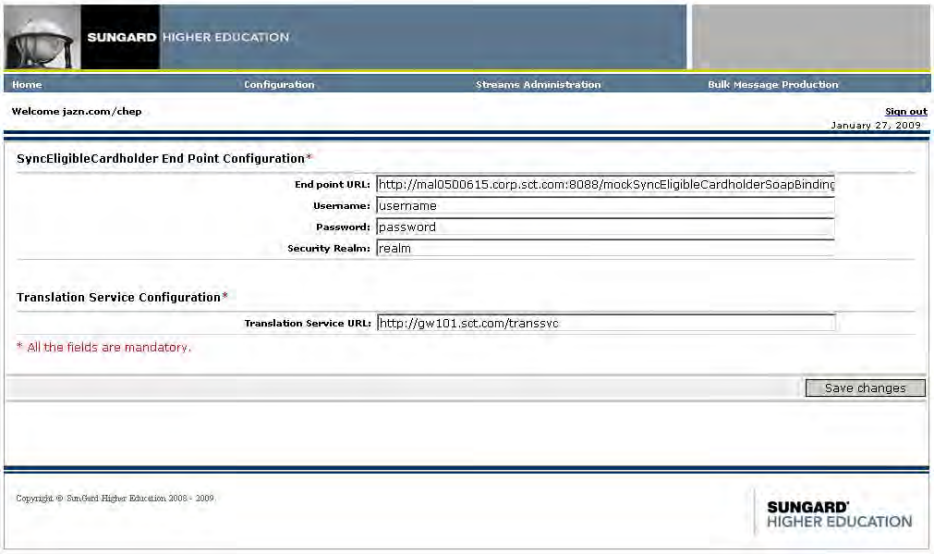
   ```
   http://<host>:<port>/cardholderEventPublisher
   ```

   Note

   This URL assumes that the default URL was used when deploying the Banner Cardholder Event Publisher. Any additions made to the URL during the deployment should be added to the URL as appropriate.

2. Enter the user name and password and click **Sign In**.

3. Select Configuration from the menu bar. The following page is displayed:



4. Enter the following information:

End point URL	URL of the external campus card system's SyncEligibleCardholder end point (provided by the campus card system administrator). Also known as the SyncEligibleCardHolder Web service location.
Username	User name for the campus card system (provided by the campus card system administrator).
Password	Password for the campus card system (provided by the campus card system administrator).
Security Realm	Security realm for the campus card system (provided by the campus card system administrator).
Translation Service URL	URL of the Banner Translation Service administrative interface. Default: is `http://<host>:<port>/transsvc` **Note:** This URL is the location of the administrative interface, not the location of the complete Banner Translation Service.

5. Click **Save Changes**.A message indicates that the update was successful.

Administer Oracle Streams

When `EligibleCardholder` information in Banner changes, the Oracle Streams technology captures, propagates, and applies the changes for use by the Banner Cardholder Event Publisher. The Publisher's administrative interface can be used to perform basic Oracle Streams administrative tasks. The following sections describe these administrative tasks for the Cardholder Oracle Streams processes (CCARD).

Note

You can also administer Oracle Streams directly from an SQL*Plus session or comparable utility. Refer to Appendix B, "Using Oracle Streams" for more details.

Note

The following steps apply only to Oracle Streams processes for the Publisher. If you want to administer Oracle Streams processes for Banner Enterprise Identity Services (processes with process code IAM), use the administrative interface provided for the Banner Identity Gateway.

Start Oracle Streams processes

Use the following steps to start the Oracle Streams processes. You can start the capture and apply processes separately or at the same time.

1. Connect to the Banner Cardholder Event Publisher administrative interface:

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

   Note

   This URL assumes that the default URL was used when deploying the Banner Cardholder Event Publisher. Any additions made to the URL during the deployment should be added to the URL as appropriate.

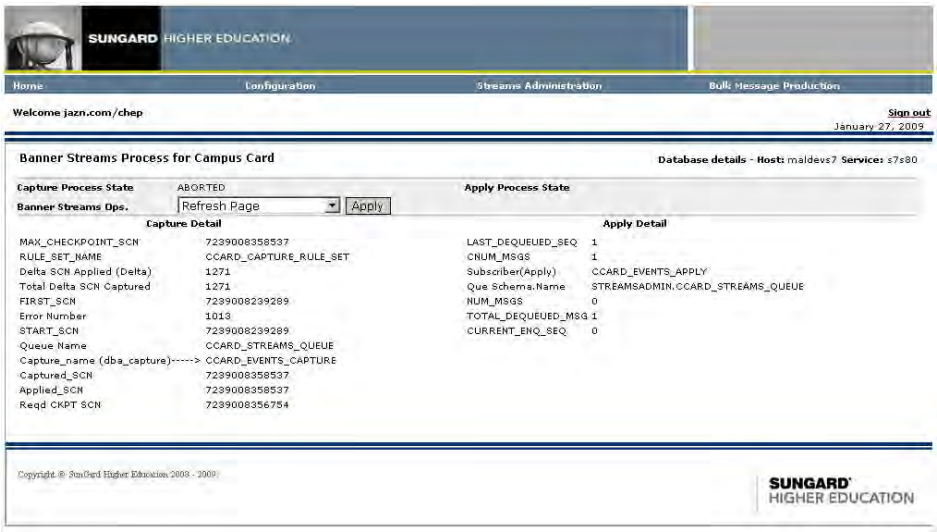
2. Enter the user name and password and click **Sign In**.

3. Select Streams Administration from the menu bar.

4. Verify that the desired process is not already running:

   4.1. Before starting the capture process, **Capture Process State** must have the status *ABORTED*.

   4.2. Before starting the apply process, **Apply Process State** must be null.



5. Select one of the following from the **Banner Streams Ops.** drop-down list:

   Start capture
   Start apply
   Start capture and apply

6. Click **Apply**.

Stop Oracle Streams processes

Stopping the capture and apply processes might be necessary when upgrading middle-tier consumers of events published by the apply process or when changing the capture process configuration.

Use the following steps to stop the Oracle Streams processes. You can stop the capture and apply processes separately or at the same time.

1. Connect to the Banner Cardholder Event Publisher administrative interface:

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

Note

This URL assumes that the default URL was used when deploying the Banner Cardholder Event Publisher. Any additions made to the URL during the deployment should be added to the URL as appropriate. ■

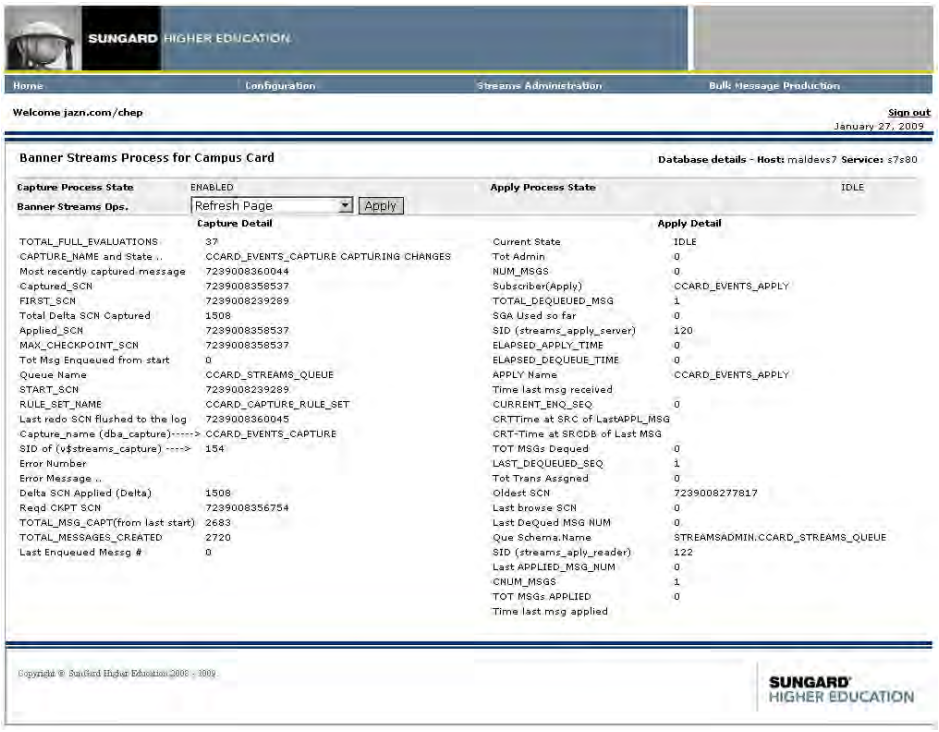
2. Enter the user name and password and click **Sign In**.

3. Select Streams Administration from the menu bar.

4. Verify that the desired process is running:

   4.1. Before stopping the capture process, **Capture Process State** must have the status *ENABLED*.

   4.2. Before stopping the apply process, **Apply Process State** must have the status *IDLE*, *DEQUEUE MESSAGES*, or *SCHEDULE MESSAGES*.



5. Select one of the following from the **Banner Streams Ops.** drop-down list:

   Stop capture
   Stop apply
   Stop capture and apply

6. Click **Apply**.

View status of Oracle Streams processes

Use the following steps to display information about the Oracle Streams processes for the Banner Cardholder Event Publisher. If there are problems with message production, this is the starting point for troubleshooting the problem.

1. Connect to the Banner Cardholder Event Publisher administrative interface:

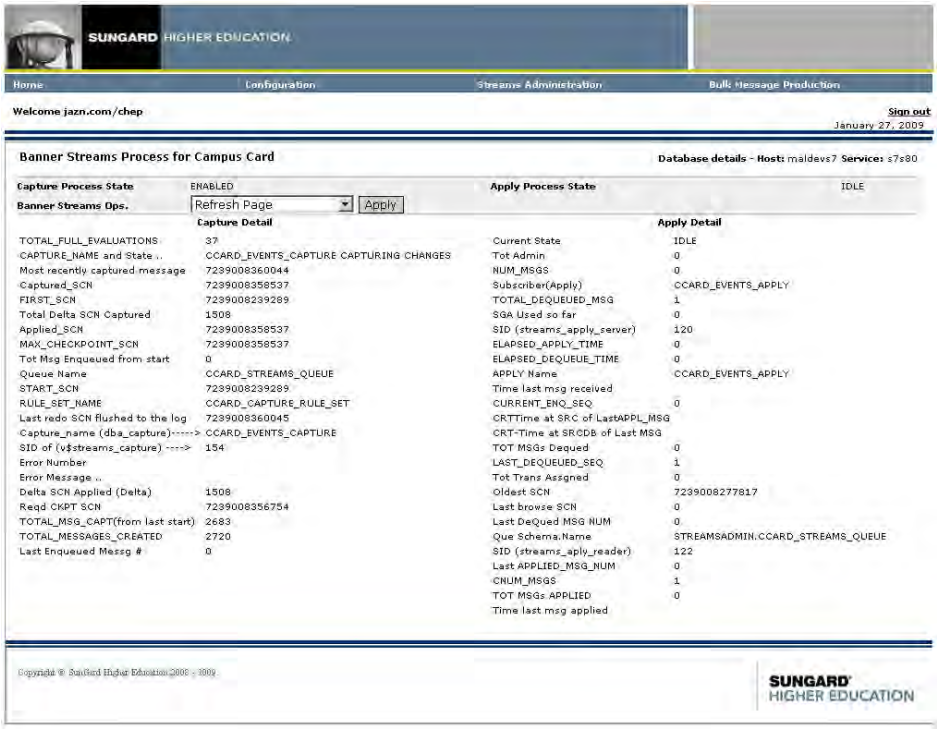
   ```
   http://<host>:<port>/cardholderEventPublisher
   ```

Note

This URL assumes that the default URL was used when deploying the Banner Cardholder Event Publisher. Any additions made to the URL during the deployment should be added to the URL as appropriate. ■

2. Enter the user name and password and click **Sign In**.

3. Select Streams Administration from the menu bar. The following page is displayed:



This page displays information about the capture and apply processes:

Process	Displayed Information
Capture process	This section displays details on the status of the capture process and queue. A status of *ENABLED* indicates the capture process is running.

Process	Displayed Information
Apply process	This section displays details on the status of the apply process and its consumption of a specified capture queue. A status of *IDLE*, *DEQUEUE MESSAGES*, or *SCHEDULE MESSAGES* indicates the apply process is running.

Load capture rules

Rules for capturing changes to `EligibleCardholder` information are maintained on the Rules Configuration Form (GUASADM). Whenever rules are added or changed on GUASADM, the rules must be loaded into the Oracle Streams capture and apply processes. Use the following steps to load new and updated capture rules.

1. Connect to the Banner Cardholder Event Publisher administrative interface:

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

Note

This URL assumes that the default URL was used when deploying the Banner Cardholder Event Publisher. Any additions made to the URL during the deployment should be added to the URL as appropriate.

2. Enter the user name and password and click **Sign In**.
3. Select Streams Administration from the menu bar.
4. Select *Load Capture Rules* in the **Banner Streams Ops.** field.
5. Click **Apply**.

These steps stop the capture and apply processes, reconfigure the Oracle Streams environment with new rules, and re-start the processes.

Produce bulk messages

Use the following steps to produce SyncEligibleCardholder messages for persons having one or more cardholder roles defined in Banner. You can publish messages in this manner

to initially and periodically synchronize data with a system that exposes a SyncEligibleCardholder end point.

1. Connect to the Banner Cardholder Event Publisher administrative interface:

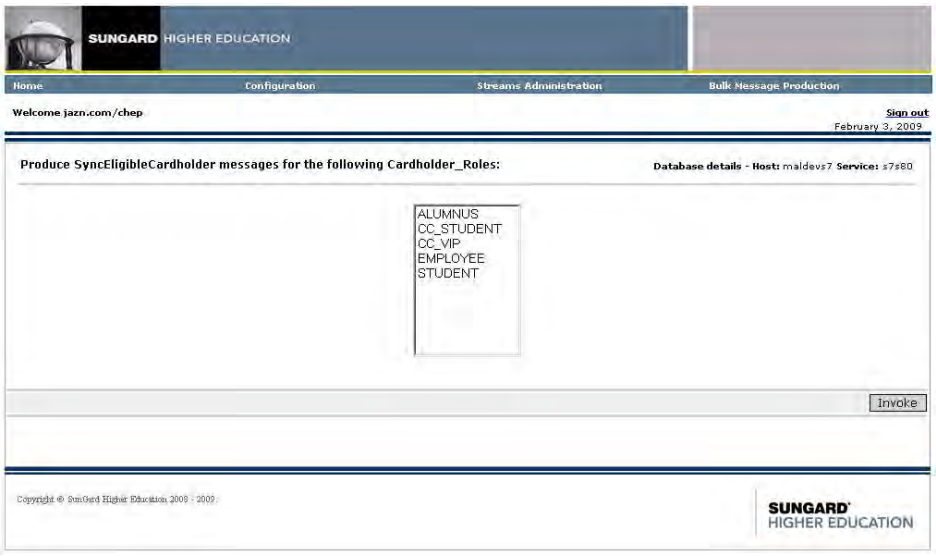
   ```
   http://<host>:<port>/cardholderEventPublisher
   ```

Note

This URL assumes that the default URL was used when deploying the Banner Cardholder Event Publisher. Any additions made to the URL during the deployment should be added to the URL as appropriate. ■

2. Enter the user name and password and click **Sign In**.

3. Select Bulk Message Production from the menu bar. The following page is displayed:



4. Select the cardholder roles for which you want to create bulk messages. (Use the Ctrl key to select multiple roles.)

5. Click **Invoke**.

Warning

This process might take a long time to complete. Finer-grained role definitions are recommended. ■

B Using Oracle Streams

When `EligibleCardholder` information in Banner® changes, the Oracle Streams technology captures, propagates, and applies the changes for use by the Banner Cardholder Event Publisher. This appendix provides a brief background of Oracle Streams and describes configuration, administration, monitoring, and troubleshooting for Oracle Streams when used with the Banner Cardholder Event Publisher.

Oracle messaging components

Oracle supports messaging via Advanced Queuing and Oracle Streams.

Advanced Queuing (AQ)

AQ is a message-oriented middleware hosted in the Oracle database. It provides the ability for a client session to place a message into a database queue table. This message is later "dequeued" by another session, typically immediately after committing, and its content is inspected. This message contains information for the other session to process.

Oracle Streams

Oracle Streams is built on top of the Oracle Advanced Queuing infrastructure, providing a framework for processing events. Events generated in applications or database triggers are captured and staged in a queue. These events can be consumed in various ways and applied automatically with a function or dequeued explicitly by an application.

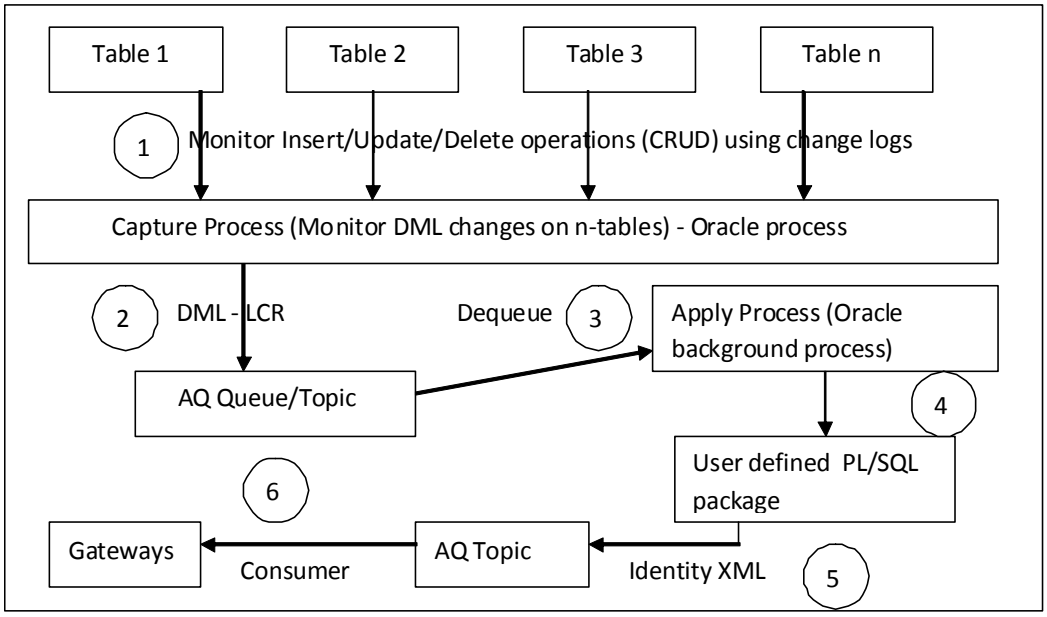
The basic objective of Oracle Streams is to capture, propagate, and apply database changes. The architecture consists of the following components:

- Capture process - This process mines the redo log files and the archived log files to extract database events such as DML and DDL changes that were done on database objects. Rules determine what events are captured. Information representing a captured event is formatted as a Logical Change Record (LCR) and placed in a queue associated with the capture process.
- Queue - Captured events are placed in queues (SGA or AQ queue or both) that act as staging areas. The queues store and manage LCRs and other types of events until subscriber or apply processes consume them.
- Apply process - This process consumes messages in the queue. An Oracle Streams apply process is flexible. It enables standard or custom apply of events. A custom apply can manipulate the data or perform other actions during apply. Applications with different data models can transform the messages while dequeuing or

enqueuing the messages to or from their own data model. These transformation mappings are defined as SQL expressions, which can involve PL/SQL functions, Java functions, or external C callouts.

Oracle Streams in Banner

The following figure shows how Banner table changes are captured and applied to external systems using Oracle Streams technology.



Capture process

Banner supports the capture process with capture definitions and supplemental logging.

Capture definitions

The Streams Rules Configuration Form (GUASADM) defines the rules for an Oracle Streams capture process. Each capture process monitors changes for a specified set of tables and selected columns from each table. You can also configure table-level rule conditions (`WHERE` clauses) that must evaluate to true to capture the transaction.

The `gp_streams_util` utility package, provided by Ellucian, uses the definitions on GUASADM to create Oracle Streams rule objects and to associate the rule objects with predefined background processes. These rule objects refer to the old and new column values.

Supplemental logging

The `gp_streams_util` utility package, provided by Ellucian, supports the supplemental logging of additional column information to the redo log. This package creates the `PRIMARY KEY` option and `UNIQUE` index logging for the tables defined on GUASADM to be monitored:

- `PRIMARY KEY` option - If a row containing a primary key is updated, all columns in the row's primary key are logged in the redo log file.
- `UNIQUE` index - If any column belonging to the composite unique index key is updated, all columns in the row's composite unique index key are logged in the redo log file.

If a row in a table defined on GUASADM is updated, "before images" of the columns specified on GUASADM are *always* logged in the redo log file, even if the specified columns are not changed.

Apply process

The apply process can be configured to invoke (call back) a PL/SQL package (handler) that performs custom processing of the captured transaction. The default callback handler for cardholder events is `p_handle_event`, which is contained in the `gp_cardholder` package delivered by Ellucian.

The `p_create_streams` API in the `gp_streams_util` utility package, delivered by Ellucian, creates and starts the apply process for a defined capture process. The apply process can be explicitly started or stopped using APIs in `gp_streams_util`.

Note

The apply process can also be started or stopped using the Banner Cardholder Event Publisher administrative interface. See Appendix A, "Administering Banner Cardholder Event Publisher" for details.

Errors encountered while consuming or processing the messages are written to the Oracle Streams metadata repository. Refer to "Troubleshoot Oracle Streams" on page B-16 for more information.

Configure the database and environment

Uses the following steps to configure the Oracle database and Banner environment for Oracle Streams:

- Step 1, "Verify Oracle version"
- Step 2, "Verify Oracle initialization parameters"

- Step 3, “Create tablespace for LogMiner objects”
- Step 4, “Verify Oracle Streams administrative user”
- Step 5, “Verify initial capture definitions”
- Step 6, “Update capture definitions”

Step 1 Verify Oracle version

Verify the Oracle database version:

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

Step 2 Verify Oracle initialization parameters

The database must be set up for Oracle Streams. Verify the recommended minimum settings for the `init.ora` parameters listed in the following table. If you have questions, open a support case with Ellucian.

Note

The database should have been set up during your upgrade to Banner General 7.5.1. ■

Parameter Name	Configured Value
`compatible`	Database version: *10.1.0* or *10.2.0*.
`global_names`	*true*
`job_queue_processes`	Maximum number of `dbms_job` processes. Increase the value by 4.

Parameter Name	Configured Value
streams_pool_size	Size of streams pool, in bytes. Streams pool contains captured events. This is part of SGA. Setting this parameter enables Oracle to assign a new pool of memory within the SGA that is dedicated to the use of Oracle Streams buffers. If the size of the pool is greater than 0, any SGA memory used by Oracle Streams is allocated from the pool. If this parameter is set to zero, the SGA memory used by Oracle Streams is allocated from the shared pool and can use up to 10% of shared pool only. Recommendations from Oracle: • 10MB for *each* capture process parallelism • 1MB for *each* apply process parallelism • 10MB or more for *each* queue that stages captured events Failure to create enough memory might cause contention. See the note later in this section.
timed_statistics	Setting that determines whether statistics related to time are collected. Set to *true* to collect elapsed time statistics in dynamic performance views related to Oracle Streams.
shared_pool_size	Increase by 20MB. See the note later in this section.
parallel_max_servers	Derived from the values of CPU_COUNT, PARALLEL_AUTOMATIC_TUNING, and PARALLEL_ADAPTIVE_MULTI_USER. Refer to Oracle documentation for details.
sga_max_size	If you run multiple capture processes on a single database, consider using this setting to increase the size of the System Global Area (SGA) for each instance.

Note

An Oracle Streams queue has a queue buffer that is used to stage captured events in shared memory. A queue buffer is memory associated

with a SYS.AnyData queue that contains only captured events. This is an optimization that Oracle performs. A queue buffer would normally overflow if there was not enough shared_pool_size available to hold captured events. Overflow events are stored in appropriate tables on disk. The limitation of using a shared pool is lifted by using a separate streams_pool_size initialization parameter. ■

Step 3 Create tablespace for LogMiner objects

Every capture process spawns a LogMiner process. By default, all LogMiner tables are created in SYSAUX tablespace. However, it is desirable to use a different tablespace for LogMiner objects. To use an alternate tablespace, create a new tablespace and move all LogMiner related objects to this tablespace. Oracle provides dbms_logmnr_d.set_tablespace packaged procedure to perform this task.

Step 4 Verify Oracle Streams administrative user

The streamsadmin user administers the Oracle Streams processes. Verify that this user exists. It is recommend that you use a separate tablespace or BANAQ for this user.

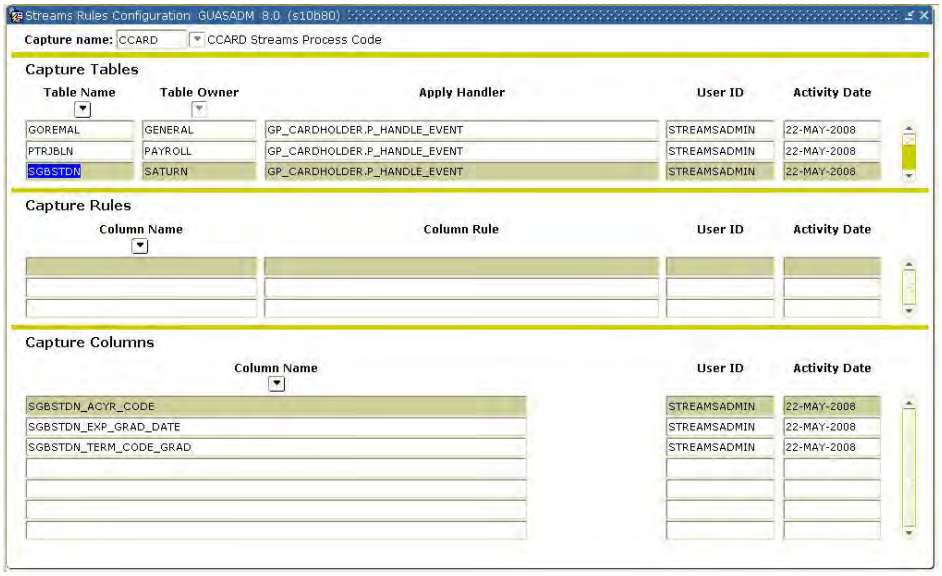
Step 5 Verify initial capture definitions

Initial rule definitions for capturing cardholder changes on the GTVSQRUI, GTVSQPRI, GORCTAB, GORCRUL, and GORCCOL tables were delivered with Banner:

- In a Banner 8.x environment, the initial definitions were delivered in the Banner General 8.0 seed data scripts.

Use the following steps to verify that the initial definitions are loaded in the database.

1. Log in to Banner as a user that has access rights to the Streams Rules Configuration Form (GUASADM).
2. Access GUASADM.
3. In the **Capture name** field, select *CCARD*.
4. Go to the Capture Tables block. The capture tables, rules, and columns are displayed:



5. Verify that capture table definitions exist for the following tables:

   GOREMAL
   PTRJBLN
   SGBSTDN
   SLRMASG
   SLRPASG
   SLRRASG
   SPBPERS
   SPRADDR
   SPRIDEN
   SPRTELE

6. If definitions do not exist for the tables, contact your database administrator to apply the following scripts.

Note

These scripts are available in the Banner General 7.5.1 patch p1-33tlib_gen70501 and in the Banner General 8.0 distribution. ■

```
gorrsqli_33tlib.sql
gorccoli_33tlib.sql
gobfpudi_33tlib.sql
gorctabi_33tlib.sql
gtvsqrui_33tlib.sql
gtvsqpri_33tlib.sql
gcardque_07050101.sql
```

Step 6 Update capture definitions

Additional rule definitions are available for capturing changes to data that affect the content of the `EligibleCardholder` structure. These definitions are loaded via SQL scripts in the following directory of the unzipped Banner Web Services 8.1.1 distribution:

```
\banner_web_services_8.1.1\cardholder_event_publisher\scripts
```

Use the following steps to update the capture definitions.

1. Execute the driver script from an SQL*Plus command prompt or comparable utility as follows, specifying the full path to the driver script as appropriate:

   ```
   SQL> @load_ccard_rules_080101.sql
   ```

   This step adds the definitions to the database.

2. Compile the enhanced capture definitions into the capture rules that are used by Oracle Streams. This can be done using either an SQL*Plus session or the Banner Cardholder Event Publisher administrative interface.

   2.1. To use SQL*Plus or comparable utility, log in as the `streamsadmin` user and execute the following command:

   ```
   exec gp_streams_util.p_configure_rules('CCARD');
   ```

   2.2. To use the Banner Cardholder Event Publisher administrative interface:

   2.2.1. Access the URL for the Banner Cardholder Event Publisher administrative interface.

   2.2.2. Select Streams Administration from the menu bar.

   2.2.3. Select *Load Capture Rules* in the **Banner Streams Ops.** field.

   2.2.4. Click **Apply**.

   Both methods stop the capture and apply processes, reconfigure the Oracle Streams environment with new rules, and re-start the processes.

Create and configure Oracle Streams processes

Use the following steps to create and configure Oracle Streams processes for the Banner Cardholder Event Publisher. Process code CCARD identifies the processes for the publisher.

1. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.

2. Run the following SQL query:

```
Select 1 FROM GSVCAST where
GSVCAST_CAPTURE_NAME='CCARD_EVENTS_CAPTURE';
```

   2.1. If the value *1* is returned, the capture process for CCARD is already created and configured. You can skip the rest of this step and start the capture process. Refer to "Start Oracle Streams processes" on page B-10.

   2.2. If the value *1* is not returned, the capture process for CCARD is not created and configured. Continue with the next step.

3. Execute `gp_streams_util.p_create_streams('CCARD');`

Note

This command might take some time to complete, depending on the number of users and data in the Banner database.

4. Execute `gp_streams_util.p_configure_rules('CCARD');`

The preceding steps accomplish the following for process code CCARD:

- Create buffered queues (type `SYS.AnyData`) and queue tables to manage events
- Create supplemental, primary key, and unique key log groups for the configured tables
- Configure the DML callback handler for the apply process
- Set the instantiation SCN for the tables in the apply process

Start Oracle Streams processes

Use the following steps to start the Oracle Streams processes using SQL*Plus.

Note

You can also start the Oracle Streams processes with the Banner Cardholder Event Publisher administrative interface. Refer to "Start Oracle Streams processes" on page A-3 for details.

Capture process

1. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
2. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

```
set serveroutput on
```

3. Run the following SQL query to determine the current status of the capture process:

```
Select GSVCAST_STATUS from GSVCAST where
GSVCAST_CAPTURE_NAME='CCARD_EVENTS_CAPTURE';
```

4. If the `GSVCAST_STATUS = ABORTED,` start the capture process as follows:

```
exec gp_streams_util.p_start_capture('CCARD');
```

Apply process

1. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
2. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

```
set serveroutput on
```

3. Run the following SQL query to determine the current status of the apply process:

```
Select 1 from GSVAPST where GSVAPST_APPLY_NAME =
'CCARD_EVENTS_APPLY';.
```

4. If *1* is not returned, start the apply process as follows:

```
exec gp_streams_util.p_start_apply('CCARD');
```

Stop Oracle Streams processes

Stopping the capture and apply processes may be necessary when upgrading middle-tier consumers of events published by the apply process or when changing the capture process configuration.

Use the following steps to stop the Oracle Streams processes using SQL*Plus.

Note

You can also stop the Oracle Streams processes with the Banner Cardholder Event Publisher administrative interface. Refer to "Stop Oracle Streams processes" on page A-4 for details.

Capture process

1. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
2. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

```
set serveroutput on
```

3. Stop the capture process as follows:

```
exec gp_streams_util.p_stop_capture('CCARD');
```

Apply process

1. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
2. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

```
set serveroutput on
```

3. Stop the apply process as follows:

```
exec gp_streams_util.p_stop_apply('CCARD');
```

View status of Oracle Streams processes

Banner provides four views that consolidate and report information about Oracle Streams-based processes in Banner:

View	Description
Streams Capture Process Status (GSVCAST)	Provides a quick report on the status of capture processes. When queried, a status (`GSVCAST_STATUS`) of *ENABLED* indicates a running capture process.
Streams Capture Process Details (GSVCADT)	Provides details on the status of capture processes and queues. Multiple rows are displayed for each capture process. One starts with the capture name, and another starts with its internal ID (SID). When queried, the row displaying the SID in the `GSVCADT_CAPTURE_DETAILS` column also includes a `CAPTURE_NAME` and state line. A state of *CAPTURING CHANGES* indicates a functioning capture process.
Streams Apply Process Status (GSVAPST)	Provides a quick report on the status of apply processes. When queried, a status (`GSVAPST_APPLY_STATE`) of *IDLE, DEQUEUE MESSAGES*, or *SCHEDULE MESSAGES* indicates a running apply process.
Streams Apply Process Details (GSVAPDT)	Provides details on the status of apply processes and their consumption of specified capture queues. Multiple rows are displayed for each apply process. A status of *IDLE, DEQUEUE MESSAGES*, or *SCHEDULE MESSAGES* indicates a functioning apply process.

The GSVCAST view provides the best indicator of the health of an Oracle Streams process. If problems are encountered with message production, querying GSVCAST should be one of the first steps to troubleshoot the process.

Modify Oracle Streams capture definitions

The Streams Rules Configuration Form (GUASADM) is used to modify definitions for a capture process. You can modify the tables that are monitored, the columns that are monitored, and the rules that are evaluated to determine whether a transaction is captured.

Add or delete tables to monitor

Use the following steps to add a table to the capture definition.

Note

Steps for deleting a table from the capture definition are similar. Remove rows from the Capture Tables block, save the changes, and load the new rules. ■

1. Access the Streams Rules Configuration Form (GUASADM).
2. In the **Capture name** field, select *CCARD*.
3. Go to the Capture Tables block.
4. Insert a new row and enter the following values:

   Table Name Table name from the list of values
   Apply Handler *GP_CARDHOLDER.P_HANDLE_EVENT*

5. Save the new row.
6. Go to the Capture Columns block.
7. Add the columns to be monitored by selecting the column names from the list of values.
8. Save the new rows.
9. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
10. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

    ```
    set serveroutput on
    ```

11. Execute the following utility packaged procedure:

    ```
    exec-gp_streams_util.p_configure_rules('CCARD');
    ```

    This procedure stops the capture and apply processes, reconfigures the Oracle Streams environment with the new rules, and re-starts the processes.

Add or delete columns to monitor

Use the following steps to monitor additional columns for a table that is already being monitored.

Note

Steps for removing columns from the capture definition are similar. Remove rows from the Capture Columns block, save the changes, and load the new rules.

1. Access the Streams Rules Configuration Form (GUASADM).
2. In the **Capture name** field, select *CCARD*.
3. Go to the Capture Tables block.
4. Select the table for which new columns are to be monitored.
5. Go to the Capture Columns block.
6. Insert new rows and add the columns to be monitored by selecting the column names from the list of values.
7. Save the new rows.
8. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
9. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

   ```
   set serveroutput on
   ```

10. Execute the following utility packaged procedure:

    ```
    exec-gp_streams_util.p_configure_rules('CCARD');
    ```

    This procedure stops the capture and apply processes, reconfigures the Oracle Streams environment with the new rules, and re-starts the processes.

Add or delete capture rules

Capture rules can be defined on the Streams Rules Configuration Form (GUASADM). These rules must evaluate to *true* for a database change to be captured and written to the capture process queue.

Example

To suppress the publication of messages for non-persons, add a capture rule in the Capture Rules block on GUASADM with the following values:

Column Name	`SPRIDEN_ENTITY_IND`
Column Rule	`IN ('P')`

Use the following steps to add a capture rule to a capture process.

Note

Steps for removing a capture rule from the capture definition are similar. Remove rows from the Capture Rules block, save the changes, and load the new rules. ■

1. Access the Streams Rules Configuration Form (GUASADM).
2. In the **Capture name** field, select *CCARD*.
3. Go to the Capture Tables block.
4. Select the table for which capture rules are to be added.
5. Go to the Capture Rules block.
6. Insert a new row and enter the following values:

Column Name	Column name from the list of values
Column Rule	Condition that must evaluate as *true*

7. Save the new row.
8. Open an SQL*Plus session and log in as the `streamsadmin` Oracle user.
9. Enable `DBMS_OUTPUT` and set the buffer size to unlimited by executing the following command:

```
set serveroutput on
```

10. Execute the following utility packaged procedure:

```
exec-gp_streams_util.p_configure_rules('CCARD');
```

This procedure stops the capture and apply processes, reconfigures the Oracle Streams environment with the new rules, and re-starts the processes.

Monitor Oracle Streams

Oracle provides two utilities for monitoring Oracle Streams in an Oracle database.

STRMMON

STRMMON is a monitoring tool that focuses on Oracle Streams. This tool provides database administrators with a quick overview of the Oracle Streams activity occurring within a database. The output comes in two formats:

- The default format reports the rate of activity occurring for Oracle Streams processes.
- The long format provides more detailed information. The reporting interval and number of iterations to display are configurable.

As of Oracle 10g Release 2, STRMMON is distributed in the demo directory of the database distribution code. It is also available from the Oracle customer support site, My Oracle Support, previously known as MetaLink (https://metalink.oracle.com).

Streams tool in Oracle Enterprise Manager Console

To help you configure, administer, and monitor Oracle Streams environments, Oracle provides a Streams tool in the Oracle Enterprise Manager Console. The Streams tool online help contains the primary documentation for this tool.

Troubleshoot Oracle Streams

Scripts in the Banner Web services download can be used to troubleshoot the capture and apply processes. The debug scripts are delivered in the following directory:

```
\banner_web_services_8.1\cardholder_event_publisher\scripts
```

Debug capture process

Use the following steps to resolve issues with a capture process. All steps must be executed as the `streamsadmin` Oracle user.

1. Query the Streams Capture Process Details View (GSVCADT) to determine if the capture process is running and its state. The state should be *CAPTURING CHANGES*.
2. Check for any error number and error messages in the results displayed on the console.

3. Check whether the capture process is *ENABLED*, *DISABLED*, or *ABORTED* by querying the `dba_capture` data dictionary view:

```
select status from dba_capture
where capture_name='CCARD_EVENTS_CAPTURE';
```

   A capture process must be enabled to capture changes. If the process is not enabled, check the alert/trace/log files for more info.

4. Perform DML activity on tables in GORCTAB to locate any issues with rules.

   4.1. Modify those columns listed in GORCCOL for the corresponding tables in GORCTAB to make sure that the rule sets associated with the capture are evaluated.

   4.2. Check the alert/trace/log files for any errors.

5. If the capture process passes the preceding tests and is not capturing changes, check the following:

   5.1. Redo scanning latency by running `redo_scanning_latency.sql`. Verify how far the capture message create time is behind the current system date. This helps explain the lag the system is experiencing.

   5.2. Run `rule_conditions.sql` to check rules and rule conditions. Rules might not be evaluating to *true*.

   5.3. Check rule syntax. Rules might have an invalid syntax. Report any invalid syntax as a bug to Ellucian.

   5.4. Run `rule_evaluation_details_for_capture.sql`. Check rule evaluation details to understand how many rules are discarded after prefiltering and how many full evaluations of rules are performed.

6. Check supplemental logging groups created by Ellucian streams utility packages:

   6.1. Execute `display_supp_log.sql` to list the log groups created for a Banner table. Every column in GORCRUL and GORCCOL that participates in evaluating a rule condition in a ruleset must have supplemental logging turned on so that old and new column values exist in the LCR. Rule conditions query the LCR to evaluate to *true* or *false*. If there are entries for columns in GORCCOL and GORCRUL, a supplemental log group called `CCARD_<`*`TABLE_NAME`*`>_LOG_GRP` must exist to instruct Oracle to log supplemental data on these columns.

   6.2. Execute `display_supp_log_columns.sql`, which takes the table name as input and displays the columns that are marked in that table for supplemental logging.

7. Add tracing to the process:

   7.1. Stop the capture process.

   7.2. Execute the following statement:

```
exec dbms_capture_adm.set_paramter
(capture_name=>'CCARD_EVENTS_CAPTURE',
parameter=>'trace_level',
value=>'127');
```

   7.3. Start the capture process.

Note

Trace level 127 turns on all tracing for Oracle Streams on the database. This setting generates a lot of output, so use it judiciously. To turn off tracing, change the parameter value from '127' to null. ■

8. If the trace files have the error message `Missing Streams MVDD information`, handle the problem as follows, depending on which situation causes the problem:

   8.1. If the object number specified in the trace file was not prepared in the database, use identification information in the trace file to determine the missing object. Execute the following as the `streamsadmin` Oracle user, providing the table name retrieved from the trace files:

```
DBMS_CAPTURE_ADM.PREPARE_TABLE_INSTANTIATION(
'<TABLE_NAME>');
```

   8.2. If information in the data dictionary table was lost by resetting the streams data dictionary tablespace with the `dbms_logminer_d.set_tablespace`, Oracle recommends running the `dbms_logminer_d.set_tablespace` procedure before any streams configuration takes place.

Debug apply process

Use the following step to resolve issues with the apply process. All steps must be executed as the `streamsadmin` Oracle user.

1. Run `show_apply_details.sql` to determine if the apply process is running and its state.

2. If the apply server has aborted, run `apply_process_abort_reason.sql` to determine why the process aborted.

3. If the apply process is in *DEQUEUE MESSAGES* state and is still not applying changes (dequeueing), run the following scripts to get information on apply errors:

`check_for_apply_errors.sql`	Displays apply errors on the console
`apply_stats_with_error_count.sql`	Shows statistics on messages dequeued, messages applied, and errors in applying

4. Determine if the apply process is encountering contention. The apply process has multiple apply servers. Apply servers apply DML changes. The parallelism apply process parameter determines the number of apply servers that can concurrently apply transactions. Check the alert/trace/log files for information regarding wait/contention.

5. Determine if the apply server is performing poorly for certain transactions by running `show_apply_transactions.sql` repeatedly to see if the apply server state, applied message number, and message sequence number change. If these values do not change for one or more apply servers, the apply server is taking too long. For each table to which the apply server applies changes, every key column must have an index.

6. Add tracing to the process:

   6.1. Stop the apply process.

   6.2. Execute the following statement:

```
exec dbms_apply_adm.set_paramter
(apply_name=>CCARD_EVENTS_APPLY',
parameter=>'trace_level',
value=>'127');
```

   6.3. Start the apply process.

Note

Trace level 127 turns on all tracing for streams on the database. This setting generates a lot of output, so use it judiciously. To turn off tracing, change the parameter value from '127' to null.