PeopleSoft Campus Solutions 9.0
Enrollment Web Services

USER’S GUIDE
Enrollment Web Services Users Guides for PeopleSoft Campus Solutions 9.0
Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.
## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>3</td>
</tr>
<tr>
<td>Change History</td>
<td>6</td>
</tr>
<tr>
<td>Audience</td>
<td>10</td>
</tr>
<tr>
<td>Overview</td>
<td>11</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>13</td>
</tr>
<tr>
<td>Common Data Types</td>
<td>15</td>
</tr>
<tr>
<td>CourseSearchResult</td>
<td>15</td>
</tr>
<tr>
<td>CourseOfferingDetailResult</td>
<td>15</td>
</tr>
<tr>
<td>ClassSearchResult</td>
<td>15</td>
</tr>
<tr>
<td>ClassSection</td>
<td>15</td>
</tr>
<tr>
<td>CourseShoppingCart</td>
<td>15</td>
</tr>
<tr>
<td>Student Enrollment Options</td>
<td>15</td>
</tr>
<tr>
<td>Enrollment Request Operations – Add/Drop/Swap/Normal Maintenance</td>
<td>16</td>
</tr>
<tr>
<td>StudyList</td>
<td>16</td>
</tr>
<tr>
<td>StudyList Deadlines</td>
<td>16</td>
</tr>
<tr>
<td><strong>Course Catalog Web Service Operations</strong></td>
<td>17</td>
</tr>
<tr>
<td>BROWSE Course Catalog</td>
<td>17</td>
</tr>
<tr>
<td>Description</td>
<td>17</td>
</tr>
<tr>
<td>Input Message: SSR_GET_COURSES_REQ</td>
<td>18</td>
</tr>
<tr>
<td>Output Message: SSR_GET_COURSES_RESP</td>
<td>19</td>
</tr>
<tr>
<td>RETRIEVE Course Offering Detail</td>
<td>21</td>
</tr>
<tr>
<td>Description</td>
<td>21</td>
</tr>
<tr>
<td>Input Message: SSR_GET_COURSE_OFFERING_REQ</td>
<td>22</td>
</tr>
<tr>
<td>Output Message: SSR_GET_COURSE_OFFERING_RESP</td>
<td>23</td>
</tr>
<tr>
<td><strong>Class Search Web Service Operations</strong></td>
<td>26</td>
</tr>
<tr>
<td>RETRIEVE Classes</td>
<td>26</td>
</tr>
<tr>
<td>Description</td>
<td>26</td>
</tr>
<tr>
<td>Input Message: SSR_GET_CLASSES_REQ</td>
<td>27</td>
</tr>
<tr>
<td>Output Message: SSR_GET_CLASSES_RESP</td>
<td>28</td>
</tr>
<tr>
<td>RETRIEVE Class Section Detail</td>
<td>30</td>
</tr>
<tr>
<td>Description</td>
<td>30</td>
</tr>
<tr>
<td>Input Message: SSR_GET_CLASS_SECTION_REQ</td>
<td>32</td>
</tr>
<tr>
<td>Output Message: SSR_GET_CLASS_SECTION_RESP</td>
<td>32</td>
</tr>
<tr>
<td>Student Enrollment Options</td>
<td>36</td>
</tr>
<tr>
<td>Description</td>
<td>36</td>
</tr>
<tr>
<td>Input Message: SSR_STDNT_ENRL_OPTIONS_REQ</td>
<td>40</td>
</tr>
<tr>
<td>Output Message: SSR_STDNT_ENRL_OPTIONS_RESP</td>
<td>41</td>
</tr>
<tr>
<td><strong>Enrollment Shopping Cart Web Service Operations</strong></td>
<td>44</td>
</tr>
<tr>
<td>Add an Item</td>
<td>44</td>
</tr>
<tr>
<td>Description</td>
<td>44</td>
</tr>
<tr>
<td>Input Message: SCC_SC_ADDITEM_REQ</td>
<td>45</td>
</tr>
<tr>
<td>Output Message: SCC_SC_ADDITEM_RESP</td>
<td>48</td>
</tr>
<tr>
<td>Clear Cart</td>
<td>53</td>
</tr>
<tr>
<td>Description</td>
<td>53</td>
</tr>
<tr>
<td>Input Message: SCC_SC_CLEARCART_REQ</td>
<td>53</td>
</tr>
<tr>
<td>Output Message: SCC_SC_CLEARCART_RESP</td>
<td>54</td>
</tr>
<tr>
<td>Get Cart</td>
<td>54</td>
</tr>
<tr>
<td>Description</td>
<td>54</td>
</tr>
<tr>
<td>Input Message: SCC_SC_GETCART_REQ</td>
<td>55</td>
</tr>
<tr>
<td>Output Message: SCC_SC_GETCART_RESP</td>
<td>55</td>
</tr>
<tr>
<td>Get Item</td>
<td>60</td>
</tr>
<tr>
<td>Description</td>
<td>60</td>
</tr>
<tr>
<td>Input Message: SCC_SC_GETITEM_REQ</td>
<td>60</td>
</tr>
<tr>
<td>Output Message: SCC_SC_GETITEM_RESP</td>
<td>61</td>
</tr>
<tr>
<td>Save Cart</td>
<td>65</td>
</tr>
</tbody>
</table>
Appendix

Configuration Hints and Tips

Entity Registry ................................................................. 128
Recommendations: ............................................................. 128
Enrollment Configuration ..................................................... 128
Recommendations: ............................................................. 128
List of Values ....................................................................... 132
Recommendations: ............................................................. 136
Enrollment Messages ........................................................... 136

Appendix .............................................................................. 138
Entity Details for EWS ......................................................... 138
Browse Course ................................................................... 138
Service Operation: SSR_GET_COURSES ................................. 138
Service Operation: SSR_GET_COURSE_OFFERING .................... 138
Class Search ....................................................................... 139
Service Operation: SSR_GET_CLASSES .................................... 139
Service Operation: SSR_GET_CLASS_SECTION .......................... 140
Study List ............................................................................ 140
Service Operation: SSR_GET_ENROLLMENT ........................................................................................................ 140
Service Operation: SSR_STUDYLIST_DEADLINES ............................................................................................ 141
Student Enrollment Options ....................................................................................................................................... 141
Service Operation: SSR_STUDENT_ENROLLMENT_OPTIONS .................................................................................. 141
Enrollment Shopping Cart ......................................................................................................................................... 141
  SCC_SC_ADDITEM ................................................................................................................................................ 141
Service Operation: SCC_SC_GETITEM ..................................................................................................................... 142
Service Operation: SCC_SC_GETCART ...................................................................................................................... 143
Service Operation: SCC_SC_CLEARCART .................................................................................................................. 143
Service Operation: SCC_SC_REMOVEITEM .............................................................................................................. 144
Service Operation: SCC_SC_VALIDATEITEM .......................................................................................................... 145
Service Operation: SCC_SC_CHECKOUT ................................................................................................................ 145
Service Operation: SCC_SC_SAVECART .................................................................................................................. 146
Enrollment Operations .............................................................................................................................................. 147
Service Operation: SSR_ENR_VALIDATE ............................................................................................................... 147
Service Operation: SSR_ADD_ENROLLMENT .......................................................................................................... 148
Service Operation: SSR_DROP_ENROLLMENT ...................................................................................................... 148
Service Operation: SSR_SWAP_ENROLLMENT ...................................................................................................... 148
Service Operation: SSR_EDIT_ENROLLMENT ........................................................................................................ 148
### Change History

<table>
<thead>
<tr>
<th>Release</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Features(AF) July 2011</td>
<td>First Release of this guide for AF July 2011 : Enrollment Web Services (EWS)</td>
</tr>
<tr>
<td>Additional Features(AF) Oct 2011</td>
<td>Second Release of this guide for AF Oct 2011: Enrollment Web Services (EWS). Updated the guide to include the following new services &amp; service operations delivered as part of Oct Bundle#23: <strong>Enrollment Shopping Cart</strong> - New Service SCC_SHOPPING_CART to perform Enrollment Shopping Cart functionalities. -Included section Enrollment Shopping Cart Web Service Operations in the guide to capture details of the new service. -Updated section Configuration Hints and tips to include: -- &quot;Enrollment Shopping Cart” under Enrollment configuration sub section. -- &quot;Enrollment Shopping Cart Service” under List Of Values sub section. -- &quot;Enrollment Messages” sub section. -Updated the Appendix section to include section Enrollment Shopping Cart to capture entity registry details. <strong>Enrollment Request</strong> - Modified the existing service SSR_ENROLLMENT to include new service operations that perform the functionalities for Add, Drop, Edit &amp; Swap Enrollment. -Updated section Enrollment Web Service Operations in the guide to capture details of the new service operations. -Updated section Configuration Hints and tips to include: -- &quot;Enrollment Requests” under Enrollment configuration sub section. -- &quot;Enrollment Messages” sub section. -- Student Enrollment Options/Enrollment Request Service’ under List Of Values sub section. -Updated the Appendix section to include section Enrollment Operations to capture entity registry details. <strong>Student Enrollment Options</strong> - Modified the existing service SSR_CLASS to include new service operation to generate student enrollment options for the selected class. -Updated section Class Search Web Service Operations in the guide to capture details of the new service operation. -Updated section Configuration Hints and tips to include: -- &quot;Student Enrollment Options” under Enrollment configuration sub section. -- &quot;Student Enrollment Options/Enrollment Request Service” under List Of Values sub section. -Updated the Appendix section to include section Student Enrollment Options to capture entity registry details.</td>
</tr>
</tbody>
</table>
Updated the guide to include the following changes made to existing services delivered in July Bundle#22:

**Class Search, Retrieve Classes** - Modified the existing service operation SSR_CLASS/SSR_GET_CLASSES to include Course Id & Course Offer Nbr as additional search parameters.
-Updated the Class Search Web Service Operations section, Service Name: SSR_CLASS/SSR_GET_CLASSES to capture details of the change under the Description heading.
- Corrected request and response diagrams of SSR_GET_CLASSES operation.

**Class Search, Retrieve Class Section Detail** - Modified the existing service operation SSR_CLAS/SSR_GET_CLASS_SECTION to capture details as to Related, Auto enrollable, Enrollable sections under the node <CLASS_COMPONENT_SECTIONS>.
-Updated the Class Search Web Service Operations section, Service Name: SSR_CLASS/SSR_GET_CLASS_SECTION to capture details of the change under the Description heading.
- Corrected request and response diagrams of SSR_GET_CLASS_SECTION operation.

**Studylist** - Updated the response xml for SSR_ENROLLMENT/SSR_GET_ENROLLMENT in the guide to reflect the new attributes SSR_SWAP_ALLOW/SSR_EDIT_ALLOW/SSR_DROP_ALLOW.
- Corrected response diagrams of SSR_GET_ENROLLMENT & SSR_STUDYLIST_DEADLINES operations.

**Course Search** - Corrected request and response diagrams of SSR_GET_COURSES & SSR_GET_COURSE_OFFERING operations
Additional Features(AF) Jan 2012

Third Release of this guide for AF Jan 2012: Enrollment Web Services (EWS).

Updated the guide to include the changes made to following services delivered in Oct Bundle#23:

**Get Class Section**
- Corrected the schema diagram of SSR_GET_CLASS_SECTION_RESP under SSR_CLASS/SSR_GET_CLASS_SECTION in the guide as part of Bug 13417533.

**StudyList**
- Modified the Studylist service SSR_ENROLLMENT/SSR_GET_ENROLLMENT to include new fields in the response as part of Bug 13491753.
- Under section GET StudyList Details, updated the Description, Response Entity Diagram, Response message structure
- Updated the Entity hierarchy Display screen shot under Appendix, Entity Details for EWS for SSR_GET_Enrollment service operation.

**List Of Values**
- Updated the section Configuration Hints and tips, List of Values sub section for Get Studylist Service, Student Enrollment Options/Enrollment Request Service, Enrollment Shopping Cart Service for Institution, Acad Career & Term LOV as part of Bug 13445953.

**Shopping Cart**
- Modified the response messages of Shopping Cart service to include Class Section details for each ITEM in the response as part of Bug 13491742.
- Updated the following subsections under Enrollment Shopping Cart Web Service Operations:
  --*Add an Item*: Updated the Description, Request & Response Entity diagram, Response xml
  --*Get Cart*: Updated the Description, Response Entity diagram, Response xml
  --*Get Item*: Updated the Description, Response Entity diagram, Response xml
  --*Save Cart*: Updated the Description, Request & Response Entity diagram, Response xml
  --*Remove Item*: Updated the Description, Request & Response Entity diagram, Response xml
  --*Validate Shopping Cart*: Updated the Request entity diagram.
  --*Checkout Shopping Cart*: Updated the Request entity diagram.
- Updated the Request Entity Diagram of SSR_ENTOLLMENT/SSR_ENR_VALIDATE under Enrollment Web Service Operation
- Updated the Entity structure details under Appendix, Entity Details for EWS for the following service operations:
  SSR_GET_CLASS_SECTION
  SCC_SC_ADDITEM
  SCC_SC_GETITEM
  SCC_SC_GETCART
  SCC_SC_CLEARCART
  SCC_SC_REMOVEITEM
  SCC_SC_VALIDATEITEM
  SCC_SC_CHECKOUT
  SCC_SC_SAVECART
  SSR_ENR_VALIDATE
Additional Features (AF) Apr 2012

Fourth Release of this guide for AF Apr 2012: Enrollment Web Services (EWS).
Updated the guide to include the following changes as part of BUG # 13616873:

Service: SCC_SHOPPING_CART
Updated the response XMLs of the following service operations to include LOV Descriptions for INSTITUTION, STRM, RQMNT_DESIGNTN, ENRL_REQ_DET1_STAT and also populated valid values for RQMNT_DESIGNTN & ENRL_REQ_DET1_STAT.
- SCC_SC_ADDITEM
- SCC_SC_GETCART
- SCC_SC_GETITEM
- SCC_SC_SAVECART
- SCC_SC_REMOVEITEM

This updated guide is available on My Oracle Support (ID 137139.1) as part of the documentation delivered for the releases indicated in the table.
Audience

This document is designed to provide important information to customers who want to access and use the following Enrollment Web Services (EWS)

- Course Catalog Web Service
- Class Search Web Service
- Shopping Cart Web Service
- Enrollment Web Service

Technical developers, consultants and, implementers should use this document to learn about the technologies used by these web services and to determine the scope of data that the web services can exchange between a self service Enrollment user interface (that the academic institution may build) and the PeopleSoft Campus Solutions system. Business and functional analysts should use this document to understand how the web services function.

This document is intended for users who are familiar with:

- A tool or technology to build and deploy the Enrollment user interface.
- PeopleCode.
- Web services concepts, mainly XML, SOAP, and WSDL.
- PeopleSoft Integration Broker.
- Campus Solutions Student Records and Enrollment functionality.
- Campus Community ‘Entity Registry’ functionality.

The following topics are covered in this document:

- The Common Data Types chapter provides information about the data that is exchanged between the Enrollment user interface (UI) and the PeopleSoft system.
- The Course Catalog Web Service Operations chapter describes the web service operations for Course Catalog and retrieving Course Offering detail.
- The Class Search Web Service Operations chapter describes the web service operations for Class Search, retrieving Class Section detail and student enrollment options.
- The Shopping Cart Web Service Operations chapter describes the web service operations for managing student shopping cart, such as, add to cart, clear cart, validate cart etc.
- The Enrollment Web Service Operations chapter describes the web service operations for add, drop, edit and swap enrollment request. It also includes StudyList operations (Get Enrollment & Get Enrollment Deadlines) for StudyList related activities and Enrollment Validate operation for validating course shopping cart items prior to checkout.

Each section that describes a web service operation contains:

- An overview of the service operation.
- Steps that a web service operation performs to execute input parameters and give an output.
- Conditions that will make the service return an error message.
Overview

Enrollment Web Service (EWS)

Institutions have different needs, requirements and preferences for interacting with their students during the enrollment process. It is essential that the interaction between the student and institution is convenient, simple to use, and powerful for both the institution and the student. The User interface (UI) through which a student interacts with the institution can vary. For many campus audiences the interface of choice could be a range of current generation handheld devices (smartphone, tablets, blackberry, etc). However, options like conventional web browser based access (such as PeopleSoft Self Service Student Enrollment) need to also be supported. In order to facilitate this flexibility a suite of web services has been delivered to support the Self Service Enrollment process regardless of the user interface technology deployed at an institution.

The EWS feature, initially delivered with Additional Features July 2011, offers access to the Campus Solutions Student Records and Enrollment functionality through service operations. The feature offers the following service operations:

- Retrieve (or Browse) Course Catalog
- Retrieve Course Offering Detail
- Retrieve Classes
- Retrieve Class Section Detail
- Retrieve StudyList
- Retrieve Enrollment Deadline Information

In Additional Features Oct 2011, the following service operations have been added:

- Display Student enrollment options
- Add class to shopping cart
- Clear shopping cart
- Remove class from shopping cart
- Validate Shopping Cart
- Retrieve Shopping Cart
- Retrieve an item from shopping cart
- Checkout from shopping cart
- Save Shopping Cart
- Validate Enrollment Request
- Add Enrollment
- Drop Enrollment
- Edit Enrollment
- Swap Enrollment

Any user interface that is web-service enabled and SOAP compliant can access these web services.

Suggested additional references:

- PeopleSoft Campus Community Fundamentals 9.0 PeopleBook, Setting Up Entity Registry. The Setting Up Entity Registry chapter was posted to My Oracle Support as part of the Campus Community documentation updates for Bundle 22. Please review Bundle 22 documentation updates and any subsequent bundle documentation updates on My Oracle Support to check if there have been updates to this chapter.

- PeopleSoft Campus Community Fundamentals 9.0 PeopleBook, Setting Up List of Values. The Setting Up List of Values chapter was posted to My Oracle Support as part of the Campus Community documentation updates for Bundle 22. Please review Bundle 22 documentation updates and any subsequent bundle documentation updates on My Oracle Support to check if there have been updates to this chapter.

- PeopleSoft Campus Community Fundamentals 9.0 PeopleBook, Using the Shopping Cart Framework. The Using the Shopping Cart Framework chapter was posted to My Oracle Support as part of the Campus Community documentation updates for Bundle 23. Please review Bundle 23 documentation updates and any subsequent bundle documentation updates on My Oracle Support to check if there have been any updates to this chapter.
• Enrollment Web Service Developers Guide.pdf on My Oracle Support: ID 1371391.1

• Enrollment Web Services October 2011 Documentation Package.pdf (PeopleBook documentation) on My Oracle Support for Additional Features Oct 2011

• Enrollment Web Services Documentation Package.pdf (PeopleBook documentation) on My Oracle Support for Additional Features July 2011

The Document ID for the My Oracle Support Campus Solutions 9.0 Documentation Home Page is 751540.1.
Prerequisites

EWS can be used by end user devices (such as a browser or a tablet), enterprise applications or any other third party software that can interface through standards based means.

EWS can be used by:

- Oracle presentation technologies (such as PORTAL, Application Development Framework [or ADF], and PeopleSoft components and pages).
- Oracle middleware (such as Enterprise Service Bus and BPEL).
- Oracle Applications (such as Enterprise, EnterpriseOne, and E-Business Suite).
- Third party presentation technologies.
- Third party middleware and applications.

EWS is constructed upon Oracle PeopleSoft PeopleTools 8.50. Oracle’s PeopleSoft PeopleTools 8.50 and higher versions support the tools and technologies listed in the following table.

<table>
<thead>
<tr>
<th>Technology/Product</th>
<th>Version/Specification</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Xalan</td>
<td>1.9. C++ version.</td>
<td>Apache Xalan is an XSLT processor for transforming XML documents into HTML, text, or any other document type.</td>
</tr>
<tr>
<td>Apache Xerces</td>
<td>2.6.0. C++ version</td>
<td>Apache Xerces is an XML parser.</td>
</tr>
<tr>
<td>AS2</td>
<td>RFC 3335</td>
<td>Based on software product IP*Works! EDI V6.</td>
</tr>
<tr>
<td>FTP/FTPS</td>
<td>5.1</td>
<td>NA</td>
</tr>
<tr>
<td>HTTP</td>
<td>1.1</td>
<td>NA</td>
</tr>
<tr>
<td>HTTPS</td>
<td>SSL 3.0</td>
<td>NA</td>
</tr>
<tr>
<td>iWay SOAPswitch</td>
<td>5.5.3</td>
<td>NA</td>
</tr>
<tr>
<td>JMS</td>
<td>1.0.2</td>
<td>NA</td>
</tr>
<tr>
<td>Oracle XSL Mapper</td>
<td>10.1.2.02</td>
<td>NA</td>
</tr>
<tr>
<td>SOAP</td>
<td>1.1 and 1.2</td>
<td>All SOAP output from PeopleSoft Integration Broker is version 1.1. PeopleSoft Integration Broker understands versions 1.1 and 1.2. EWS uses Schema for the SOAP/1.1 envelope, and SOAP/1.2 envelope found at <a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>.</td>
</tr>
<tr>
<td>UDDI</td>
<td>2.0</td>
<td>NA</td>
</tr>
<tr>
<td>WSDL</td>
<td>1.1</td>
<td>EWS uses the schema found at <a href="http://schemas.xmlsoap.org/wsdl">http://schemas.xmlsoap.org/wsdl</a>.</td>
</tr>
<tr>
<td>WS-Addressing</td>
<td>1.1</td>
<td>Currently the WS-Addressing document is a formal submission to W3C, the current submission is using schema under <a href="http://schemas.xmlsoap.org/ws/2004/08/addressing/">http://schemas.xmlsoap.org/ws/2004/08/addressing/</a>. However, the integration gateway uses an older version, <a href="http://schemas.xmlsoap.org/ws/2003/03/addressing/">http://schemas.xmlsoap.org/ws/2003/03/addressing/</a>.</td>
</tr>
<tr>
<td>Technology/Product</td>
<td>Version/Specification</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
<td>----------</td>
</tr>
<tr>
<td>WS-I Basic Profile</td>
<td>1.0</td>
<td>The WS-I Basic Profile 1.0 specification mandates support for SOAP 1.1, WSDL 1.1, UDDI 2.0, HTTP 1.1, SSL 3.0 (or HTTPS), XML 1.0 (2nd Edition), and XML Schema (Part 1 and 2).</td>
</tr>
<tr>
<td>WS-Interoperability</td>
<td>1.2.1</td>
<td>NA</td>
</tr>
<tr>
<td>XML</td>
<td>1.0 (Second edition.)</td>
<td>NA</td>
</tr>
<tr>
<td>XML schema</td>
<td>Parts 1 and 2</td>
<td>NA</td>
</tr>
<tr>
<td>XSLT</td>
<td>1.0</td>
<td>Use of Apache Xerces 2.6.0 and Apache Xalan 1.9, XSLT 1.0 is supported.</td>
</tr>
<tr>
<td>XQuery, XPath</td>
<td>1.0</td>
<td>Used in PeopleSoft Integration Broker transform programs. Use of Apache Xerces 2.6.0 and Apache Xalan 1.9, XQuery and XPath 1.0 is supported.</td>
</tr>
</tbody>
</table>
Common Data Types

This chapter lists the data types that EWS uses to exchange data between the Campus Solutions system and the student.

CourseSearchResult

The CourseSearchResult object represents an institution’s core Subject and Course information. This data is derived from the complete set of Subjects and Course catalog related data available in the Campus Solutions Student Records and Enrollment core data model.

CourseOfferingDetailResult

The CourseOfferingDetailResult object represents the core information associated with a Course. The Course object is derived from the complete set of Course related data available in the Campus Solutions Student Records and Enrollment core data model.

ClassSearchResult

The ClassSearchResult object represents the core information associated with a Class. The Class object is derived from the complete set of Class related data available in the Campus Solutions Student Records and Enrollment core data model.

ClassSection

The ClassSection object represents the core information associated with a Class Section. The ClassSection object is derived from the complete set of Class related data available in the Campus Solutions Student Records and Enrollment core data model.

CourseShoppingCart

The CourseShoppingCart object represents the Courses that are part of the course shopping cart for a student in Campus Solutions. This data is derived from the complete set of Class related data and student enrollment options available in the Campus Solutions Student Records and Enrollment core data model.

Student Enrollment Options

The StudentEnrollmentOptions object represents the enrollment options for the selected class and for the associated enrollment request action for a student. This data is derived from the complete set of Student Enrollment Details and Class related data available in the Campus Solutions Student Records and Enrollment core data model.
Enrollment Request Operations – Add/Drop/Swap/Normal Maintenance

The EnrollmentRequestDetail object represents a request for enrollment operations Add, Drop, Swap, Normal maintenance for a student in Campus Solutions. This object will be created using the complete set of Course, Class related data and student enrollment options available in the Campus Solutions Student Records and Enrollment core data modal.

StudyList

The StudyList is the core object containing the complete view of a student’s enrollment. The StudyList object is derived from the Student Enrollment details within the Campus Solutions Student Records and Enrollment core data model.

StudyList Deadlines

The StudyList Deadlines is the core object containing the complete details of class deadlines. This data is derived from the complete set of deadlines related data available in Campus Solutions Student Records and Enrollment core data model.
Course Catalog Web Service Operations

Service Name: **SSR_COURSE**

Course Catalog Web Service offers functionality in the following areas:

- **Course Catalog Search:**
  
  This operation enables a student to:
  
  - Browse the Campus Solutions Course Catalog.
  - View detail on individual Course Offerings.

**BROWSE Course Catalog**

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_COURSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_GET_COURSES: Browse Course Catalog</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to initiate searches against the Student Records Course Catalog using a number of criteria. Matching course results returned from this search can be further processed using other services such as Class Search.</td>
</tr>
<tr>
<td>Description:</td>
<td>The Browse Course Catalog service can be used to retrieve matches in one of two different modes, Subject Header or Course Detail. When used in Subject Header mode the returned matches are summarized and only Subject header information is returned. This mode is useful when presenting results to a student in a hierarchical summary display. It also has the performance benefit of not returning complete course detail for all matching courses on the initial search. When used in Course Detail mode the returned matches contain further course detail. Each match will include a full course summary. If further detailed information is required for a particular course then that information can be obtained via the Retrieve Course Offering Detail (SSR_GET_COURSE_OFFERING) service detailed in this document. The values for SSR_CRS_SRCH_MODE parameter are ‘H’ and ‘D’ for subject header and course details modes respectively.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
</tbody>
</table>
### Input Message: SSR_GET_COURSES_REQ

The following diagram shows the input message structure that the SSR_GET_COURSES_REQ service operation receives from an Enrollment UI:

![Diagram of SSR_GET_COURSES_REQ Message Structure](image)

The following input parameters are required for the Enrollment application to pass to the service operation:

- Institution (INSTITUTION)*
- Search Character (SSR_ALPHANUM)
- Subject (SUBJECT)
- Search Mode (SSR_CRS_SRCH_MODE)
- As of Date (ASOFDATE)

The parameters that are marked with an asterisk (*) are mandatory.

The following is an example of the SSR_GET_COURSES_REQ message that the SSR_GET_COURSES service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_GET_COURSES_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <COURSE_SEARCH_REQUEST>
    <INSTITUTION>PSUNV</INSTITUTION>
    <SSR_ALPHANUM></SSR_ALPHANUM>
  </COURSE_SEARCH_REQUEST>
</SSR_GET_COURSES_REQ>
```
Output Message: SSR_GET_COURSES_RESP

When the Integration Broker receives the SSR_GET_COURSES_REQ message, it responds with the SSR_GET_COURSES_RESP message.

The following diagram shows the output message structure that the SSR_GET_COURSES service operation passes to the UI:

![Diagram of SSR_GET_COURSES_RESP Message Structure]

The following is an example of the SSR_GET_COURSES_RESP message that the SSR_GET_COURSES service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_GET_COURSES_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <COURSE_SEARCH_RESULT>
    <SSR_CRS_GEN_MSG/>
    <SSR_CRS_SRCH_COUNT>5</SSR_CRS_SRCH_COUNT>
    <SUBJECTS>
      <SUBJECT>
        <INSTITUTION>PSUNV</INSTITUTION>
        <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
        <SUBJECT>STATS</SUBJECT>
        <SUBJECT_LOVDescr>Statistics</SUBJECT_LOVDescr>
        <SUBJECT_CRS_COUNT>5</SUBJECT_CRS_COUNT>
        <COURSE_SUMMARIES>
          <COURSE_SUMMARY>
            <CRSE_ID>001151</CRSE_ID>
            <CRSE_ID_LOVDescr>Introductory Statistics</CRSE_ID_LOVDescr>
          </COURSE_SUMMARY>
        </COURSE_SUMMARIES>
      </SUBJECT>
    </SUBJECTS>
  </COURSE_SEARCH_RESULT>
</SSR_GET_COURSES_RESP>
```
Fault: SSR_FAULT_RESP

Refer to the fault message example in the GET Enrollment section.

RETRIEVE Course Offering Detail

Description

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_COURSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_GET_COURSE_OFFERING: Retrieve the detail for a Course Offering</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to obtain detailed information associated with a Course. Course offering details returned from this service operation can be further processed using other services such as Class Search.</td>
</tr>
<tr>
<td>Description:</td>
<td>The SSR_GET_COURSE_OFFERING service operation can be used to get the Course offering details in the context of the Browse Course service operation (SSR_GET_COURSES). The Course ID and Offer Number from the Browse Course service operation is used for the request parameters for getting the Course Offering details for a selected Subject and Course combination. The Operation returns the Course Details along with the Terms Offered for the particular Course. The user can then continue the Class Search for one of the Terms Offered.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
</tbody>
</table>
Processing: This service operation performs the following steps:

- Verify required input parameters exist.
- If no AS OF DATE is supplied then the current date is used as a default.
- Perform prompt edit validations on the input parameters.
- Use the CourseOfferingDetailResult entity and the input parameters to get the course offering details.
- Prepare the response message with the course offering details.

Output: Course Details along with the Terms Offered for a particular Course.

Error Conditions: The following condition results in an error from the service:

- The supplied input parameters are invalid.

Input Message: SSR_GET_COURSE_OFFERING_REQ

The following diagram shows the input message structure that the SSR_GET_COURSE_OFFERING service operation receives from the UI:

![Figure 7: SSR_GET_COURSE_OFFERING_REQ Message Structure](image)

The service operation accepts the following parameters:

- Institution (INSTITUTION)*
- Course ID (CRSE_ID)*
- Course Offer Number (CRSE_OFFER_NBR)*
- Course Topic ID (CRS_TOPIC_ID)
- As of Date (ASOFDATE)

The parameters marked with * are mandatory parameters.

The following is an example of the SSR_GET_COURSE_OFFERING_REQ message that the SSR_GET_COURSE_OFFERING service operation receives from an Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_GET_COURSE_OFFERING_REQ
xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <COURSE_OFFERING_REQUEST>
    <INSTITUTION>*</INSTITUTION>
    <CRSE_ID>*</CRSE_ID>
    <CRSE_OFFER_NBR>*</CRSE_OFFER_NBR>
    <CRS_TOPIC_ID></CRS_TOPIC_ID>
    <ASOFDATE></ASOFDATE>
  </COURSE_OFFERING_REQUEST>
</SSR_GET_COURSE_OFFERING_REQ>
```
Output Message: SSR_GET_COURSE_OFFERING_RESP

When the Integration Broker receives the SSR_GET_COURSE_OFFERING_REQ message, it responds with the SSR_GET_COURSE_OFFERING_RESP message.

The following diagram shows the output message structure that the SSR_GET_COURSE_OFFERING service operation passes to the UI:

![Diagram of SSR_GET_COURSE_OFFERING_RESP Message Structure]

The following is an example of the SSR_GET_COURSE_OFFERING_RESP message that the SSR_GET_COURSE_OFFERING service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_GET_COURSE_OFFERING_RESP
xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <COURSE_OFFERING_RESULT>
    <SSR_TERMS_OFFERED_COUNT>4</SSR_TERMS_OFFERED_COUNT>
    <SSR_CRS_GEN_MSG>Course Offering Details</SSR_CRS_GEN_MSG>
    <COURSE_OFFERING>
      <CRSE_ID>003080</CRSE_ID>
      <CRSE_OFFER_NBR>2</CRSE_OFFER_NBR>
      <INSTITUTION>PSUNV</INSTITUTION>
      <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
    </COURSE_OFFERING>
  </COURSE_OFFERING_RESULT>
</SSR_GET_COURSE_OFFERING_RESP>
```
<table>
<thead>
<tr>
<th>Subject</th>
<th>Sociology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog NBR</td>
<td>601</td>
</tr>
<tr>
<td>Rqrmnt Group</td>
<td>008023</td>
</tr>
<tr>
<td>Descrlong</td>
<td></td>
</tr>
<tr>
<td>Course Title Long</td>
<td>Human Information Processing</td>
</tr>
<tr>
<td>Units Minimum</td>
<td>3</td>
</tr>
<tr>
<td>Units Maximum</td>
<td>3</td>
</tr>
<tr>
<td>Grading Basis</td>
<td>Graded</td>
</tr>
<tr>
<td>Consent</td>
<td>No Special Consent Required</td>
</tr>
<tr>
<td>Ssr_drop Consent</td>
<td>No Special Consent Required</td>
</tr>
<tr>
<td>Rqrrmnt Designtn</td>
<td></td>
</tr>
<tr>
<td>Acad Career</td>
<td>Graduate</td>
</tr>
<tr>
<td>Acad Group</td>
<td>LBART</td>
</tr>
<tr>
<td>Acad Org</td>
<td>LIBARTS</td>
</tr>
<tr>
<td>Campus</td>
<td>Main Hacienda Campus</td>
</tr>
<tr>
<td>Units Range</td>
<td>3.00</td>
</tr>
<tr>
<td>Course Components</td>
<td></td>
</tr>
<tr>
<td>Course Id</td>
<td>003080</td>
</tr>
<tr>
<td>Crse Id Lovdescr</td>
<td>Humn Info Process</td>
</tr>
<tr>
<td>Ssr Component</td>
<td>Lec</td>
</tr>
<tr>
<td>Optional Section</td>
<td>N</td>
</tr>
<tr>
<td>Course Attributes</td>
<td></td>
</tr>
<tr>
<td>Terms Offered</td>
<td></td>
</tr>
<tr>
<td>Crse Id</td>
<td>003080</td>
</tr>
<tr>
<td>Crse Offer Nbr</td>
<td>2</td>
</tr>
<tr>
<td>Institution</td>
<td>PeopleSoft University</td>
</tr>
<tr>
<td>Strm</td>
<td>0690</td>
</tr>
<tr>
<td>Strm Lovdescr</td>
<td>2012 Fall</td>
</tr>
<tr>
<td>Acad Career</td>
<td>Graduate</td>
</tr>
<tr>
<td>Acad Career Lovdescr</td>
<td></td>
</tr>
<tr>
<td>Term Offered</td>
<td></td>
</tr>
<tr>
<td>Crse Id</td>
<td>003080</td>
</tr>
<tr>
<td>Crse Offer Nbr</td>
<td>2</td>
</tr>
<tr>
<td>Institution</td>
<td>PeopleSoft University</td>
</tr>
<tr>
<td>Strm</td>
<td>0640</td>
</tr>
<tr>
<td>Strm Lovdescr</td>
<td>2010 Spring</td>
</tr>
<tr>
<td>Acad Career</td>
<td>Graduate</td>
</tr>
<tr>
<td>Acad Career Lovdescr</td>
<td></td>
</tr>
<tr>
<td>Term Offered</td>
<td></td>
</tr>
</tbody>
</table>
Fault: SSR_FAULT_RESP

Refer to the fault message example in the GET Enrollment section.
Class Search Web Service Operations

Service Name: **SSR_CLASS**

The Class Search Web Service which offers functionality in the following areas:

- Class Search

This operation enables a student to:

- Search for classes matching certain criteria.
- View detail on individual Class Sections.
- Allows UI to generate student enrollment options

### RETRIEVE Classes

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td><strong>SSR_GET_CLASSES</strong>: Retrieve Classes</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to initiate a search of classes and view the search results.</td>
</tr>
</tbody>
</table>
| Description: | The Class search service can be used to search for the classes based on the input search criteria specified by the user. The service operation also takes in additional search criteria as parameters as in the current SS. Along with the class search result, the response also includes the following:
  - SSR_COURSE_COUNT- Count of courses matching the search criteria.
  - SSR_CLASS_COUNT - Count of class section matching the search criteria.  
  - SSR_ERR_LMT_EXCEED - Set to Y if search result exceeds error limit specified in the installation table.
  - SSR_WRN_LMT_EXCEED - Set to Y if search result exceeds warning limit specified in the installation table and the obey_warning_limit in the request payload is set to YES. |
| Users:     | Student |
| Processing | This service operation performs the following steps:
  - Validates input parameters.
  - Initiates the class search processing by invoking the appropriate classes under the SSR_CLASS application class.
  - If search parameters include Course id, then all other criteria are ignored by the search(though specified) and the search will be performed on INSTITUTION, STRM, COURSE ID, COURSE OFFER NBR(if present).
  - When the Search Criteria does not include course id, search will be performed as usual.
  - Prepare response message containing the class search results. |
| Output:    | A list of zero or more matches is returned to the user. |
Error Conditions:
The following conditions result in a service error:

- The supplied input parameters are invalid.
- The service operation obeys the Term values setup as defined in the Term Values table. So if the Term is set not to be available for self service users, then an error message is shown.
- The service operation obeys the Error/Warning limit as set in Student Record Installation set up. If the SSR_CLASS_COUNT is not within the error limit, then an error message is shown in the response.
- If it is not within the warning limit and the Obey Warning Limit in the request is Y, then a warning message is displayed in the response along with the class search results. It is up to the UI to either show or not show the class search results in case of warning.

Input Message: SSR_GET_CLASSES_REQ

The following diagram shows the SSR_GET_CLASSES_REQ message structure:

![Diagram of SSR_GET_CLASSES_REQ message structure]

The service operation accepts the following parameters:

- INSTITUTION
- TERM
- SUBJECT
- CATALOG NBR
- COURSE CAREER
- CLASS NBR
- COURSE ID
- COURSE OFFERING NBR
- SESSION
- CLASS SECTION
- OPEN ONLY
- OPEN ENTRY/EXIT CLASSES ONLY
- COURSE TITLE KEYWORD
- COURSE COMPONENT
- INSTRUCTION MODE
- CAMPUS
- LOCATION
- MEETING START AND END TIME
- DAY OF WEEK
- INSTRUCTOR LAST NAME
- MAXIMUM AND MINIMUM UNITS
• OBEY WARNING LIMITS
The following is an example of the SSR_GET_CLASSES_REQ message that the SSR_GET_CLASSES service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_GET_CLASSES_REQ xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <CLASS_SEARCH_REQUEST>
    <INSTITUTION>PSUNV</INSTITUTION>
    <STRM>0665</STRM>
    <CLASS_NBR>1014</CLASS_NBR>
    <CRSE_ID></CRSE_ID>
    <CRSE_OFFER_NBR>0</CRSE_OFFER_NBR>
    <SESSION_CODE>12W</SESSION_CODE>
    <CLASS_SECTION>SF</CLASS_SECTION>
    <SUBJECT>MUSIC</SUBJECT>
    <CATALOG_NBR></CATALOG_NBR>
    <SSR_EXACT_MATCH1>E</SSR_EXACT_MATCH1>
    <SSR_OPEN_ONLY>Y</SSR_OPEN_ONLY>
    <OEE_IND>N</OEE_IND>
    <DESCR></DESCR>
    <ACAD_CAREER>UGRD</ACAD_CAREER>
    <SSR_COMPONENT></SSR_COMPONENT>
    <INSTRUCTION_MODE></INSTRUCTION_MODE>
    <CAMPUS></CAMPUS>
    <LOCATION></LOCATION>
    <MEETING_TIME_START></MEETING_TIME_START>
    <MEETING_TIME_END></MEETING_TIME_END>
    <MON>N</MON>
    <TUES>N</TUES>
    <WED>N</WED>
    <THURS>N</THURS>
    <FRI>N</FRI>
    <SAT>N</SAT>
    <SUN>N</SUN>
    <INCLUDE_CLASS_DAYS>I</INCLUDE_CLASS_DAYS>
    <LAST_NAME></LAST_NAME>
    <SSR_EXACT_MATCH2></SSR_EXACT_MATCH2>
    <UNITS_MINIMUM>0</UNITS_MINIMUM>
    <UNITS_MAXIMUM>0</UNITS_MAXIMUM>
    <SSR_START_TIME_OPR></SSR_START_TIME_OPR>
    <SSR_END_TIME_OPR></SSR_END_TIME_OPR>
    <SSR_UNITS_MIN_OPR></SSR_UNITS_MIN_OPR>
    <SSR_UNITS_MAX_OPR></SSR_UNITS_MAX_OPR>
    <OBEY_WARNING_LIMIT>N</OBEY_WARNING_LIMIT>
  </CLASS_SEARCH_REQUEST>
</SSR_GET_CLASSES_REQ>
```

Output Message: SSR_GET_CLASSES_RESP

When the Integration Broker receives the SSR_GET_CLASSES_REQ message, it responds with the SSR_GET_CLASSES_RESP message.

The following diagram shows the SSR_GET_CLASSES_RESP message structure:
Figure 10: SSR_GET_CLASSES_RESP Message Structure

The following is an example of the SSR_GET_CLASSES_RESP message that the SSR_GET_CLASSES service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_GET_CLASSES_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <SEARCH_RESULT>
    <ERROR_WARN_TEXT></ERROR_WARN_TEXT>
    <SSR_COURSE_COUNT>1</SSR_COURSE_COUNT>
    <SSR_ERR_LMT_EXCEED></SSR_ERR_LMT_EXCEED>
    <SSR_WRN_LMT_EXCEED></SSR_WRN_LMT_EXCEED>
    <SSR_CLASS_COUNT>1</SSR_CLASS_COUNT>
    <SUBJECTS>
      <SUBJECT>
        <CRSE_ID>003146</CRSE_ID>
        <CRSE_ID_LOVDescr>Indiv Instruction</CRSE_ID_LOVDescr>
        <SUBJECT>MUSIC</SUBJECT>
        <SUBJECT_LOVDescr>Music</SUBJECT_LOVDescr>
        <CATALOG_NBR>250</CATALOG_NBR>
        <INSTITUTION>PSUNV</INSTITUTION>
        <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
        <ACAD_CAREER>UGRD</ACAD_CAREER>
        <ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
        <COURSE_TITLE_LONG>Individual Instruction II</COURSE_TITLE_LONG>
        <CLASSES_SUMMARY>
          <CLASS_SUMMARY>
            <CRSE_ID>003146</CRSE_ID>
            <CRSE_ID_LOVDescr>Indiv Instruction</CRSE_ID_LOVDescr>
            <SUBJECT>MUSIC</SUBJECT>
            <SUBJECT_LOVDescr>Music</SUBJECT_LOVDescr>
            <CATALOG_NBR>250</CATALOG_NBR>
          </CLASS_SUMMARY>
        </CLASSES_SUMMARY>
      </SUBJECT>
    </SUBJECTS>
  </SEARCH_RESULT>
</SSR_GET_CLASSES_RESP>
```
<CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
<STRM>0665</STRM>
<STRM_LOVDescr>2011 Summer</STRM_LOVDescr>
<SESSION_CODE>12W</SESSION_CODE>
<SESSION_CODE_LOVDescr>Twelve Week</SESSION_CODE_LOVDescr>
<Class_SECTION>SF</CLASS_SECTION>
<Class_NBR>1014</CLASS_NBR>
<SCHEDULE_PRINT>Y</SCHEDULE_PRINT>
<SCHEDULE_PRINT_LOVDescr>Yes</SCHEDULE_PRINT_LOVDescr>
<COMBINED_SECTION></COMBINED_SECTION>
<COMBINED_SECTION_LOVDescr></COMBINED_SECTION_LOVDescr>
<Class_TOPIC></CLASS_TOPIC>
<SSR_CLASSNAME_LONG></SSR_CLASSNAME_LONG>
<STATUS>O</STATUS>
<STATUS_LOVDescr>Open</STATUS_LOVDescr>
<SSR_COMPONENT>LAB</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Laboratory</SSR_COMPONENT_LOVDescr>
<Class_MEETING_PATTERNS>
<Class_MEETING_PATTERN>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
<SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
<SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
<SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
</CLASS_MEETING_PATTERN>
</CLASSES_MEETING_PATTERNS>
</CLASS_SUMMARY>
</CLASSES_SUMMARY>
</SUBJECT>
</SUBJECTS>
</SEARCH_RESULT>
</SSR_GET_CLASSES_RESP>

**RETRIEVE Class Section Detail**

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_GET_CLASS_SECTION: Retrieve the detail for a Class Section</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to search for a particular class section and see the search results. The response includes Class Section, Class Combined Section, Class Component, Class Meeting Pattern, Class Text Books, and Enrollment Information.</td>
</tr>
</tbody>
</table>
The SSR_GET_CLASS_SECTION service operation can be used to perform a search for a particular class section. After the Get Class Search operation is performed, the UI can invoke the Get Class Section operation to retrieve details of the class section. Input to the service operation comes from the output values of the Get Class operation.

The operation executes on either of the two class search modes-Class Section or Class Number. If class search mode is Class Section, then the parameters will include:

- Institution
- Course ID
- Offer Nbr
- Term
- Session
- Class Section

If the class search mode is Class Number then the request will include Term and Class Nbr.

This service operation also provides the following details under node <CLASS_COMPONENT_SECTIONS>:

- Related class sections
- Auto Enroll Sections
- Enroll sections when a non-enroll section is passed in the request.
- Node <CLASS_MEETING_PATTERNS> which is the child of <CLASS_COMPONENT_SECTIONS> provides Class Meeting Pattern details for each component section.

Some use cases and behavior for data shown in <CLASS_COMPONENT_SECTIONS> node in response xml:

- User enters a non-enrollment class section - service op should return a list of enrollment sections in the <CLASS_COMPONENT_SECTIONS> Node CLASS_TYPE for all sections returned should be ‘E’.
- User enters an enrollment class section with 2 auto-enrollment section and no related classes - service op should return the schedule info for the auto-enrollment sections in the <CLASS_COMPONENT_SECTIONS> node.
- User enters an enrollment class section with no auto-enrollment section and no related classes - no data in the <CLASS_COMPONENT_SECTIONS> node.
- User enters an enrollment class section with no auto-enrollment section and related classes - service op should return the schedule info for related class sections in the <CLASS_COMPONENT_SECTIONS> node. If there are multiple components, the ui can just filter the payload based on ssr_component.
- User enters an enrollment class section with auto-enrollment section and related classes - service op should return the schedule info for related class sections and auto-enrollment sections. UI can just use the SSR_AUTO_ENROLL_SECT flag and SSR_COMPONENT to distinguish data.

<table>
<thead>
<tr>
<th>Users:</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Verify required input parameters exists</td>
</tr>
<tr>
<td></td>
<td>• Use the appropriate classes in SSR_CLASS application package to retrieve the class section details based on the input search criteria.</td>
</tr>
<tr>
<td></td>
<td>• Prepares response message comprising of class section details.</td>
</tr>
<tr>
<td>Output:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Response message containing the class section details is sent back to the user.</td>
</tr>
<tr>
<td>Error Conditions:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The following condition results in an error from the service:</td>
</tr>
<tr>
<td></td>
<td>• The supplied input parameters are invalid</td>
</tr>
</tbody>
</table>
**Input Message: SSR_GET_CLASS_SECTION_REQ**

The following diagram shows the SSR_GET_CLASS_SECTION_REQ message structure:

![Diagram of SSR_GET_CLASS_SECTION_REQ message structure]

The following are the input parameters for the UI to pass to the service operation:

- Course ID
- Offer Nbr
- Term
- Session
- Class Nbr
- Class Section
- Class Search Mode

The following is an example of the SSR_GET_CLASS_SECTION_REQ message that the SSR_GET_CLASS_SECTION service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_GET_CLASS_SECTION_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
    <CLASS_SECTION_REQUEST>
        <CRSE_ID>003746</CRSE_ID>
        <CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
        <STRM>0670</STRM>
        <SESSION_CODE>1</SESSION_CODE>
        <CLASS_SECTION>2</CLASS_SECTION>
        <SSR_SEARCH_MODE>S</SSR_SEARCH_MODE>
        <SSR_INCLUD_CLS_SET></SSR_INCLUD_CLS_SET>
        <CLASS_NBR>1718</CLASS_NBR>
        <ACAD_CAREER></ACAD_CAREER>
        <ACAD_PROG></ACAD_PROG>
    </CLASS_SECTION_REQUEST>
</SSR_GET_CLASS_SECTION_REQ>
```

**Output Message: SSR_GET_CLASS_SECTION_RESP**

When the Integration Broker receives the SSR_GET_CLASS_SECTION_REQ message, it responds with the SSR_GET_CLASS_SECTION_RESP message.

The following diagram shows the SSR_GET_CLASS_SECTION_RESP message structure:
The following diagram shows the structure of SCC_ENTITY_CLASS_SECTION_RES.V1.xsd

Figure 12: SSR_GET_CLASS_SECTION_RESP Message Structure

The following is an example of the SSR_GET_CLASS_SECTION_RESP message that the SSR_GET_CLASS SECTION service
operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_GET_CLASS_SECTION_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <CLASS_SECTION_RESULT>
    <SSR_CRS_SRCH_COUNT>0</SSR_CRS_SRCH_COUNT>
    <SSR_CRS_GEN_MSG/>
  </CLASS_SECTION_RESULT>
  <CLASS_SECTIONS>
    <SSR_CLASS_SECTION>
      <ACAD_CAREER>UGRD</ACAD_CAREER>
      <ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
      <ASSOCIATED_CLASS>2</ASSOCIATED_CLASS>
      <AVAILABLE_SEATS>100</AVAILABLE_SEATS>
      <CAMPUS>MAIN</CAMPUS>
      <CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
      <CATALOG_NBR>201</CATALOG_NBR>
      <CLASS_NBR>1718</CLASS_NBR>
      <CLASS_SECTION>2</CLASS_SECTION>
      <CLASS_TYPE>E</CLASS_TYPE>
      <CLASS_TYPE_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
      <CONSENT>N</CONSENT>
      <CONSENT_LOVDescr>No Special Consent Required</CONSENT_LOVDescr>
      <CRS_TOPIC_ID>0</CRS_TOPIC_ID>
      <CRS_TOPIC_DESCR/>
      <CRSE_ID>003746</CRSE_ID>
      <CRSE_ID_LOVDescr>Organic Chem I</CRSE_ID_LOVDescr>
      <CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
      <DESCR200>CHEM 201 - 2 Organic Chemistry I</DESCR200>
      <DESCRLONG/></DESCRLONG>
      <EFFDT>2011-08-30</EFFDT>
      <END_DT>2011-12-12</END_DT>
      <ENRL_CAP>100</ENRL_CAP>
      <ENRL_STAT>O</ENRL_STAT>
      <ENRL_TOT>0</ENRL_TOT>
      <GRADING_BASIS>GRD</GRADING_BASIS>
      <GRADING_BASIS_LOVDescr>Graded</GRADING_BASIS_LOVDescr>
      <INSTITUTION>PSUNV</INSTITUTION>
      <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
      <INSTRUCTION_MODE>P</INSTRUCTION_MODE>
      <INSTRUCTION_MODE_LOVDescr>In Person</INSTRUCTION_MODE_LOVDescr>
      <LOCATION>PSCSHCDA</LOCATION>
      <LOCATION_DESCR>Hacienda</LOCATION_DESCR>
      <MIN_ENRL>0</MIN_ENRL>
      <PRINT_TOPIC>N</PRINT_TOPIC>
      <PRINT_TOPIC_LOVDescr>No</PRINT_TOPIC_LOVDescr>
      <SESSION_CODE>1</SESSION_CODE>
      <SESSION_CODE_LOVDescr>Regular Academic Session</SESSION_CODE_LOVDescr>
      <SSR_CLASSNOTE_LONG/>
      <SSR_COMPONENT>LEC</SSR_COMPONENT>
      <SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
      <SSR_CRSE_ATTR_LONG/>
      <SSR_CRSE_TYPOFF_CD>FALL, SPR</SSR_CRSE_TYPOFF_CD>
      <SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
      <SSR_DROP_CONSENT>N</SSR_DROP_CONSENT>
      <SSR_DROP_CONSENT_LOVDescr>No Special Consent Required</SSR_DROP_CONSENT_LOVDescr>
      <SSR_GBL_NOTE_LONG/>
      <SSR_GBL_NOTE_LOVDescr></SSR_GBL_NOTE_LOVDescr>
      <SSR_REQUISITE_LONG/>
      <START_DT>2011-08-30</START_DT>
    </SSR_CLASS_SECTION>
  </CLASS_SECTIONS>
</SSR_GET_CLASS_SECTION_RESP>
```
Student Enrollment Options

Description

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_STUDENT_ENROLLMENT_OPTIONS: Generate Student enrollment options</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the user interface to generate the enrollment options for the selected class and for the associated enrollment request action.</td>
</tr>
</tbody>
</table>
Description: Student Enrollment Options service provides the details to enable the UI to generate the enrollment options for the selected class and associated enrollment request action for a specified emplid. Enrollment request action can take the following values – Enroll, Swap, Normal Maintenance and Drop. In case of Swap, ‘change to Class Nbr’ value also needs to be provided. The UI would need to invoke this operation first to obtain Enrollment Request Defaults. It would then pass the response values as request for invoking subsequent operations like – Shopping Cart Add/Save or Enrollment operations Add/Drop/Swap/Edit.

The response consists of:
- **ENROLLMENT_REQUEST_DEFAULT**
  - The parent node provides the enrollment request defaults
  - Enroll and Swap, values such as Unit Taken, Course Count, Grading Basis Enroll, Requirement Designation, Requirement Designation Option, Start Dt, Wait List Ok etc. are defaulted from setup in the response message.
  - Edit or Drop, all the values under this node is populated from Student Enrollment Table.
- **CLASS_ENROLLMENT_OPTIONS**
  - enabling/disabling a field (for example, Related Class)
  - prompt records which the UI can use through LOV (list of values) service to obtain appropriate values.(for example, prompt for Grading basis)

Following explains each of the elements that are part of CLASS_ENROLLMENT_OPTIONS:
- **SSR_REL_CLS_ENBL** - Enable/Disable Indicator for RELATED_CLASS_NBR1 and RELATED_CLASS_NBR2. UI should prompt the student for their selection if set to True. Can make use of LOV Service – SCC_LOV_WRK/RELATE_CLASS_NBR_1/2
- **SSR_WT_LTST_OK_ENBL** - Enable/Disable Indicator for WAIT_LIST_OK flag. UI should prompt the student for their selection if set to True.
- **SSR_PRM_NBR_ENBL** - Enable/Disable Indicator for CLASS_PRMSN_NBR flag. UI should prompt the student for their selection if set to True.
- **SSR_VAR_UNIT_INCR, UNITS_MAXIMUM and UNITS_MINIMUM** - Support numeric incrementer widget on UNT_TAKEN
  - UNT_TAKEN_ENBL - Enable/Disable Indicator for UNT_TAKEN. UI should open unt_taken for input if set to True.
  - **SSR_START_DT_ENBL** - Enable/Disable Indicator for START_DT. UI should prompt the student for their selection if set to True.
  - **SSR_START_DT_MIN and SSR_START_DT_MAX** - Support date prompt on start_Dt
  - **SSR_INSTR_ID_PRMPT** - This is needed to support LOV/prompting on INSTRUCTOR_ID. It will contain the field name on the LOV record SCC_LOV_WRK which will be used by the LOV service.
  - **SSR_ACAD_PRG_ENBL** - Enable/Disable Indicator for ACAD_PROG. UI should prompt the student for their selection if set to True.
  - **RQMT_DESIGTN_OPT_ENBL** - Enable/Disable Indicator for RQMNT_DESIGNTN_OPT. UI should prompt the student for their selection if set to True.
  - **ELECT_GRADE_BASIS** - Enable/Disable Indicator for GRADING_BASIS_ENRL. UI should open grading_basis_enrl for input if set to Y.
  - **High level keys for grading basis prompt/lov - CRSE_GRP_SCHEMA,GB_MAP_RULE,GRADING_BASIS,GRADING_BASIS_CNV,GRADING_SCHEME,GRADING_BASIS_INV_SETID**
  - **SSR_GRD_BASE_PRMPT** - This is needed to support LOV/prompting. It will contain the field name on the LOV record SCC_LOV_WRK which will be used by the LOV service.

Users: Student

Processing: This service operation performs the following steps:
- Validates input parameters.
- Initiates the student enrollment options processing by invoking the appropriate classes under the SSR_CLASS application package.
- Prepare response message containing Enrollment Request Defaults and student enrollment options.

Output: Enrollment request default and enrollment options are sent in the response.
Error Conditions:

Validations performed by the service operation:
The service operation performs validation depending on whether it is invoked to obtain enrollment options for Enrollment Request service or for Shopping Cart service. The flag SSR_ENROLLMENT_REQ on the request indicates this. The UI will set this value to ‘Y’ if it is invoked to obtain enrollment options for Enrollment Request service else it indicates that it is invoked for Shopping Cart service.

General Validations – common to all enrollment request actions-Enroll/Drop/Swap/Edit:

The following validations are performed irrespective of whether student enrollment options service operation is invoked prior to Enrollment Request Service or Shopping Cart service.

- Student is term activated and eligible to enroll.
- Student is not withdrawn or cancelled.
- Student has an active program for the term.
- Error when a non-enroll class section is passed in the request.
- Error in case of invalid input.
- Following validations are performed only when student enrollment options is invoked prior to enrollment request service.

- Students can submit enrollment transactions only when their UserId is associated to the Self Serv Permission List - Setup SACR > Security > Secure Student Administration > Permission List > Self-Serv Enrollment Perm List. (See: PeopleSoft Campus Solutions 9.0 Application Fundamentals PeopleBook, Securing Student Records, Setting Up Enrollment Security for Self-Service Enrollment)
- If online appointment checking is enforced, check if student has enrollment appointment for the selected class session.

Enrollment Request Action Specific Validations:

- Enroll
  - Class Nbr validation
  - Check Career Rules
  - OEE Edits
  - Check Last date to Enroll
  - Check duplicate Enrollment

Note: The above validations are performed irrespective of whether the student enrollment options are invoked prior to Enrollment Request service or Shopping Cart Service.

- Drop
  Classes that may be dropped are classes that have not been graded, have an enrollment status of enrolled and the enrollment status reason is not WDRW or DROP, or the enrollment status of waitlisted.

- Edit
  Classes that may be edited are classes that have not been graded, have an enrollment status of enrolled and the enrollment status reason is not WDRW or DROP, or the enrollment status of waitlisted.

Class will not be Edited unless at least one of following attributes are updatable:

- Requirement Designation Option
- Unit Taken
- Related Classes
- Instructor ID
- Grading Basis
- **Swap**
  Classes that may be swapped are classes that have not been graded or dropped.

  If the Self Service Enrollment Option for Swap is ‘Drop if Enrolled’ and the student is not enrolled in the new class nbr or either related classes, then the enrollment action will be E (Enroll) and the class nbr will be the new class and the original class number will be in the drop class if enrolled field. (See: *PeopleSoft Campus Self Service 9.0 PeopleBook, Setting Up Student Records Self-Service*)

  If the Self Service Enrollment Option for Swap is “Swap” then the enrollment action will be S (Swap) and the class nbr goes in the “original class nbr” field and the new class number goes in the “class number to change to” field.

  Note: The above validations under Drop/Edit/Swap are performed only when the student enrollment options is invoked prior to Enrollment Request service.
**Input Message: SSR_STDNT_ENRL_OPTIONS_REQ**

The following diagram shows the SSR_STDNT_ENRL_OPTIONS_REQ message structure:

```
<schema>
  targetNamespace "http://xmlns.oracle.com/Enterprise/HCM/services"
</schema>
```

Figure 9: SSR_STDNT_ENRL_OPTIONS_REQ Message Structure

The service operation accepts the following parameters:

- *USERID
- *Password
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number
- Change to Class Nbr
- *Enrollment Request Action
- Enrollment Request (Y/N)

Parameters marked with * are mandatory.

- User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
- EmplID - the ID of the Student.
- Institution – the Institution of the student.
- Academic Career – the Career of the student.
- Term – the Term of the student.
- Class Number - the Class Number for which the enrollment options is requested.
- In case of Swap, ‘change to Class Nbr’ value also needs to be provided.
- Enrollment request action can take any of the following values –
  - Enroll
  - Swap
  - Normal Maintenance
  - Drop
- Enrollment Request – Indicates whether student enrollment options service operation is invoked for Enrollment Request Service or for shopping Cart service. This will need to be set by the UI. A value of ‘Y’ indicates that it is invoked for Enrollment Request Service. A blank value indicates that it is invoked to obtain enrollment options for Shopping Cart Add Item service.

The following is an example of the SSR_STDNT_ENRL_OPTIONS_REQ message that SSR_STDNT_ENRL_OPTIONS service operation receives from an Enrollment application:

```
<?xml version="1.0"?>
<SSR_STDNT_ENRL_OPTIONS_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <STUDENT_ENRL_OPTION_REQUEST>
    <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
    <EMPLID>SR12030</EMPLID>
  </STUDENT_ENRL_OPTION_REQUEST>
</SSR_STDNT_ENRL_OPTIONS_REQ>
```
Output Message: SSR_STDNT_ENRL_OPTIONS_RESP

When the Integration Broker receives the: SSR_STDNT_ENRL_OPTIONS_REQ message, it responds with the SSR_STDNT_ENRL_OPTIONS_RESP message.

The following diagram shows the SSR_STDNT_ENRL_OPTIONS_RESP message structure:

![SSR_STDNT_ENRL_OPTIONS_RESP Message Structure Diagram]

Figure 30: SSR_STDNT_ENRL_OPTIONS_RESP Message Structure

The following is an example of SSR_STDNT_ENRL_OPTIONS_RESP message that SSR_STDNT_ENRL_OPTIONS service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_STDNT_ENRL_OPTIONS_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
    <STUDENT_ENROLLMENT_OPTIONS>
        <ENROLLMENT_REQUEST_DEFAULT>
            <ENRL_REQUEST_ID>0</ENRL_REQUEST_ID>
            <ENRL_REQ_DETL_SEQ>0</ENRL_REQ_DETL_SEQ>
            <EMPLID>SR12030</EMPLID>
            <ACAD_CAREER>UGRD</ACAD_CAREER>
            <INSTITUTION>PSUNV</INSTITUTION>
            <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
            <STRM>0665</STRM>
            <CLASS_NBR>1014</CLASS_NBR>
            <ENRL_REQ_ACTION>E</ENRL_REQ_ACTION>
            <ENRL_REQ_ACTION_LOVDescr>Enroll</ENRL_REQ_ACTION_LOVDescr>
            <ENRL_ACTION_REASON></ENRL_ACTION_REASON>
            <ENRL_ACTION_REASON_LOVDescr></ENRL_ACTION_REASON_LOVDescr>
            <ENRL_ACTION_DT></ENRL_ACTION_DT>
            <UNT_TAKEN>3</UNT_TAKEN>
        </ENROLLMENT_REQUEST_DEFAULT>
    </STUDENT_ENROLLMENT_OPTIONS>
</SSR_STDNT_ENRL_OPTIONS_RESP>
```
<GB_MAP_RULE/>
<GRADE_SCHEME_STDNT>UGD</GRADE_SCHEME_STDNT>
<GRADE_SCHEME_STDNT_LOVDescr>Undergraduate Grading Scheme</GRADE_SCHEME_STDNT_LOVDescr>
<GRADING_BASIS>GRD</GRADING_BASIS>
<GRADING_BASIS_LOVDescr>Graded</GRADING_BASIS_LOVDescr>
<GRADING_BASIS_CONV>GRADING_BASIS_CONV</GRADING_BASIS_CONV>
<GRADING_BASIS_CONV_LOVDescr></GRADING_BASIS_CONV_LOVDescr>
<GRADING_SCHEME>UGD</GRADING_SCHEME>
<GRADING_SCHEME_LOVDescr>Undergraduate Grading Scheme</GRADING_SCHEME_LOVDescr>
<GRADING_BASIS_INV>GRADING_BASIS_INV</GRADING_BASIS_INV>
<SSR_GRD_BASE_PRMPT></SSR_GRD_BASE_PRMPT>
<SETID>PSUNV</SETID>
</CLASS_ENROLLMENT_OPTIONS>
</ENROLLMENT_REQUEST_DEFAULT>
</STUDENT_ENROLLMENT_OPTIONS>
</SSR_STDNT_ENRL_OPTIONS_RESP>
Enrollment Shopping Cart Web Service Operations

Service Name: **SCC_SHOPPING_CART**

The Shopping Cart Web Service allows a student to manage Enrollment shopping cart.

This Service enables a student to:

- Add an item(s) to the cart (SCC_SC_ADDITEM).
- Clear the entire cart (SCC_SC_CLEARCART).
- Retrieve the entire cart (SCC_SC_GETCART).
- Retrieve a particular item from the cart (SCC_SC_GETITEM).
- Enroll cart item(s) (SCC_SC_CHECKOUT).
- Remove selected cart item(s) (SCC_SC_REMOVEITEM).
- Save multiple items to the cart (SCC_SC_SAVECART).
- PreEnrollment Validate (SCC_SC_VALIDATE).

This is a generic shopping cart web service upon which the enrollment shopping cart is built. See: *PeopleSoft Campus Community Fundamentals 9.0 PeopleBook, Using the Shopping Cart Framework*. The *Using the Shopping Cart Framework* chapter was posted to My Oracle Support as part of the Campus Community documentation updates for Bundle 23. Please review Bundle 23 documentation updates and any subsequent bundle documentation updates on My Oracle Support to check if there have been any updates to this chapter.

### Add an Item

**Description**

<table>
<thead>
<tr>
<th>Service</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SCC_SC_ADDITEM: Add an item to the cart</td>
</tr>
<tr>
<td>Summary</td>
<td>This Service Operation adds a single or multiple items to the shopping cart.</td>
</tr>
</tbody>
</table>
Description: This Service Operation adds a single or multiple items to the shopping cart (does not create enrollment request). The enrollment options such as grading basis, units, Instructor Id, class permission nbr etc. will be provided by the student enrollment options service. If the item to be added already exists in the cart, AddItem deletes it from the cart and adds it again. So the UI can make use of AddItem to update an existing cart item also. The UI should invoke SSR_CLASS/SSR_STUDENT_ENROLLMENT_OPTIONS service operation prior to invoking SCC_SC_ADDITEM_REQ to obtain the enrollment options. If the UI is invoking Add Item independently without invoking student enrollment options, then the enrollment options are defaulted if they are supposed to be. Otherwise, if a user value is expected (for example, Related Class) then an error is sent back to the UI to specify a value for the enrollment option.

Users: Student

Processing: Identified incoming cart request (course) and instantiated Course Shopping Cart entities. Validates request parameters. Deletes item(s) if exists in the cart already. Adds new item(s) to the cart. Returns the complete cart for the student along with class section details for each item.

Output: Response message consists of the complete cart along with the newly added items. Each item in the cart will also include the complete class section details.

Error Conditions: Validations performed: Following validations are performed before an item is added to the cart:
- Validate anonymous User.
- Student Active Term Validation.
- Class Number.
- Career pointer exception.
- Grading basis exception.
- Check if student's program allows OEE enrollment.
- Check Last Date to Enroll.
- Check if student is already enrolled in this class.
- System Edits—required, translate, Yes/No, prompt.
- Validate/Default enrollment options:
  - Validate Instructor ID.
  - If a change to the grading basis is allowed, validate the grading basis if a value exists. If a change is not allowed or a value is not provided, assign default.
  - If Related Class is enabled and a value exists then validate related class numbers. If related class is enabled but a value does not exist then send error.
  - If Wait List is enabled and a value exists then validate. If disabled or enabled but value does not exist, assign the default.
  - If Start Date is enabled and a value exists then validate if it is in the start date and end date range and error accordingly. If disabled or enabled but is blank then assign default.
  - Default Course Count.
  - If Unit Taken is disabled, assign default. If enabled and a value exists, validate. If enabled but is blank, then assign default.
  - If Acad Prog is disabled, then assign default. If enabled and a value exists, validate. If enabled but is blank, then assign default.

Input Message: SCC_SC_ADDITEM_REQ

The following diagram shows the SCC_SC_ADDITEM_REQ message structure:
The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:

The Request and Response xsds of SCC_SC_ADDITEM service operation will have the same structure since both represent the Items on course shopping cart. In the above diagram, SSR_CLASS_SECTION has been defined as the child of ITEM so this will reflect both in request as well as in the response xsd. The request xml need not include the SSR_CLASS_SECTION at all since it represents the class section details of the ITEM and hence it will only be part of the response xml. Refer to the sample SCC_SC_ADDITEM Request xml shown in the following section.

**Figure 41: SCC_SC_ADDITEM_REQ Message Structure**
The following are the input parameters for the UI to pass to the service operation:

- *Userid/Password
- *SCC_SHOP_CART_TYPE
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number
- Wait List Okay
- Class Permission Nbr
- Grading Basis
- Units Taken
- Instructor ID
- Requirement Designation Option
- Start Date
- Course Count
- Academic Program
- Related Class Number 1
- Related Class Number 2

- Parameters marked with * are mandatory.
- Parameter SCC_SHOP_CART_TYPE should be set to COURSE by the UI.
- The UI should invoke SSR_CLASS/SSR_STUDENT_ENROLLMENT_OPTIONS service operation prior to invoking SCC_SC_ADDITEM_REQ to obtain the enrollment options (values for all non required fields above). If the UI is invoking Add Item independently without invoking student enrollment options, then the enrollment options are either defaulted if they are supposed to be otherwise if a user value is expected (for example, Related Class) then an error is sent back to the UI to specify a value for the enrollment option.
- User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
- SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.
- EmplID - the ID of the Student.
- Institution – the Institution of the student.
- Academic Career – the Career of the student.
- Term – the Term of the student.
- Class Number - the Class Number that the student is trying adding.
- Related Class Number 1 & 2 – Classes that are related to the selected Class number. Can be selected from LOV by user if enabled.
- Wait list okay – Y/N flag to be set by user if enabled.
- Grading Bases – Can be selected from LOV by user if enabled.
- Units taken – Number of units, can be selected by user if enabled.
- Academic Program - Can be selected from LOV by user if enabled.
- Start Date - if enabled, updated by user.
- Class Permission Nbr – Permission number to be entered by user if enabled.
- Instructor Id - Can be selected from LOV by user if enabled.
- Requirement Designation Option - Y/N flag to be set by user if enabled.

The following is an example of the SCC_SC_ADDITEM_REQ message that the SCC_SC_ADDITEM service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SCC_SC_ADDITEM_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
    <ITEMS>
      <ITEM>
        <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
        <EMPLID>SRI2030</EMPLID>
        <INSTITUTION>PSUNV</INSTITUTION>
      </ITEM>
    </ITEMS>
  </SHOPPING_CART>
</SCC_SC_ADDITEM_REQ>
```
<schema targetNamespace="http://xmlns.oracle.com/Enterprise/HCM/services">
  <include schemaLocation="SCC_ENTITY_SHOP_CART.V1.xsd"/>
</schema>

Output Message: SCC_SC_ADDITEM_RESP

When the Integration Broker receives the SCC_SC_ADDITEM_REQ message, it responds with the SCC_SC_ADDITEM_RESP message.

The following diagram shows the SCC_SC_ADDITEM_RESP message structure:

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:
Figure 12: SCC_SC_ADDITEM_RESP Message Structure

The following is an example of the SCC_SC_ADDITEM_RESP message that the SCC_SC_ADDITEM service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?><SCC_SC_ADDITEM_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services"><SHOPPING_CART><COURSE_SHOP_CART><SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE><ITEMS><ITEM><EMPLID>SR12030</EMPLID><INSTITUTION>PSUNV</INSTITUTION><INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr><ACAD_CAREER>UGRD</ACAD_CAREER><ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr><STRM>0670</STRM></ITEM></ITEMS></COURSE_SHOP_CART></SHOPPING_CART></SCC_SC_ADDITEM_RESP>
<INSTRUCTION_MODE>P</INSTRUCTION_MODE>
<INSTRUCTION_MODE_LOVDescr>In Person</INSTRUCTION_MODE_LOVDescr>
<LOCATION>PSCSHCDA</LOCATION>
<LOCATION_DESCR>Hacienda</LOCATION_DESCR>
<MIN_ENRL>0</MIN_ENRL>
<PRINT_TOPIC>N</PRINT_TOPIC>
<SCHEDULE_PRINT>Y</SCHEDULE_PRINT>
<SCHEDULE_PRINT_LOVDescr>Yes</SCHEDULE_PRINT_LOVDescr>
<SESSION_CODE>1</SESSION_CODE>
<SESSION_CODE_LOVDescr>Regular Academic Session</SESSION_CODE_LOVDescr>
<SSR_CLASSNOTE_LONG></SSR_CLASSNOTE_LONG>
<SSR_COMPONENT>LEC</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
<SSR_CRSE_ATTR_LONG></SSR_CRSE_ATTR_LONG>
<SSR_CRSE_TYPOFF_CD>FALL, SPR</SSR_CRSE_TYPOFF_CD>
<SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
<SSR.Drop Consent>N</SSR.Drop Consent>
<SSR_DROP_CONSENT_LOVDescr>No Special Consent Required</SSR_DROP_CONSENT_LOVDescr>
<SSR_GBL_NOTE_LONG></SSR_GBL_NOTE_LONG>
<SSR_REQUISITE_LONG></SSR_REQUISITE_LONG>
<START_DT>2011-08-30</START_DT>
<STRM>0670</STRM>
<STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
<SUBJECT>CHEM</SUBJECT>
<SUBJECT_LOVDescr>Chemistry</SUBJECT_LOVDescr>
<UNITS_RANGE>3 units</UNITS_RANGE>
<WAIT_TOT>0</WAIT_TOT>
<WAIT_CAP>35</WAIT_CAP>
<Class Combined Section/>
<Class Components>
<Class Component>
<CLASS_NBR>1721</CLASS_NBR>
<Class Type>E</CLASS_TYPE>
<Class Type_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<Class Section>2A</CLASS_SECTION>
<ENRL_STAT>O</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<Class Meeting Patterns>
<Class Meeting Pattern>
<br>SCHL_MTG_SCHL_LONG>TBA</br><SSR_MTG_SCHED_LONG>
<br>SCHL_MTG_LOC_LONG>TBA</br><SSR_MTG_LOC_LONG>
<br>SCHL_INSTR_LONG>Staff</br><SSR_INSTR_LONG>
<br>SCHL_MTG_DT_LONG>TBA</br><SSR_MTG_DT_LONG>
<br>SCHL_TOPIC_LONG>TBA</br><SSR_TOPIC_LONG>
</CLASS_MEETING_PATTERN>
</CLASS_MEETING_PATTERNS>
</CLASS_COMPONENT_SECTION>
<Class Component Section>
<Class NBR>1722</CLASS_NBR>
<Class Type>E</CLASS_TYPE>
<Class Type_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<Class Section>2B</CLASS_SECTION>
<SSR_AUTO_ENRL_SECT>N</SSR_AUTO_ENRL_SECT>
<ENRL_STAT>0</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>

<CLASS_MEETING_PATTERNS>
  <CLASS_MEETING_PATTERN>
    <SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
    <SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
    <SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
    <SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
    <SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
  </CLASS_MEETING_PATTERN>
</CLASS_MEETING_PATTERNS>

<CLASS_COMPONENT_SECTIONS/>

<CLASS_COMPONENT>
  <SSR_COMPONENT>LEC</SSR_COMPONENT>
  <SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
</CLASS_COMPONENT>

<CLASS_COMPONENTS/>

<CLASS_MEETING_PATTERNS>
  <CLASS_MEETING_PATTERN>
    <SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
    <SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
    <SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
    <SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
    <SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
  </CLASS_MEETING_PATTERN>
</CLASS_MEETING_PATTERNS>

<CLASS_TEXT_BOOKS>
  <CLASS_TEXT_BOOK>
    <SSR_CLS_TXB_NONE></SSR_CLS_TXB_NONE>
    <SSR_CLS_TXB_STATUS></SSR_CLS_TXB_STATUS>
    <SSR_CLS_TXB_STATUS_LOVDescr></SSR_CLS_TXB_STATUS_LOVDescr>
    <SSR_CLS_TXB_TEXT></SSR_CLS_TXB_TEXT>
    <TEXTBOOK_MESSAGE>Textbooks to be determined</TEXTBOOK_MESSAGE>
    <SSR_CLS_TXB_NONE_DESCR></SSR_CLS_TXB_NONE_DESCR>
    <TEXT_BOOKS_DETAILS/>
  </CLASS_TEXT_BOOK>
</CLASS_TEXT_BOOKS>

<ENROLLMENT_DETAILS>
  <ENROLLMENT_INFORMATION>
    <ADD_CONSENT></ADD_CONSENT>
    <ADD_CONSENT_LOVDescr></ADD_CONSENT_LOVDescr>
    <DROP_CONSENT></DROP_CONSENT>
    <DROP_CONSENT_LOVDescr></DROP_CONSENT_LOVDescr>
    <ENROLLMENT_REQUIREMENTS></ENROLLMENT_REQUIREMENTS>
    <REQUIREMENT_DESIGNATION></REQUIREMENT_DESIGNATION>
    <CLASS_ATTRIBUTES></CLASS_ATTRIBUTES>
    <ENROLLMENT_INFORMATION>
  </ENROLLMENT_INFORMATION>
</ENROLLMENT_DETAILS>
Clear Cart

Description

<table>
<thead>
<tr>
<th>Service:</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SCC_SC_CLEARCART: delete all items for a cart for a student</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation is used to delete all items from the enrollment cart for a student.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation is used to delete all items from the enrollment cart for a student. User Id is part of the request and the Emplid is the emplid associated with the User Id.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>Identifies incoming cart request (course) and instantiates Course Shopping Cart entities. Clears the existing enrollment cart for the request User Id’s Emplid. Returns blank xml because the enrollment cart is empty.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response consists of blank xml because the enrollment cart is empty.</td>
</tr>
</tbody>
</table>

Input Message: SCC_SC_CLEARCART_REQ

The following diagram shows the SCC_SC_CLEARCART_REQ message structure:

```
<schema targetNamespace="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SCC_SC_CLEARCART_REQ>
    <SHOPPING_CART>
      <COURSE_SHOP_CART>
        <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      </COURSE_SHOP_CART>
    </SHOPPING_CART>
  </SCC_SC_CLEARCART_REQ>
</schema>
```

Figure 51: SCC_SC_CLEARCART_REQ Message Structure

The following are the input parameters for the UI to pass to the service operation:

- *UserID
- *Password
- *SCC_SHOP_CART_TYPE

The following is an example of the SCC_SC_CLEARCART_REQ message that the SCC_SC_CLEARCART service operation receives from an Enrollment application:

```
<?xml version="1.0"?>
<SCC_SC_CLEARCART_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SCC_SC_CLEARCART_REQ>
```
<SCC_SC_CLEARCART_REQ>

Parameters marked with * are mandatory. User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.

**Output Message: SCC_SC_CLEARCART_RESP**

When the Integration Broker receives the SCC_SC_CLEARCART_REQ message, it responds with the SCC_SC_CLEARCART_RESP message.
Note: Clear cart responds with a blank xml since the cart is cleared and there will be nothing in the cart to send back.

The following diagram shows the SCC_SC_CLEARCART_RESP message structure:

```
<schema>
  targetNamespace "http://xmlns.oracle.com/Enterprise/HCM/services"
  <SCC_SC_CLEARCART_RESP/>
</schema>
```

**Figure 12: SCC_SC_CLEARCART_RESP Message Structure**

The following is an example of the SCC_SC_CLEARCART_RESP message that the SCC_SC_CLEARCART service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SCC_SC_CLEARCART_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services"/>
```

**Get Cart**

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SCC_SC_GETCART: retrieve all items from a cart for a student</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation is used to retrieve all items from the cart for a student.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation is used to retrieve all items from the cart for a student along with the class section details for each item.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>Identifies incoming cart request (course) and instantiates Course Shopping Cart entities. Retrieves the existing enrollment cart for the request User Id’s Emplid. Returns enrollment cart in the response along with class section details for each item.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response consists of enrollment cart for the student along with class section details for each item.</td>
</tr>
</tbody>
</table>
**Input Message: SCC_SC_GETCART_REQ**

The following diagram shows the SCC_SC_GETCART_REQ message structure:

```
<schema>
    targetNamespace "http://xmlns.oracle.com/Enterprise/HCM/services"

    SCC_SC_GETCART_REQ -- SHOPPING_CART -- SCC_SHOP_CART_TYPE
```

**Figure 61: SCC_SC_GETCART_REQ Message Structure**

The following are the input parameters for the UI to pass to the service operation:

- *Userid/Password
- *SCC_SHOP_CART_TYPE
- *EMPLID

Parameters marked with * are mandatory. User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token. SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.

The following is an example of the SCC_SC_GETCART_REQ message that the SCC_SC_GETCART service operation receives from an Enrollment application:

```xml
<?XML version="1.0"?>
<SCC_SC_GETCART_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
    <SHOPPING_CART>
        <COURSE_SHOP_CART>
            <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
        </COURSE_SHOP_CART>
    </SHOPPING_CART>
</SCC_SC_GETCART_REQ>
```

**Output Message: SCC_SC_GETCART_RESP**

When the Integration Broker receives the SCC_SC_GETCART_REQ message, it responds with the SCC_SC_GETCART_RESP message.

The following diagram shows the SCC_SC_GETCART_RESP message structure:
The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:

```xml
<targetNamespace xmlns:scce="http://xmlns.oracle.com/Enterprise/HCM/services">
  <include>
    <schemaLocation>SCC_ENTITY_SHOP_CART.V1.xsd</schemaLocation>
  </include>
</targetNamespace>
```

**Figure 12: SCC_SC_GETCART_RESP Message Structure**

The following is an example of the SCC_SC_GETCART_RESP message that the SCC_SC_GETCART service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SCC_SC_GETCART_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
      </ITEMS>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SCC_SC_GETCART_RESP>
```
<ENRL_STAT>0</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<ENRL_TOT>0</ENRL_TOT>
<GRADING_BASIS>GRD</GRADING_BASIS>
<GRADING_BASIS_LOVDescr>Graded</GRADING_BASIS_LOVDescr>
<INSTITUTION>PSUNV</INSTITUTION>
<INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
<INSTRUCTION_MODE>P</INSTRUCTION_MODE>
<INSTRUCTION_MODE_LOVDescr>In Person</INSTRUCTION_MODE_LOVDescr>
<LOCATION>PSCSHCDA</LOCATION>
<LOCATION_DESCR>Hacienda</LOCATION_DESCR>
<MIN_ENRL>0</MIN_ENRL>
<PRINT_TOPIC>N</PRINT_TOPIC>
<SCHEDULE_PRINT>Y</SCHEDULE_PRINT>
<SCHEDULE_PRINT_LOVDescr>Yes</SCHEDULE_PRINT_LOVDescr>
<SESSION_CODE>1</SESSION_CODE>
<SESSION_CODE_LOVDescr>Regular Academic Session</SESSION_CODE_LOVDescr>
<SSR_CLASSNOTE_LONG></SSR_CLASSNOTE_LONG>
<SSR_COMPONENT>LEC</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
<SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
<SSR_DROP_CONSENT>N</SSR_DROP_CONSENT>
<SSR_DROP_CONSENT_LOVDescr>No Special Consent Required</SSR_DROP_CONSENT_LOVDescr>
<SSR_GBL_NOTE_LONG></SSR_GBL_NOTE_LONG>
<SSR_REQUISITE_LONG></SSR_REQUISITE_LONG>
<START_DT>2011-08-30</START_DT>
<STRM>0670</STRM>
<STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
<SUBJECT>CHEM</SUBJECT>
<SUBJECT_LOVDescr>Chemistry</SUBJECT_LOVDescr>
<UNITS_RANGE>3 units</UNITS_RANGE>
<WAIT_TOT>0</WAIT_TOT>
<WAIT_CAP>35</WAIT_CAP>
<Class Combined Section>
<Class Component>
<Class Component></Class Component>
-Class Component LAB</SSR_COMPONENT>
-Class Component_LOVDescr>Laboratory</SSR_COMPONENT_LOVDescr>
<OPTIONAL_SECTION>N</OPTIONAL_SECTION>
<OPTIONAL_SECTION_LOVDescr>Required</OPTIONAL_SECTION_LOVDescr>
-Class Component Sections>
-Class NBR>1721</CLASS NBR>
-Class Type></CLASS TYPE>
-Class Type_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
-Class Section>2A</CLASS_SECTION>
-Class Auto Enrl Sect>N</CLASS_AUTO_ENRL_SECT>
-Class Enrl Stat>O</CLASS_ENRL_STAT>
-Class Status Descr>Open</CLASS_STATUS_DESCR>
-Class Campus>MAIN</CLASS_CAMPUS>
-Class LOVDescr>Main Hacienda Campus</CLASS_LOVDescr>
-Class Meeting Patterns>
-Class Meeting Pattern>
-Class Meeting Pattern></CLASS_MEETING_PATTERN>
-Class Meeting Pattern TBA</SSR_MTG_SCHED_LONG>
-Class Meeting Pattern TBA</SSR_MTG_LARGE_LONG>
-Class Meeting Pattern Staff</SSR_INSTR_LONG>
-Class Meeting Pattern TBA</SSR_MTG_DT_LONG>
-Class Meeting Pattern TBA</SSR_TOPIC_LONG>
-Class Meeting Pattern></CLASS_MEETING_PATTERN>
Get Item

Description

<table>
<thead>
<tr>
<th>Service</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SCC_SC_GETITEM: retrieve an item based on the index of the item passed in the request.</td>
</tr>
<tr>
<td>Summary</td>
<td>This Service Operation is used to retrieve an item based on the index of the item passed in the request. Index no is a numeric value.</td>
</tr>
<tr>
<td>Description</td>
<td>This service operation can be used to retrieve a particular item from the cart. The item retrieved is based on the index of the item passed in the request.</td>
</tr>
<tr>
<td>Users</td>
<td>Student</td>
</tr>
<tr>
<td>Processing</td>
<td>Based on the Cart Type (Course) builds the course shopping cart entities. Validates the index no. Retrieves the requested item and sends back in the response.</td>
</tr>
<tr>
<td>Output</td>
<td>Responds with the requested item along with class section details of the item.</td>
</tr>
<tr>
<td>Error Conditions</td>
<td>Validate anonymous user. Validate index no – Error if Index No is greater than the number of cart items.</td>
</tr>
</tbody>
</table>

Input Message: SCC_SC_GETITEM_REQ

The following diagram shows the SCC_SC_GETITEM_REQ message structure:

![Diagram of SCC_SC_GETITEM_REQ Message Structure]

Figure 71: SCC_SC_GETITEM_REQ Message Structure

The following are the input parameters for the UI to pass to the service operation:

- *UserID
- *Password
- *Index No
• *SCC_SHOP_CART_TYPE*

The following is an example of the SCC_SC_GETITEM_REQ message that the SCC_SC_GETITEM service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SCC_SC_GETITEM_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
 <SHOPPING_CART>
  <COURSE_SHOP_CART>
   <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
   <SCC_INDEX>1</SCC_INDEX>
  </COURSE_SHOP_CART>
 </SHOPPING_CART>
</SCC_SC_GETITEM_REQ>
```

Parameters marked with * are mandatory.
User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.
Index No: Number value, index of an item on the cart.

**Output Message: SCC_SC_GETITEM_RESP**

When the Integration Broker receives the SCC_SC_GETITEM_REQ message, it responds with the SCC_SC_GETITEM_RESP message.

The following diagram shows the SCC_SC_GETITEM_RESP message structure:

```
<schema>
    <xs:annotation>
        <xs:appinfo nameSpaceInput=http://xmlns.oracle.com/Enterprise/HCM/services>
            <xs:sinceRelease>1.0</xs:sinceRelease>
        </xs:appinfo>
        <xs:documentation />
    </xs:annotation>
    <xs:element name="SCC_INDEX" type="xs:decimal"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:string">COURSE</xs:element>
    <xs:element name="SHOPPING_CART" type="xs:QName"/>
    <xs:element name="COURSE_SHOP_CART" type="xs:QName"/>
    <xs:element name="SCC_ENTITY_INST_ID" type="xs:QName"/>
    <xs:element name="ITEMS" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_RESP" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_REQ" type="xs:QName"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:QName"/>
    <xs:element name="SCC_INDEX" type="xs:QName"/>
    <xs:element name="SHOPPING_CART" type="xs:QName"/>
    <xs:element name="COURSE_SHOP_CART" type="xs:QName"/>
    <xs:element name="SCC_ENTITY_INST_ID" type="xs:QName"/>
    <xs:element name="ITEMS" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_RESP" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_REQ" type="xs:QName"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:QName"/>
    <xs:element name="SCC_INDEX" type="xs:QName"/>
    <xs:element name="SHOPPING_CART" type="xs:QName"/>
    <xs:element name="COURSE_SHOP_CART" type="xs:QName"/>
    <xs:element name="SCC_ENTITY_INST_ID" type="xs:QName"/>
    <xs:element name="ITEMS" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_RESP" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_REQ" type="xs:QName"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:QName"/>
    <xs:element name="SCC_INDEX" type="xs:QName"/>
    <xs:element name="SHOPPING_CART" type="xs:QName"/>
    <xs:element name="COURSE_SHOP_CART" type="xs:QName"/>
    <xs:element name="SCC_ENTITY_INST_ID" type="xs:QName"/>
    <xs:element name="ITEMS" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_RESP" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_REQ" type="xs:QName"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:QName"/>
    <xs:element name="SCC_INDEX" type="xs:QName"/>
    <xs:element name="SHOPPING_CART" type="xs:QName"/>
    <xs:element name="COURSE_SHOP_CART" type="xs:QName"/>
    <xs:element name="SCC_ENTITY_INST_ID" type="xs:QName"/>
    <xs:element name="ITEMS" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_RESP" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_REQ" type="xs:QName"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:QName"/>
    <xs:element name="SCC_INDEX" type="xs:QName"/>
    <xs:element name="SHOPPING_CART" type="xs:QName"/>
    <xs:element name="COURSE_SHOP_CART" type="xs:QName"/>
    <xs:element name="SCC_ENTITY_INST_ID" type="xs:QName"/>
    <xs:element name="ITEMS" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_RESP" type="xs:QName"/>
    <xs:element name="SCC_SC_GETITEM_REQ" type="xs:QName"/>
    <xs:element name="SCC_SHOP_CART_TYPE" type="xs:QName"/>
    <xs:element name="SCC_INDEX" type="xs:QName"/>
Figure 12: SCC_SC_GETITEM__RESP Message Structure

The following is an example of the SCC_SC_GETITEM__RESP message that the SCC_SC_GETITEM service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SCC_SC_GETITEM_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
        <ITEM>
          <EMPLID>SR12030</EMPLID>
          <INSTITUTION>PSUNV</INSTITUTION>
          <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
          <ACAD_CAREER>UGRD</ACAD_CAREER>
          <ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
          <STRM>0670</STRM>
          <STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
          <CLASS_NBR>1718</CLASS_NBR>
          <CLASS_NBR_LOVDescr>CHEM 201 - Organic Chem I</CLASS_NBR_LOVDescr>
          <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
          <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
          <GRADING_BASIS_ENRL>GRD</GRADING_BASIS_ENRL>
          <GRADING_BASIS_ENRL_LOVDescr>Graded</GRADING_BASIS_ENRL_LOVDescr>
          <UNT_TAKEN>3</UNT_TAKEN>
          <INSTRUCTOR_ID></INSTRUCTOR_ID>
          <RQMNT_DESIGNTN_OPT>N</RQMNT_DESIGNTN_OPT>
        </ITEM>
      </ITEMS>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SCC_SC_GETITEM_RESP>
```
<CRSE_COUNT>1</CRSE_COUNT>
<CLASS_SECTION>
<ACAD_PROGRAM>LAU</ACAD_PROGRAM>
<RELATE_CLASS_NBR_1>1721</RELATE_CLASS_NBR_1>
<RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
<SCC_ROW_ADD_DTTM>01/09/2012 06:49:20.000000</SCC_ROW_ADD_DTTM>
<QUANTITY>0</QUANTITY>
<RQMNT_DESIGNTN>DSGN</RQMNT_DESIGNTN>
<RQMNT_DESIGNTN_LOVDescr>Design Course</RQMNT_DESIGNTN_LOVDescr>
<ENRL_REQ_DETL_STAT>P</ENRL_REQ_DETL_STAT>
<ENRL_REQ_DETL_STAT_LOVDescr>Pending</ENRL_REQ_DETL_STAT_LOVDescr>
<DTTM_STAMP_SEC>01/09/2012 06:49:19.000000</DTTM_STAMP_SEC>
<SSR_VALIDATE>N</SSR_VALIDATE>
<SCC_ROW_UPD_OPRID>SSS_SR12030</SCC_ROW_UPD_OPRID>
<SCC_ROW_UPD_DTTM>01/09/2012 06:49:20.000000</SCC_ROW_UPD_DTTM>
<CLASS_SECTIONS>
<SSR_CLASS_SECTION>
<ACAD_CAREER>UGRD</ACAD_CAREER>
<ASSOCIATED_CLASS>2</ASSOCIATED_CLASS>
<AVALIABLE_SEATS>100</AVAILABLE_SEATS>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
<CATALOG_NBR>201</CATALOG_NBR>
<class_nbr>1718</CLASS_NBR>
<class_section>2</CLASS_SECTION>
<class_type>E</CLASS_TYPE>
<class_type_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<combined_section>2</combined_section>
<combined_section_LOVDescr></COMBINED_SECTION_LOVDescr>
<CONSENT>N</CONSENT>
<CONSENT_LOVDescr>No Special Consent Required</CONSENT_LOVDescr>
<CRS_TOPIC_ID>1</CRS_TOPIC_ID>
<CRS_TOPIC_DESCR>Chem 201 - 2 Organic Chemistry I</CRS_TOPIC_DESCR>
<CRS_ID_LOVDescr>Organic Chem I</CRS_ID_LOVDescr>
<CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
<CRSE_ID>003746</CRSE_ID>
<DESCR200>CHEM  201 - 2   Organic Chemistry I</DESCR200>
<DESCRLONG></DESCRLONG>
<EFFDT>2011-08-30</EFFDT>
<END_DT>2011-12-12</END_DT>
<ENRL_CAP>100</ENRL_CAP>
<ENRL_STAT>O</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<ENRL_TOT>0</ENRL_TOT>
<GRADING_BASIS>GRD</GRADING_BASIS>
<GRADING_BASIS_LOVDescr>Graded</GRADING_BASIS_LOVDescr>
<INSTITUTION>PSUNV</INSTITUTION>
<INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
<INSTRUCTION_MODE>P</INSTRUCTION_MODE>
<INSTRUCTION_MODE_LOVDescr>In Person</INSTRUCTION_MODE_LOVDescr>
,LOCATION>PSCSHCDA</LOCATION>
,LOCATION_DESCR>Hacienda</LOCATION_DESCR>
<MIN_ENRL>0</MIN_ENRL>
<PRINT_TOPIC>N</PRINT_TOPIC>
<SCHEDULE_PRINT>Y</SCHEDULE_PRINT>
<SCHEDULE_PRINT_LOVDescr>Yes</SCHEDULE_PRINT_LOVDescr>
<SESSION_CODE>1</SESSION_CODE>
<SESSION_CODE_LOVDescr>Regular Academic Session</SESSION_CODE_LOVDescr>
<SSR_CLASSNOTE_LONG></SSR_CLASSNOTE_LONG>
<SSR_COMPONENT>LEC</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
<SSR_CRSE_ATTR_LONG><SSR_CRSE_ATTR_LONG>
<SSR_CRSE_TYPOFF_CD>FALL, SPR</SSR_CRSE_TYPOFF_CD>
<SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
<SSR_DROP_CONSENT>N</SSR_DROP_CONSENT>
<SSR_DROP_CONSENT_LOVDescr>No Special Consent Required</SSR_DROP_CONSENT_LOVDescr>
<SSR_GBL_NOTE_LONG><SSR_GBL_NOTE_LONG>
<SSR_REQUISITE_LONG><SSR_REQUISITE_LONG>
<START_DT>2011-08-30</START_DT>
<STRM>0670</STRM><STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
<SUBJECT>CHEM</SUBJECT><SUBJECT_LOVDescr>Chemistry</SUBJECT_LOVDescr>
<UNITS_RANGE>3 units</UNITS_RANGE>
<WAIT_TOT>0</WAIT_TOT>
<WAIT_CAP>35</WAIT_CAP>
<class_components/>
<class_components>
<CLASS_COMPONENT>
<SSR_COMPONENT>LAB</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Laboratory</SSR_COMPONENT_LOVDescr>
<OPTIONAL_SECTION>N</OPTIONAL_SECTION>
<OPTIONAL_SECTION_LOVDescr>Required</OPTIONAL_SECTION_LOVDescr>
<class_component_sections>
<class_component_section>
<class_nbr>1721</class_nbr>
<class_type>E</class_type>
<class_type_LOVDescr>Enrollment Section</class_type_LOVDescr>
<class_section>2A</class_section>
<enrl_stat>0</enrl_stat>
<enrl_status_descr>Open</enrl_status_descr>
<campus>MAIN</campus>
<class_meeting_patterns>
<class_meeting_pattern>
<ssr_mtg_sched_long>TBA</ssr_mtg_sched_long>
<ssr_mtg_loc_long>TBA</ssr_mtg_loc_long>
<ssr_instr_long>Staff</ssr_instr_long>
<ssr_mtg_dt_long>TBA</ssr_mtg_dt_long>
<ssr_topic_long>TBA</ssr_topic_long>
</class_meeting_pattern>
</class_meeting_patterns>
</class_component_section>
<class_component_section>
<class_nbr>1722</class_nbr>
<class_type>E</class_type>
<class_type_LOVDescr>Enrollment Section</class_type_LOVDescr>
<class_section>2B</class_section>
<enrl_stat>0</enrl_stat>
<enrl_status_descr>Open</enrl_status_descr>
<campus>MAIN</campus>
<class_meeting_patterns>
<class_meeting_pattern>
<ssr_mtg_sched_long>TBA</ssr_mtg_sched_long>
<ssr_mtg_loc_long>TBA</ssr_mtg_loc_long>
<ssr_instr_long>Staff</ssr_instr_long>
<ssr_mtg_dt_long>TBA</ssr_mtg_dt_long>
<ssr_topic_long>TBA</ssr_topic_long>
</class_meeting_pattern>
</class_meeting_patterns>
</class_component_section>
<class_component_section>
<class_nbr>1722</class_nbr>
<class_type>E</class_type>
<class_type_LOVDescr>Enrollment Section</class_type_LOVDescr>
<class_section>2B</class_section>
<enrl_stat>0</enrl_stat>
<enrl_status_descr>Open</enrl_status_descr>
<campus>MAIN</campus>
<class_meeting_patterns>
<class_meeting_pattern>
<ssr_mtg_sched_long>TBA</ssr_mtg_sched_long>
<ssr_mtg_loc_long>TBA</ssr_mtg_loc_long>
<ssr_instr_long>Staff</ssr_instr_long>
<ssr_mtg_dt_long>TBA</ssr_mtg_dt_long>
<ssr_topic_long>TBA</ssr_topic_long>
</class_meeting_pattern>
</class_meeting_patterns>
</class_component_section>
</class_component_sections>
</class_component>
</class_components>
<table>
<thead>
<tr>
<th>Service:</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SCC_SC_SAVECART: add multiple items to the cart</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation is used to save multiple items to the cart. The operation deletes the existing items on the cart and saves the new items.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation is used to save multiple items to the cart. The operation deletes the existing items on the cart and saves the new items. The UI should invoke SSR_CLASS/SSR_STUDENT_ENROLLMENT_OPTIONS service operation prior to invoking SCC_SC_SAVECART_REQ to obtain the enrollment options. If the UI is invoking SaveCart independently without invoking student enrollment options, then the enrollment options are either defaulted if they are supposed to be, otherwise, if a user value is expected (for example, Related Class) then an error is sent back to the UI to specify a value for the enrollment option.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>Identifies incoming cart request (course) and instantiates Course Shopping Cart entities. Validates request parameters. Clears the existing items. Saves new item(s) to the cart. Returns the complete cart for the student along with class section details of each item.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response consists of the existing cart which will have the newly saved items. The response also includes the class section details of each item.</td>
</tr>
<tr>
<td>Error Conditions:</td>
<td>Validations performed: Following validations are performed before an item is saved to the cart: • Validate anonymous User. • Student Active Term Validation. • Class Number. • Career pointer exception. • Grading basis exception. • Check if student’s program allows OEE enrollment. • Check Last Date to Enroll. • Check if student is already enrolled in this class. • System Edits-- required, translate, Yes/No, prompt. • Validate/Default enrollment options: • Validate Instructor ID. • If a change to the grading basis is allowed, validate the grading basis if a value exists. If a change is not allowed or a value is not provided, assign default. • If Related Class is enabled and a value exists then validate related class numbers. If related class is enabled but a value does not exist then send error. • If Wait List is enabled and a value exists then validate. If disabled or enabled but value does not exist, assign the default. • If Start Date is enabled and a value exists then validate if it is in the start date and end date range and error accordingly. If disabled or enabled but is blank then assign default. • Default Course Count. • If Unit Taken is disabled, assign default. If enabled and a value exists, validate. If enabled but is blank, then assign default. • If Acad Prog is disabled, then assign default. If enabled and a value exists, validate. If enabled but is blank, then assign default.</td>
</tr>
</tbody>
</table>

**Input Message: SCC_SC_SAVECART_REQ**

The following diagram shows the SCC_SC_SAVECART_REQ message structure:
The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:

```
<schema>
  targetNamespace http://xmlns.oracle.com/EnterpriseHCM/services
  <include>
    schemaLocation SCC_ENTITY_SHOP_CART.V1.xsd
  </include>

  <element name="SHOPPING_CART">
    <complexType>
      <sequence>
        <element ref="ENROLLMENT_INFORMATION" />
        <element ref="TEXT_BOOK_DETAILS" />
        <element ref="CLASS_TEXT_BOOK" />
        <element ref="CLASS_MEETING_PATTERN" />
        <element ref="CLASS_COMPONENT_PATTERN" />
        <element ref="CLASS_COMPONENT" />
        <element ref="COMBINED_SECTION" />
        <element ref="SSR_CLASS_SECTION" />
        <element ref="ITEM" />
        <element ref="SHOPPING_CART" />
        <element ref="COURSE_SHOP_CART" />
      </sequence>
    </complexType>
  </element>

  <element name="COURSE_SHOP_CART">
    <complexType>
      <sequence>
        <element ref="SCC_PROCESS_TYPE" />
        <element ref="SCC_PROCESS_MSGBODY" />
      </sequence>
    </complexType>
  </element>

  <element name="ITEM">
    <complexType>
      <sequence>
        <element ref="SCC_SHOP_CART_TYPE" />
        <element ref="SCC_ENTITY_INST_ID" />
      </sequence>
    </complexType>
  </element>

  <element name="SCC_SHOP_CART_TYPE">
    <complexType>
      <attributeName name="name" type="xs:string" />
      <attributeName name="class" type="xs:string" />
      <attributeName name="category" type="xs:string" />
    </complexType>
  </element>

  <element name="SCC_ENTITY_INST_ID">
    <complexType>
      <attributeName name="value" type="xs:string" />
    </complexType>
  </element>
```

The Request and Response xsds of SCC_SC_SAVECART service operation will have the same structure since both represent the Items on course shopping cart. In the above diagram, SSRCLASSSECTION has been defined as the child of ITEM so this will reflect both in request as well as in the response xsd. The request xml need not include the SSRCLASSSECTION at all since it represents the class section details of the ITEM and hence it will only be part of the response xml.

**Figure 81: SCC_SC_SAVECART_REQ Message Structure**

The following are the input parameters for the UI to pass to the service operation:

- *Userid/Password
- *SCC_SHOP_CART_TYPE
• *EMPLID
• *INSTITUTION
• *Academic Career
• *Term
• *Class Number
• Wait List Okay
• Class Permission Nbr
• Grading Basis
• Units Taken
• Instructor ID
• Requirement Designation Option
• Start Date
• Course Count
• Academic Program
• Related Class Number 1
• Related Class Number 2

• Parameters marked with * are mandatory.
• Parameter SCC_SHOP_CART_TYPE should be set to COURSE by the UI.
• The UI should invoke SSR_CLASS/SSR_STUDENT_ENROLLMENT_OPTIONS service operation prior to invoking SCC_SC_SAVECART_REQ to obtain the enrollment options (values for all non required fields above). If the UI is invoking Save Cart independently without invoking student enrollment options, then the enrollment options are either defaulted if they are supposed to be otherwise if a user value is expected (for example, Related Class) then an error is sent back to the UI to specify a value for the enrollment option.
• User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
• SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.
• EmplID - the ID of the Student.
• Institution – the Institution of the student.
• Academic Career – the Career of the student.
• Term – the Term of the student.
• Class Number - the Class Number that the student is trying to add.
• Related Class Number 1 & 2 – Classes that are related to the selected Class number. Can be selected from LOV by user if enabled.
• Wait list okay – Y/N flag to be set by user if enabled.
• Grading Bases – Can be selected from LOV by user if enabled.
• Units taken – Number of units, can be selected by user if enabled.
• Academic Program - Can be selected from LOV by user if enabled.
• Start Date - Can be selected by user if enabled.
• Class Permission Nbr – Permission number to be entered by user if enabled.
• Instructor Id - Can be selected from LOV by user if enabled.
• Requirement Designation Option - Y/N flag to be set by user if enabled.

The following is an example of the SCC_SC_SAVECART_REQ message that the SCC_SC_SAVECART service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SCC_SC_SAVECART_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
        <ITEM>
          <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
          <EMPLID>SR12030</EMPLID>
          <INSTITUTION>PSUNV</INSTITUTION>
          <ACAD_CAREER>UGRD</ACAD_CAREER>
          <STRM>0670</STRM>
          <CLASS_NBR>1718</CLASS_NBR>
          <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
          <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
```
When the Integration Broker receives the SCC_SC_SAVECART_REQ message, it responds with the SCC_SC_SAVECART_RESP message.

The following diagram shows the SCC_SC_SAVECART_RESP message structure:

![Diagram showing the structure of SCC_SC_SAVECART_RESP message](image.png)

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:

```xml
<include>
    <schemaLocation>http://xmlns.oracle.com/Enterprise/HCMessages
      <include>
        <SCC_ENTITY_SHOP_CART.V1.xsd
```
Figure 12: SCC_SC_SAVECART_RESP Message Structure

The following is an example of the SCC_SC_SAVECART_RESP message that the SCC_SC_SAVECART service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SCC_SC_SAVECART_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
    <SHOPPING_CART>
        <COURSE_SHOP_CART>
            <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
            <ITEMS>
                <ITEM>
                    <EMPLID>SR12030</EMPLID>
                    <INSTITUTION>PSUNV</INSTITUTION>
                    <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
                    <ACAD_CAREER>UGRD</ACAD_CAREER>
                    <ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
                    <STRM>0670</STRM>
                    <STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
                    <CLASS_NBR>1718</CLASS_NBR>
                    <CLASS_NBR_LOVDescr>CHEM 201 - Organic Chem I</CLASS_NBR_LOVDescr>
                    <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
                    <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
                    <GRADING_BASIS_ENRL>GRD</GRADING_BASIS_ENRL>
                    <GRADING_BASIS_ENRL_LOVDescr>Graded</GRADING_BASIS_ENRL_LOVDescr>
                    <UNT_TAKEN>3</UNT_TAKEN>
                    <INSTRUCTOR_ID></INSTRUCTOR_ID>
                    <RQMNT_DESIGNTN_OPT>N</RQMNT_DESIGNTN_OPT>
                    <START_DT></START_DT>
                </ITEM>
            </ITEMS>
        </COURSE_SHOP_CART>
    </SHOPPING_CART>
</SCC_SC_SAVECART_RESP>
```
<table>
<thead>
<tr>
<th><code>&lt;CRSE_COUNT&gt;</code></th>
<th><code>&lt;CRSE_COUNT&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ACAD_PROG&gt;</code></td>
<td><code>&lt;ACAD_PROG&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ACAD_PROG_LOVDescr&gt;</code></td>
<td><code>&lt;ACAD_PROG_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;RELATE_CLASS_NBR_1&gt;</code></td>
<td><code>&lt;RELATE_CLASS_NBR_1&gt;</code></td>
</tr>
<tr>
<td><code>&lt;RELATE_CLASS_NBR_2&gt;</code></td>
<td><code>&lt;RELATE_CLASS_NBR_2&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SCC_ROW_ADD_DTTM&gt;</code></td>
<td><code>&lt;SCC_ROW_ADD_DTTM&gt;</code></td>
</tr>
<tr>
<td><code>&lt;QUANTITY&gt;</code></td>
<td><code>&lt;QUANTITY&gt;</code></td>
</tr>
<tr>
<td><code>&lt;RQMNT_DESIGNTN&gt;</code></td>
<td><code>&lt;RQMNT_DESIGNTN&gt;</code></td>
</tr>
<tr>
<td><code>&lt;RQMNT_DESIGNTN_LOVDescr&gt;</code></td>
<td><code>&lt;RQMNT_DESIGNTN_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ENRL_REQ_DETL_STAT&gt;</code></td>
<td><code>&lt;ENRL_REQ_DETL_STAT&gt;</code></td>
</tr>
<tr>
<td><code>&lt;DTTM_STAMP_SEC&gt;</code></td>
<td><code>&lt;DTTM_STAMP_SEC&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SSR_VALIDATE&gt;</code></td>
<td><code>&lt;SSR_VALIDATE&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SCC_ROW_ADD_OPRID&gt;</code></td>
<td><code>&lt;SCC_ROW_ADD_OPRID&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SCC_ROW_UPD_OPRID&gt;</code></td>
<td><code>&lt;SCC_ROW_UPD_OPRID&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SCC_ROW_UPD_DTTM&gt;</code></td>
<td><code>&lt;SCC_ROW_UPD_DTTM&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CLASS_SECTIONS&gt;</code></td>
<td><code>&lt;CLASS_SECTIONS&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ACAD_CAREER&gt;</code></td>
<td><code>&lt;ACAD_CAREER&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ACAD_CAREER_LOVDescr&gt;</code></td>
<td><code>&lt;ACAD_CAREER_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ASSOCIATED_CLASS&gt;</code></td>
<td><code>&lt;ASSOCIATED_CLASS&gt;</code></td>
</tr>
<tr>
<td><code>&lt;AVAILABLE_SEATS&gt;</code></td>
<td><code>&lt;AVAILABLE_SEATS&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CAMPUS&gt;</code></td>
<td><code>&lt;CAMPUS&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CAMPUS_LOVDescr&gt;</code></td>
<td><code>&lt;CAMPUS_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CATALOG_NBR&gt;</code></td>
<td><code>&lt;CATALOG_NBR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CLASS_NBR&gt;</code></td>
<td><code>&lt;CLASS_NBR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CLASS_SECTION&gt;</code></td>
<td><code>&lt;CLASS_SECTION&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CLASS_TYPE&gt;</code></td>
<td><code>&lt;CLASS_TYPE&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CLASS_TYPE_LOVDescr&gt;</code></td>
<td><code>&lt;CLASS_TYPE_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;COMBINED_SECTION&gt;</code></td>
<td><code>&lt;COMBINED_SECTION&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CONSENT&gt;</code></td>
<td><code>&lt;CONSENT&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CRS_TOPIC_ID&gt;</code></td>
<td><code>&lt;CRS_TOPIC_ID&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CRS_TOPIC_DESCR&gt;</code></td>
<td><code>&lt;CRS_TOPIC_DESCR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CRSE_ID&gt;</code></td>
<td><code>&lt;CRSE_ID&gt;</code></td>
</tr>
<tr>
<td><code>&lt;CRSE_ID_LOVDescr&gt;</code></td>
<td><code>&lt;CRSE_ID_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;DESCR200&gt;</code></td>
<td><code>&lt;DESCR200&gt;</code></td>
</tr>
<tr>
<td><code>&lt;DESCRLONG&gt;</code></td>
<td><code>&lt;DESCRLONG&gt;</code></td>
</tr>
<tr>
<td><code>&lt;EFFDT&gt;</code></td>
<td><code>&lt;EFFDT&gt;</code></td>
</tr>
<tr>
<td><code>&lt;END_DT&gt;</code></td>
<td><code>&lt;END_DT&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ENRL_CAP&gt;</code></td>
<td><code>&lt;ENRL_CAP&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ENRL_STATUS_DESCR&gt;</code></td>
<td><code>&lt;ENRL_STATUS_DESCR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;ENRL_TOT&gt;</code></td>
<td><code>&lt;ENRL_TOT&gt;</code></td>
</tr>
<tr>
<td><code>&lt;GRADING_BASIS&gt;</code></td>
<td><code>&lt;GRADING_BASIS&gt;</code></td>
</tr>
<tr>
<td><code>&lt;GRADING_BASIS_LOVDescr&gt;</code></td>
<td><code>&lt;GRADING_BASIS_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;INSTITUTION&gt;</code></td>
<td><code>&lt;INSTITUTION&gt;</code></td>
</tr>
<tr>
<td><code>&lt;INSTITUTION_LOVDescr&gt;</code></td>
<td><code>&lt;INSTITUTION_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;INSTRUCTION_MODE&gt;</code></td>
<td><code>&lt;INSTRUCTION_MODE&gt;</code></td>
</tr>
<tr>
<td><code>&lt;INSTRUCTION_MODE_LOVDescr&gt;</code></td>
<td><code>&lt;INSTRUCTION_MODE_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;LOCATION&gt;</code></td>
<td><code>&lt;LOCATION&gt;</code></td>
</tr>
<tr>
<td><code>&lt;LOCATION_DESCR&gt;</code></td>
<td><code>&lt;LOCATION_DESCR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;MIN_ENRL&gt;</code></td>
<td><code>&lt;MIN_ENRL&gt;</code></td>
</tr>
<tr>
<td><code>&lt;PRINT_TOPIC&gt;</code></td>
<td><code>&lt;PRINT_TOPIC&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SCHEDULE_PRINT&gt;</code></td>
<td><code>&lt;SCHEDULE_PRINT&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SCHEDULE_PRINT_LOVDescr&gt;</code></td>
<td><code>&lt;SCHEDULE_PRINT_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SESSION_CODE&gt;</code></td>
<td><code>&lt;SESSION_CODE&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SESSION_CODE_LOVDescr&gt;</code></td>
<td><code>&lt;SESSION_CODE_LOVDescr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SSR_CLASSNOTE_LONG&gt;</code></td>
<td><code>&lt;SSR_CLASSNOTE_LONG&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SSR_COMPONENT&gt;</code></td>
<td><code>&lt;SSR_COMPONENT&gt;</code></td>
</tr>
</tbody>
</table>
<SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
<SSR_CRSE_ATTR_LONG></SSR_CRSE_ATTR_LONG>
<SSR_CRSE_TYPOFF_CD>FALL, SPR</SSR_CRSE_TYPOFF_CD>
<SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
<SSR_DROP_CONSENT>N</SSR_DROP_CONSENT>
<SSR_DROP_CONSENT_LOVDescr>No Special Consent
Required</SSR.Drop_CONSENT_LOVDescr>
<SSR_GBL_NOTE_LONG></SSR_GBL_NOTE_LONG>
<SSR_REQUISITE_LONG></SSR_REQUISITE_LONG>
<SSR_COMPONENT>LAB</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Laboratory</SSR_COMPONENT_LOVDescr>
<OPTIONAL_SECTION>N</OPTIONAL_SECTION>
<OPTIONAL_SECTION_LOVDescr>Required</OPTIONAL_SECTION_LOVDescr>
<CLASS_COMPONENT_SECTIONS>
<CLASS_COMPONENT_SECTION>
<CLASS_NBR>1721</CLASS_NBR>
<CLASS_TYPE>E</CLASS_TYPE>
<CLASS_TYPE_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<CLASS_SECTION>2A</CLASS_SECTION>
<SSR_AUTO_ENRL_SECT>N</SSR_AUTO_ENRL_SECT>
<ENRL_STAT>O</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
<class meeting pattern>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
<SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
<SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
<SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
</class meeting pattern>
<CLASS_COMPONENT_SECTION>
<CLASS_NBR>1722</CLASS_NBR>
<CLASS_TYPE>E</CLASS_TYPE>
<CLASS_TYPE_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<CLASS_SECTION>2B</CLASS_SECTION>
<SSR_AUTO_ENRL_SECT>N</SSR_AUTO_ENRL_SECT>
<ENRL_STAT>O</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
<class meeting pattern>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
<SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
<SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
<SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
<SESSION_CODE>1</SESSION_CODE>
<SESSION_CODE_LOVDescr>Regular Academic Session</SESSION_CODE_LOVDescr>
<SSR_CLASSNOTE_LONG></SSR_CLASSNOTE_LONG>
<SSR_COMPONENT>LEC</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
<SSR_CRSE_ATTR_LONG></SSR_CRSE_ATTR_LONG>
<SSR_CRSE_TYPOFF_CD>FALL</SSR_CRSE_TYPOFF_CD>
<SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
<SSR_DROP_CONSENT>N</SSR_DROP_CONSENT>
<SSR_DROP_CONSENT_LOVDescr>No Special Consent Required</SSR_DROP_CONSENT_LOVDescr>
<SSR_GBL_NOTE_LONG></SSR_GBL_NOTE_LONG>
<SSR_REQUISITE_LONG></SSR_REQUISITE_LONG>
<START_DT>2011-08-30</START_DT>
<STRM>0670</STRM><STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
<SUBJECT>BIOLOGY</SUBJECT>
<SUBJECT_LOVDescr>Biology</SUBJECT_LOVDescr>
<UNITS_RANGE>3.5 units</UNITS_RANGE>
<WAIT_TOT>0</WAIT_TOT>
<WAIT_CAP>30</WAIT_CAP>
<CLASS_COMBINED_SECTION/>
<CLASS_COMPONENTS/>
<CLASS_COMPONENT>DIS</CLASS_COMPONENT>
<CLASS_COMPONENT_LOVDescr>Discussion</CLASS_COMPONENT_LOVDescr>
<OPTIONAL_SECTION>Y</OPTIONAL_SECTION>
<OPTIONAL_SECTION_LOVDescr>Optional</OPTIONAL_SECTION_LOVDescr>
<CLASS_COMPONENT_SECTIONS/>
<Class_NBR>1744</CLASS_NBR>
<Class_TYPE>E</CLASS_TYPE>
<Class_TYPE_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<Class_SECTION>1D</CLASS_SECTION>
<ENRL_STAT>0</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
<Class_MEETING_PATTERNS/>
<Class_MEETING_PATTERN>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
<SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
<SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
<SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
</CLASS_MEETING_PATTERN>
</CLASS_MEETING_PATTERNS>
<Class_NBR>1745</CLASS_NBR>
<Class_TYPE>E</CLASS_TYPE>
<Class_TYPE_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<Class_SECTION>2D</CLASS_SECTION>
<ENRL_STAT>0</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
<Class_MEETING_PATTERNS/>
<Class_MEETING_PATTERN>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
</CLASS_MEETING_PATTERN>
</CLASS_MEETING_PATTERNS>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_Component_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_Component_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_section>
<CLASS_MEETING_PATTERNS>
  <CLASS_MEETING_PATTERN>
    <SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
    <SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
    <SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
    <SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
    <SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
  </CLASS_MEETING_PATTERN>
  <CLASS_MEETING_PATTERN>
    <SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
    <SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
    <SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
    <SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
    <SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
  </CLASS_MEETING_PATTERN>
  <CLASS_MEETING_PATTERNS>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT_SECTIONS>
</CLASS_COMPONENT>
</CLASS_COMPONENTS>
</CLASS_MEETING_PATTERNS>
</CLASS_COMPONENT_SECTION>
</CLASS_SECTIONS>
</ITEM>
</ITEMS>
</COURSE_SHOP_CART>
</SHOPPING_CART>
</SCC_SC_SAVECART_RESP>
Remove Item

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SCC_SC_REMOVEITEM: remove multiple items from the cart</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation is used to remove multiple items from the cart and sends back the existing cart items if any along with their class section details.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation is used to remove multiple items from the cart and sends back the existing cart items if any along with their class section details.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>Identifies the incoming cart type (Course) and builds course shopping cart entities. Removes the requested cart item from the cart. Sends back the existing cart item if any along with their class section details.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response consists of existing cart item if any along with their class section details, else blank xml.</td>
</tr>
<tr>
<td>Error Conditions:</td>
<td>No particular error condition. Usually the UI will first show the existing items to the user through the Get Cart operation and the user will select item(s) from the result and these are passed as request to Remove Item. So no error condition applicable here.</td>
</tr>
</tbody>
</table>

**Input Message: SCC_SC_REMOVEITEM_REQ**

The following diagram shows the SCC_SC_REMOVEITEM_REQ message structure:

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:
The Request and Response xsds of SCC_SC_REMOVEITEM service operation will have the same structure since both represent the Items on course shopping cart. In the above diagram, SSR_CLASS_SECTION has been defined as the child of ITEM so this will reflect both in request as well as in the response xsd. The request xml need not include the SSR_CLASS_SECTION at all since it represents the class section details of the ITEM and hence it will only be part of the response xml.

**Figure 91: SCC_SC_REMOVEITEM_REQ Message Structure**

The following are the input parameters for the UI to pass to the service operation:

- *UserID
- *Password
- *SCC_SHOP_CART_TYPE
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number

Parameters marked with * are mandatory. User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.

SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.
EMPLID - the ID of the Student.
INSTITUTION – the Institution of the student.
Academic Career – the Career of the student.
Term – the Term of the student.
Class Number - the Class Number that the student is trying to remove.

The following is an example of the SCC_SC_REMOVEITEM_REQ message that the SCC_SC_REMOVEITEM service operation receives from an Enrollment application:
<?xml version="1.0"?>
<SCC_SC_REMOVEITEM_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
        <ITEM>
          <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
          <EMPLID>SR12030</EMPLID>
          <INSTITUTION>PSUNV</INSTITUTION>
          <ACAD_CAREER>UGRD</ACAD_CAREER>
          <STRM>0670</STRM>
          <CLASS_NBR>1737</CLASS_NBR>
          <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
          <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
          <GRADING_BASIS_ENRL></GRADING_BASIS_ENRL>
          <UNT_TAKEN>0</UNT_TAKEN>
          <INSTRUCTOR_ID></INSTRUCTOR_ID>
          <RQMNT_DESIGNTN_OPT></RQMNT_DESIGNTN_OPT>
          <START_DT></START_DT>
          <CRSE_COUNT>0</CRSE_COUNT>
          <ACAD_PROG></ACAD_PROG>
          <RELATE_CLASS_NBR_1>0</RELATE_CLASS_NBR_1>
          <RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
        </ITEM>
      </ITEMS>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SCC_SC_REMOVEITEM_REQ>

Output Message: SCC_SC_REMOVEITEM_RESP

When the Integration Broker receives the SCC_SC_REMOVEITEM_REQ message, it responds with the SCC_SC_REMOVEITEM_RESP message.

The following diagram shows the SCC_SC_REMOVEITEM_RESP message structure:

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Figure 12: SCC_SC_REMOVEITEM_RESP Message Structure

The following is an example of the SCC_SC_REMOVEITEM_RESP message that the SCC_SC_REMOVEITEM service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SCC_SC_REMOVEITEM_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
        <ITEM>
          <EMPLID>SR12030</EMPLID>
          <INSTITUTION>PSUNV</INSTITUTION>
          <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
          <ACAD_CAREER>UGRD</ACAD_CAREER>
          <ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
          <STRM>0670</STRM>
          <STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
          <CLASS_NBR>1718</CLASS_NBR>
          <CLASS_NBR_LOVDescr>CHEM 201 - Organic Chem I</CLASS_NBR_LOVDescr>
          <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
          <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
          <GRADING BASIS_ENRL>GRD</GRADING BASIS_ENRL>
          <GRADING BASIS ENRL_LOVDescr>Graded</GRADING BASIS ENRL_LOVDescr>
          <UNT_TAKEN>3</UNT_TAKEN>
          <INSTRUCTOR_ID></INSTRUCTOR_ID>
          <RQMNT DESIGNTN_OPT>N</RQMNT DESIGNTN_OPT>
          <START_DT></START_DT>
          <CRSE_COUNT>1</CRSE_COUNT>
          <ACAD_PROG>LAU</ACAD_PROG>
        </ITEM>
        <ITEM>...
        ...
      </ITEMS>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SCC_SC_REMOVEITEM_RESP>
```
<ACAD_PROG_LOVDescr>Liberal Arts Undergraduate</ACAD_PROG_LOVDescr>
<RELATE_CLASS_NBR_1>1721</RELATE_CLASS_NBR_1>
<RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
<SCELL_ADD_DTTM>01/09/2012 06:49:20.000000</SCCell_ADD_DTTM>
<QUANTITY>0</QUANTITY>
<RQMNT_DESIGNTN>DSGN</RQMNT_DESIGNTN>
<RQMNT_DESIGNTN_LOVDescr>Design Course</RQMNT_DESIGNTN_LOVDescr>
<ENRL_REQ_DETSTAT>P</ENRL_REQ_DETSTAT>
<DTTM_STAMP_SEC>01/09/2012 06:49:19.000000</DTTM_STAMP_SEC>
<SSR_VALIDATE>N</SSR_VALIDATE>
<SSR_VALIDATN_DTTM>

<CLASS_SECTIONS>
<ACAD_CAREER>UGRD</ACAD_CAREER>
<ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
<ASSOCIATED_CLASS>2</ASSOCIATED_CLASS>
<AVAILABLE_SEATS>100</AVAILABLE_SEATS>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
<CATALOG_NBR>201</CATALOG_NBR>
<CLASS_NBR>1718</CLASS_NBR>
<CLASS_SECTION>2</CLASS_SECTION>
<CLASS_TYPE>E</CLASS_TYPE>
<CLASS_TYPE_LOVDescr>Enrollment Section</CLASS_TYPE_LOVDescr>
<COMBINED_SECTION></COMBINED_SECTION>
<COMBINED_SECTION_LOVDescr></COMBINED_SECTION_LOVDescr>
<CONSENT>N</CONSENT>
<CONSENT_LOVDescr>No Special Consent Required</CONSENT_LOVDescr>
<CRSE_ID>003746</CRSE_ID>
<CRSE_ID_LOVDescr>Organic Chem I</CRSE_ID_LOVDescr>
<CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
<DESCR200>CHEM 201 - 2 Organic Chemistry I</DESCR200>
<DESCR_LONG></DESCR_LONG>
<EFFDT>2011-08-30</EFFDT>
<END_DT>2011-12-12</END_DT>
<ENRL_CAP>100</ENRL_CAP>
<ENRL_STATUS>0</ENRL_STATUS>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<ENRL_TOT>0</ENRL_TOT>
<GRADING_BASIS>GRD</GRADING_BASIS>
<GRADING_BASIS_LOVDescr>Graded</GRADING_BASIS_LOVDescr>
<INSTITUTION>PSUNV</INSTITUTION>
<INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
<INSTRUCTION_MODE>P</INSTRUCTION_MODE>
<INSTRUCTION_MODE_LOVDescr>In Person</INSTRUCTION_MODE_LOVDescr>
<LOCATION>PSCSHCD</LOCATION>
<LOCATION_DESCR>Hacienda</LOCATION_DESCR>
<MIN_ENRL>0</MIN_ENRL>
<PRINT_TOPIC>N</PRINT_TOPIC>
<SCHEDULE_PRINT>Y</SCHEDULE_PRINT>
<SCHEDULE_PRINT_LOVDescr>Yes</SCHEDULE_PRINT_LOVDescr>
<SESSION_CODE>1</SESSION_CODE>
<SESSION_CODE_LOVDescr>Regular Academic Session</SESSION_CODE_LOVDescr>
<SSR_CLASSNOTE_LONG></SSR_CLASSNOTE_LONG>
<SSR_COMPONENT>LEC</SSR_COMPONENT>
<SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
<SSR_CRSE_ATTR_LONG/></SSR_CRSE_ATTR_LONG>
<SSR_CRSE_TYPOFF_CD>FALL, SPR</SSR_CRSE_TYPOFF_CD>
<SSR_DATE_LONG>8/30/2011 - 12/12/2011</SSR_DATE_LONG>
<SSR_DROP_CONSENT>Y</SSR_DROP_CONSENT>
<SSR_DROP_CONSENT_LOVDescr>No Special Consent Required</SSR_DROP_CONSENT_LOVDescr>
<SSR_GBL_NOTE_LONG/></SSR_GBL_NOTE_LONG>
<SSR_REQUISITE_LONG/></SSR_REQUISITE_LONG>
<START_DT>2011-08-30</START_DT>
<STRM>0670</STRM>
<STRM_LOVDescr>2011 Fall</STRM_LOVDescr>
<SUBJECT>Chemistry</SUBJECT>
<SUBJECT_LOVDescr>Chemistry</SUBJECT_LOVDescr>
<UNITS_RANGE>3 units</UNITS_RANGE>
<WAIT_TOT>0</WAIT_TOT>
<WAIT_CAP>35</WAIT_CAP>

<CLASS_COMBINED_SECTION/>
<CLASS_COMPONENTS/>
<CLASS_COMPONENT>
<SSR_COMPONENT>LAB</SSR_COMPONENT>
<OPTIONAL_SECTION>N</OPTIONAL_SECTION>
<OPTIONAL_SECTION_LOVDescr>Required</OPTIONAL_SECTION_LOVDescr>

<CLASS_COMPONENT_SECTIONS/>
<CLASS_COMPONENT_SECTION>
<CLASS_NBR>1721</CLASS_NBR>
<CLASS_TYPE�� Enrollment Section</CLASS_TYPE>
<CLASS_SECTION>2A</CLASS_SECTION>
<SSR_AUTO_ENRL_SECT>N</SSR_AUTO_ENRL_SECT>
<ENRL_STAT>0</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>

<CLASS_MEETING_PATTERNS/>
<CLASS_MEETING_PATTERN>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
<SSR_Instr_LONG>Staff</SSR_Instr_LONG>
<SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
<SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>

</CLASS_MEETING_PATTERN>

</CLASS_MEETING_PATTERNS>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT>

<CLASS_COMPONENT>
<SSR_COMPONENT>LAB</SSR_COMPONENT>
<OPTIONAL_SECTION>N</OPTIONAL_SECTION>
<OPTIONAL_SECTION_LOVDescr>Required</OPTIONAL_SECTION_LOVDescr>

<CLASS_COMPONENT_SECTIONS/>
<CLASS_COMPONENT_SECTION>
<CLASS_NBR>1722</CLASS_NBR>
<CLASS_TYPE型 Enrollment Section</CLASS_TYPE>
<CLASS_SECTION>2B</CLASS_SECTION>
<SSR_AUTO_ENRL_SECT>N</SSR_AUTO_ENRL_SECT>
<ENRL_STAT>0</ENRL_STAT>
<ENRL_STATUS_DESCR>Open</ENRL_STATUS_DESCR>
<CAMPUS>MAIN</CAMPUS>
<CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>

<CLASS_MEETING_PATTERNS/>
<CLASS_MEETING_PATTERN>
<SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
<SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
<SSR_Instr_LONG>Staff</SSR_Instr_LONG>
<SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
<SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>

</CLASS_MEETING_PATTERN>

</CLASS_MEETING_PATTERNS>
</CLASS_COMPONENT_SECTION>
</CLASS_COMPONENT>
Validate Shopping Cart

Description
**Service:** SCC_SHOPPING_CART

**Operation:** SCC_SC_VALIDATE: validate shopping cart

**Summary:** This Service Operation is used to validate items before they can be checked out.

**Description:**
This Service Operation is used to validate items before they can be checked out. It can validate multiple items and send back the validation results of all the items. This operation checks for validation access first and then performs the validations using cobol engine, the same way as in the current system.

**Users:** Student

**Processing:**
Identifies the cart type (course) and builds course shopping cart entities. Validates user access to perform validation. Invokes cobol validation engine. Sends back the validation results.

**Output:**
Response consists of validation results which can be an informational message, for example ‘Ok to Add’ on successful validation or an error message.

**Error Conditions:**
If the user has no access to perform the validation, an error is sent back conveying the same.

---

**Input Message: SCC_SC_VALIDATE_REQ**

The following diagram shows the SCC_SC_VALIDATE_REQ message structure:

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:

---

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:
In the above diagram, SSR_CLASS_SECTION has been defined as a child of ITEM. But the request xml should not include this and should only include the elements as outlined in the following section. SSR_CLASS_SECTION just forms as part of request xsd since it is the same common xsd SCC_ENTITY_SHOP_CART that is referenced in SCC_SC_VALIDATE_REQ message.

**Figure 101: SCC_SC_VALIDATE_REQ Message Structure**

The following are the input parameters for the UI to pass to the service operation:

- *UserID
- *Password
- *SCC_SHOP_CART_TYPE
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number

The following is an example of the SCC_SC_VALIDATE_REQ message that the SCC_SC_VALIDATE service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SCC_SC_VALIDATE_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
    <SHOPPING_CART>
        <COURSE_SHOP_CART>
            <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
            <ITEMS>
                <ITEM>
                    <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
                    <EMPLID>SR12030</EMPLID>
                    <INSTITUTION>PSUNV</INSTITUTION>
                    <ACAD_CAREER>UGRD</ACAD_CAREER>
                    <STRM>0670</STRM>
                    <CLASS_NBR>1718</CLASS_NBR>
                    <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
                </ITEM>
            </ITEMS>
        </COURSE_SHOP_CART>
    </SHOPPING_CART>
</SCC_SC_VALIDATE_REQ>
```
<CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
<GRADING_BASIS_ENRL></GRADING_BASIS_ENRL>
<UNT_TAKEN>0</UNT_TAKEN>
<INSTRUCTOR_ID></INSTRUCTOR_ID>
<RQMNT_DESIGNTN_OPT></RQMNT_DESIGNTN_OPT>
<START_DT></START_DT>
<CRSE_COUNT>0</CRSE_COUNT>
<ACAD_PROG></ACAD_PROG>
<RELATE_CLASS_NBR_1>0</RELATE_CLASS_NBR_1>
<RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
</ITEM>
</ITEMS>
</COURSE_SHOP_CART>
</SHOPPING_CART>
</SCC_SC_VALIDATE_REQ>

Parameters marked with * are mandatory. User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.
EmplID - the ID of the Student.
Institution – the Institution of the student.
Academic Career – the Career of the student.
Term – the Term of the student.
Class Number - the Class Number that the student is trying to validate.

Output Message: SCC_SC_VALIDATE_RESP

When the Integration Broker receives the SCC_SC_VALIDATE_REQ message, it responds with the SCC_SC_VALIDATE_RESP message.

The following diagram shows the SCC_SC_VALIDATE_RESP message structure:

Figure 12: SCC_SC_VALIDATE_RESP Message Structure

The following is an example of the SCC_SC_VALIDATE_RESP message that the SCC_SC_VALIDATE service operation transmits to the Enrollment UI:

```
<?xml version="1.0"?>
<SCC_SC_VALIDATE_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
```
Checkout Shopping Cart

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SCC_SHOPPING_CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SCC_SC_CHECKOUT: checkout shopping cart</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation is used to Checkout (enroll) the cart items.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation is used to Checkout (enroll) the cart items. User can select multiple items and checkout all the items. The operation can handle multiple items and returns the result of all the items back to the UI. This service operation converts the shopping cart item into enrollment request, saves the enrollment request and enrolls the student in the class and finally deletes the classes from the shopping cart on successful enrollment. The operation returns the enrollment messages in the response.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>Identifies the cart type (course) and builds course shopping cart entities. Converts cart items into enrollment requests. Performs validations as detailed in the Description section above. Invokes enrollment engine. Deletes the cart item on successful enrollment of an item. Reads the message log table to identify the messages posted by the enrollment engine. Populates the response message with the messages from the log table. Sends back the messages to the UI for all items.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response consists of the messages generated by the enrollment engine.</td>
</tr>
</tbody>
</table>
### Error Conditions:

<table>
<thead>
<tr>
<th>Validations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Class Nbr validation – checks if the passed Class Nbr is valid for the</td>
</tr>
<tr>
<td>Institution and Term.</td>
</tr>
<tr>
<td>• Enrollment requests should only be created when:</td>
</tr>
<tr>
<td>• Student is term activated and eligible to enroll.</td>
</tr>
<tr>
<td>• Student is not withdrawn or cancelled.</td>
</tr>
<tr>
<td>• Student has an active program for the term.</td>
</tr>
<tr>
<td>• Users can submit enrollment transactions only if their UserId is</td>
</tr>
<tr>
<td>associated to the Permission List specified in Setup SACR &gt; Security &gt;</td>
</tr>
<tr>
<td>Secure Student Administration &gt; Permission List &gt; Self-Service</td>
</tr>
<tr>
<td>Enrollment Perm List (See: PeopleSoft Campus Solutions 9.0 Application</td>
</tr>
<tr>
<td>Fundamentals PeopleBook, Securing Student Records, Setting Up</td>
</tr>
<tr>
<td>Enrollment Security for Self-Service Enrollment)</td>
</tr>
<tr>
<td>• If online appointment checking is enforced, check if student has</td>
</tr>
<tr>
<td>enrollment appointment for the selected session.</td>
</tr>
<tr>
<td>• Self-service: Enrollment requests are always built with EMPLID =</td>
</tr>
<tr>
<td>UserId’s EMPLID.</td>
</tr>
<tr>
<td>• Check Career Rules – checks if the student is allowed to take this</td>
</tr>
<tr>
<td>course, based on career rules. For example, an undergraduate student</td>
</tr>
<tr>
<td>should be prevented from putting Law classes in their cart.</td>
</tr>
<tr>
<td>• OEE Edits - If the class selected is an Open Entry/Open Exit class,</td>
</tr>
<tr>
<td>check if the student’s program allows enrollment to this type of class.</td>
</tr>
<tr>
<td>• Check Last Date to Enroll – if the last date to enroll for this class</td>
</tr>
<tr>
<td>has passed, don’t allow the class to be added to the cart.</td>
</tr>
<tr>
<td>• Check duplicate enrollment – if the student is already enrolled in the</td>
</tr>
<tr>
<td>class, don’t allow the class to be added to the cart.</td>
</tr>
<tr>
<td>• System edits on enrollment request entity hierarchy – Required field,</td>
</tr>
<tr>
<td>Translate, Yes/No, Prompt.</td>
</tr>
</tbody>
</table>

**Input Message: SCC_SC_CHECKOUT_REQ**

The following diagram shows the SCC_SC_CHECKOUT_REQ message structure:

```
<schema>

targetNamespace: http://xmlns.oracle.com/Enterprise/OM/services

<include>

schemaLocation: SCC_ENTITY_SHOP_CART.V1.xsd
```

The structure of SCC_ENTITY_SHOP_CART.V1.xsd is shown below:
In the above diagram, SSR_CLASS_SECTION has been defined as a child of ITEM. But the request XML should not include this and should only include the elements as outlined in the following. SSR_CLASS_SECTION just forms as part of request XSD since it is the same common XSD SCC_ENTITY_SHOP_CART that is referenced in SCC_SC_CHECKOUT_REQ message.

Figure 111: SCC_SC_CHECKOUT_REQ Message Structure

The following are the input parameters for the UI to pass to the service operation:

- UserID
- *Password
- *SCC_SHOP_CART_TYPE
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number

The following is an example of the SCC_SC_CHECKOUT_REQ message that the SCC_SC_CHECKOUT service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SCC_SC_CHECKOUT_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
        <ITEM>
          <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
          <EMPLID>SR12030</EMPLID>
          <INSTITUTION>PSUNV</INSTITUTION>
        </ITEM>
      </ITEMS>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SCC_SC_CHECKOUT_REQ>
```
<ACAD_CAREER>UGRD</ACAD_CAREER>
<STRM>0670</STRM>
<class_nbr>1718</class_nbr>
<wait_list_okay>N</wait_list_okay>
<class_prmsn_nbr>0</class_prmsn_nbr>
<grading_basis_enrl></grading_basis_enrl>
<unt_taken>0</unt_taken>
<instructor_id></instructor_id>
<rqmnt_designtn_opt></rqmnt_designtn_opt>
<start_dt></start_dt>
<crse_count></crse_count>
<acad_prog></acad_prog>
<relate_class_nbr_1>0</relate_class_nbr_1>
<relate_class_nbr_2>0</relate_class_nbr_2>
</item>
</ITEM>
</COURSE_SHOP_CART>
</SHOPPING_CART>
</SCC_SC_CHECKOUT_REQ>

Parameters marked with * are mandatory.

Notes:
User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.

SCC_SHOP_CART_TYPE – The UI should set the value to COURSE.
EmplID - the ID of the Student
Institution – the Institution of the student.
Academic Career – the Career of the student.
Term – the Term of the student.
Class Number - the Class Number that the student is trying to enroll.

Output Message: SCC_SC_CHECKOUT_RESP

When the Integration Broker receives the SCC_SC_CHECKOUT_REQ message, it responds with the SCC_SC_CHECKOUT_RESP message.

The following diagram shows the SCC_SC_CHECKOUT_RESP message structure:

Figure 12: SCC_SC_CHECKOUT_RESP Message Structure
The following is an example of the SCC_SC_CHECKOUT_RESP message that the SCC_SC_CHECKOUT service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SCC_SC_CHECKOUT_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <MSGS>
      <MSG>
        <ID>14690-563</ID>
        <DESCR>This class has been added to your schedule.</DESCR>
        <MESSAGE_SEVERITY>I</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
    </MSGS>
  </detail>
</SCC_SC_CHECKOUT_RESP>
```
Enrollment Web Service Operations

Service Name: **SSR_ENROLLMENT**

Enrollment web service offers functionality in the following areas:

- **Add Enrollment**: Enrolls the student in one or more class.
- **Drop Enrollment**: Drops the student from one or more class.
- **Edit Enrollment**: Allows the student to edit a class in which the student is enrolled.
- **Swap Enrollment**: Allows the student to swap a class in which the student is enrolled with another class.
- **Enrollment Validation**: Allows the student to validate the course shopping cart item prior to checkout.
- **Get Enrollment**: Once a student is enrolled for a set of class sections in a specified term, the classes can be retrieved to show the current StudyList.
- **Retrieve Class Deadlines**: Allows the student to view class enrollment deadline information.

### Add Enrollment

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_ENROLLMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_ADD_ENROLLMENT: enroll student in a one or more class</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to enroll in a one or more class.</td>
</tr>
</tbody>
</table>
Description: This Service Operation enrolls the student in a one or more class. The enrollment options to be provided will be derived from SSR_STUDENT_ENROLLMENT_OPTIONS service operation. The operation allows enrollment in multiple classes. Enrollment request action should be ‘Enroll’.

Validations performed for Add Enrollment:

- Enrollment requests should only be created when:
  - Student is term activated and eligible to enroll.
  - Student is not withdrawn or cancelled.
  - Student has an active program for the term.
- Users can submit enrollment transactions only if their UserId is associated to the Permission List specified in Setup SACR > Security > Secure Student Administration > Permission List > Self-Serv Enrollment Perm List.
- If online appointment checking is enforced, check if student has enrollment appointment for the selected class session.
- Self-service: enrollment requests are always built with EMPLID = UserId’s EMPLID.
- Check Career Rules – checks if the student is allowed to take this course, based on career rules. For example, an undergraduate student should be prevented from putting Law classes in their cart.
- OEE Edits - If the class selected is an Open Entry/Open Exit class, check if the student’s program allows enrollment to this type of class.
- Check Last Date to Enroll – if the last date to enroll for this class has passed, don’t allow the class to be added to the cart.
- Check duplicate enrollment – if the student is already enrolled in the class, don’t allow the class to be added to the cart.
- Create enrollment request detail with enrollment action = ‘E’

Users: Student

Processing: This service operation performs the following steps:

- Verify required input parameters exist.
- Performs the above mentioned validations on the input parameters.
- Creates an entry in Enrollment Request data modal and invokes the COBOL process.
- Responds with the Info/error messages.

Output: Response message contains Information/error messages in enrollment

Error Conditions: The service operation results in error in the following conditions.

- Invalid input parameters
- Any of the validations mentioned above fail.
- COBOL process results in error.

Input Message: SSR_ADD_ENROLLMENT_REQ

The following diagram shows the input message structure:
Figure 12: SSR_ADD_ENROLLMENT_REQ Message Structure

Input parameters:

- User ID
- *Password
- EMPLID
- INSTITUTION
- *Academic Career
- *Term
- *Class Number
- Related Class Number 1
- Related Class Number 2
- Change to Class nbr
- Drop this class if enrolled
- Wait List Okay
- Grading Basis
- Units Taken
- Course Count
- Academic Program
- Start Date
- Class Permission Nbr
- Instructor ID
- Requirement Designation Option
- *Enrollment request Action

Parameters marked with * are mandatory.

- User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
- EmplID - the ID of the Student.
- Class Number - the Class Number that the student is trying to enroll.
- Related Class Number 1 & 2 – Classes that are related to the selected Class number.
- Change to class nbr & drop this class if enrolled – are used in case of swap operation.
- Wait list okay – Y/N flag to be set by user if enabled.
- Grading Bases – Can be selected from LOV by user if enabled.
- Units taken – Number of units, can be selected by user if enabled.
• Academic Program - Can be selected from LOV by user if enabled.
• Start Date - if enabled, is selected by user.
• Class Permission Nbr – Permission number to be entered by user if enabled.
• Instructor Id - Can be selected from LOV by user if enabled.
• Requirement Designation Option - Y/N flag to be set by user if enabled.
• Enrollment request Action – to be set as “E” (Enroll).

The following is an example of the SSR_ADD_ENROLLMENT_REQ message that the SSR_ADD_ENROLLMENT service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_ADD_ENROLLMENT_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <ENROLL_REQ_HEADER>
    <ENRL_REQUEST_ID></ENRL_REQUEST_ID>
    <OPRID></OPRID>
    <ENRL_REQ_PROC_ST></ENRL_REQ_PROC_ST>
    <ENRL_REQ_SOURCE></ENRL_REQ_SOURCE>
    <PROCESS_INSTANCE>0</PROCESS_INSTANCE>
    <DTTM_STAMP_SEC></DTTM_STAMP_SEC>
    <ENROLL_REQUEST_DETAILS>
      <ENROLL_REQUEST.Detail>
        <ENRL_REQUEST_ID></ENRL_REQUEST_ID>
        <ENRL_REQ_DETL_SEQ>0</ENRL_REQ_DETL_SEQ>
        <EMPLID>SR13454</EMPLID>
        <ACAD_CAREER>UGRD</ACAD_CAREER>
        <INSTITUTION>PSUNV</INSTITUTION>
        <STRM>0530</STRM>
        <CLASS_NBR>1650</CLASS_NBR>
        <ENRL_REQ_ACTION>E</ENRL_REQ_ACTION>
        <ENRL_ACTION_REASON></ENRL_ACTION_REASON>
        <ENRL_ACTION_DT></ENRL_ACTION_DT>
        <UNT_TAKEN>3</UNT_TAKEN>
        <UNT_EARNED>0</UNT_EARNED>
        <CRSE_COUNT>1</CRSE_COUNT>
        <REPEAT_CODE></REPEAT_CODE>
        <CRSE_GRADE_INPUT></CRSE_GRADE_INPUT>
        <GRADING_BASIS_ENRL>GRD</GRADING_BASIS_ENRL>
        <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
        <CLASS_NBR_CHG_TO>0</CLASS_NBR_CHG_TO>
        <DROP_CLASS_IF_ENRL>0</DROP_CLASS_IF_ENRL>
        <CHG_TO_WL_NUM>0</CHG_TO_WL_NUM>
        <RELATE_CLASS_NBR_1>1651</RELATE_CLASS_NBR_1>
        <RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
        <OVRD_CLASS_LIMIT></OVRD_CLASS_LIMIT>
        <OVRD_GRADING_BASIS></OVRD_GRADING_BASIS>
        <OVRD_CLASS_UNITS></OVRD_CLASS_UNITS>
        <OVRD_UNIT_LOAD></OVRD_UNIT_LOAD>
        <OVRD_CLASS_LINKS></OVRD_CLASS_LINKS>
        <OVRD_CLASS_PRMSN></OVRD_CLASS_PRMSN>
        <OVRD_REQUISITES></OVRD_REQUISITES>
        <OVRD_TIME_CNFLCT></OVRD_TIME_CNFLCT>
        <OVRD_CAREER></OVRD_CAREER>
        <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
        <OVRD_ENRL_ACTN_DT></OVRD_ENRL_ACTN_DT>
        <OVRD_RQMT_DESIG></OVRD_RQMT_DESIG>
        <OVRD_SRVC_INDIC></OVRD_SRVC_INDIC>
        <OVRD_APPT></OVRD_APPT>
        <INSTRUCTOR_ID></INSTRUCTOR_ID>
        <ENRL_REQ_DETL_STAT></ENRL_REQ_DETL_STAT>
        <RQMT_DESIGNTN></RQMT_DESIGNTN>
        <RQMT_DESIGNTN_OPT></RQMT_DESIGNTN_OPT>
        <RQMT_DESIGNTN_GRD></RQMT_DESIGNTN_GRD>
      </ENROLL_REQUEST_Detail>
    </ENROLL_REQUEST_DETAILS>
  </ENROLL_REQ_HEADER>
</SSR_ADD_ENROLLMENT_REQ>
```
<TSCRPT_NOTE_ID></TSCRPT_NOTE_ID>
<TSCRPT_NOTE_EXISTS></TSCRPT_NOTE_EXISTS>
<OPRID></OPRID>
<DTTM_STAMP_SEC></DTTM_STAMP_SEC>
<START_DT></START_DT>
<ACAD_PROG>CEOEE</ACAD_PROG>
<ACAD_PROG>CCEOEE</ACAD_PROG>
<SCC_ENTITY_INST_ID>ENTITY_INST_1</SCC_ENTITY_INST_ID>
</ENROLL_REQUEST_DETAIL>
<ENROLL_REQUEST_DETAIL>
<ENRL_REQUEST_ID></ENRL_REQUEST_ID>
<ENRL_REQ_DETL_SEQ>0</ENRL_REQ_DETL_SEQ>
<EMPLID>SR13454</EMPLID>
<ACAD_CAREER>UGRD</ACAD_CAREER>
<INSTITUTION>PSUNV</INSTITUTION>
<STRM>0530</STRM>
<class_nbr>1111</class_nbr>
<ENRL_REQ_ACTION>E</ENRL_REQ_ACTION>
<ENRL_ACTION_REASON></ENRL_ACTION_REASON>
<ENRL_ACTION_DT></ENRL_ACTION_DT>
<unt_taken>3</unt_taken>
<unt_earned>0</unt_earned>
<crse_count>1</crse_count>
<repeat_code></repeat_code>
<crse_grade_input></crse_grade_input>
<grading_basis_enrl>GRD</grading_basis_enrl>
<class_prmsn_nbr>0</class_prmsn_nbr>
<class_nbr_chg_to>0</class_nbr_chg_to>
<drop_class_if_enrl></drop_class_if_enrl>
<chg_to_wl_num>0</chg_to_wl_num>
<relate_class_nbr_1>0</relate_class_nbr_1>
<relate_class_nbr_2>0</relate_class_nbr_2>
<ovrd_class_limit></ovrd_class_limit>
<ovrd_grading_basis></ovrd_grading_basis>
<ovrd_class_units></ovrd_class_units>
<ovrd_unit_load></ovrd_unit_load>
<ovrd_class_links></ovrd_class_links>
<ovrd_class_prmsn></ovrd_class_prmsn>
<ovrd_requisites></ovrd_requisites>
<ovrd_time_cnflct></ovrd_time_cnflct>
<ovrd_career></ovrd_career>
<wait_list_okay>N</wait_list_okay>
<ovrd_enrl_actn_dt></ovrd_enrl_actn_dt>
<ovrd_rqmnt_desig></ovrd_rqmnt_desig>
<ovrd_svrc_indic></ovrd_svrc_indic>
<ovrd_appt></ovrd_appt>
<instructor_id>CC0011</instructor_id>
<enrl_req_detl_stat></enrl_req_detl_stat>
<rqmnt_designtn></rqmnt_designtn>
<rqmnt_designtn_opt></rqmnt_designtn_opt>
<rqmnt_designtn_grd></rqmnt_designtn_grd>
<tscrpt_note_id></tscrpt_note_id>
<tscrpt_note_exists></tscrpt_note_exists>
<oprid></oprid>
<dttm_stamp_sec></dttm_stamp_sec>
<start_dt></start_dt>
<acad_prog>CEOEE</acad_prog>
<acad_prog>CCEOEE</acad_prog>
<sc_entity_inst_id>ENTITY_INST_2</sc_entity_inst_id>
</enroll_request_detail>
<enroll_request_details>
</ssr_add_enrollment_req>
Output Message: **SSR_ADD_ENROLLMENT_RESP**

When the Integration Broker receives the SSR_ADD_ENROLLMENT_REQ message, it responds with the SSR_ADD_ENROLLMENT_RESP message the structure of which is shown below.

```xml
<?xml version="1.0"?>
<SSR_ADD_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <MSGS>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>This class has been added to your schedule.</DESCR>
        <MESSAGE_SEVERITY>M</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_1</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>Unable to add this class - term maximum exceeded. Adding this class would exceed the maximum number of units or courses allowed for this term.</DESCR>
        <MESSAGE_SEVERITY>E</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_2</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
    </MSGS>
  </detail>
</SSR_ADD_ENROLLMENT_RESP>
```

Figure 13: SSR_ADD_ENROLLMENT_RESP Message Parameters

The following is an example of the SSR_ADD_ENROLLMENT_RESP message that the SSR_ADD_ENROLLMENT service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_ADD_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <MSGS>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>This class has been added to your schedule.</DESCR>
        <MESSAGE_SEVERITY>M</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_1</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>Unable to add this class - term maximum exceeded. Adding this class would exceed the maximum number of units or courses allowed for this term.</DESCR>
        <MESSAGE_SEVERITY>E</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_2</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
    </MSGS>
  </detail>
</SSR_ADD_ENROLLMENT_RESP>
```
Drop Enrollment

Description

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_ENROLLMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_DROP_ENROLLMENT: drop student from a one or more class.</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to drop enrollment for a one or more class.</td>
</tr>
</tbody>
</table>
| Description:             | This Service Operation drops the student from a one or more class. The enrollment options to be provided will be derived from SSR_STUDENT_ENROLLMENT_OPTIONS service operation. The operation allows student to be dropped from multiple classes. Enrollment request action from service tester should be ‘Drop’. Validations performed for Drop Enrollment

  • Enrollment requests should only be created when:
    • Student is term activated and eligible to enroll.
    • Student is not withdrawn or cancelled.
    • Student has an active program for the term.
  • Users can submit enrollment transactions only if their UserId is associated to the Permission List specified in Setup SACR > Security > Secure Student Administration > Permission List > Self-Serv Enrollment Perm List.
  • If online appointment checking is enforced, check if student has enrollment appointment for the selected class’ session.
  • Self-service: enrollment requests are always built with EMPLID = UserId’s EMPLID.
  • Classes that may be dropped are classes that have not been graded, have an enrollment status of enrolled and the enrollment status reason is not WDRW or DROP, or the enrollment status of waitlisted.
  • Create enrollment request detail with enrollment action = ‘D’. |
| Users:                   | Student |
| Processing:              | This service operation performs the following steps: |

  • Verify required input parameters exist.
  • Performs the above mentioned validations on the input parameters.
  • Creates an entry in Enrollment Request data modal and invokes the COBOL process.
  • Responds with the Info/error messages. |
| Output:                  | Response message contains Information/error messages occurred in enrollment |
| Error Conditions:        | The service operation results in error in the following conditions. |

  • Invalid input parameters.
  • Any of the validations mentioned above fail.
  • COBOL process results in error. |

Input Message: SSR_DROP_ENROLLMENT_REQ

The following diagram shows the input message structure:
Figure 14: SSR_DROP_ENROLLMENT_REQ Message Structure

Input parameters:

- *User ID
- *Password
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number
- Related Class Number 1
- Related Class Number 2
- Change to Class nbr
- Drop this class if enrolled
- Wait List Okay
- Grading Basis
- Units Taken
- Course Count
- Academic Program
- Start Date
- Class Permission Nbr
- Instructor ID
- Requirement Designation Option
- *Enrollment request Action

Parameters marked with * are mandatory.
The following is an example of the SSR_DROP_ENROLLMENT_REQ message that the SSR_DROP_ENROLLMENT service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_DROP_ENROLLMENT_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <ENROLL_REQUEST_ID/>
  <OPRID/>
  <ENRL_REQ_PROC_ST/>
  <ENRL_REQ_SOURCE/>
  <PROCESS_INSTANCE/>
  <DTTM_STAMP_SEC/>
  <ENROLL_REQUEST_DETAILS>
    <ENROLL_REQUEST_DETAIL>
      <ENRL_REQUEST_ID/>
      <ENRL_REQ_DETL_SEQ>0</ENRL_REQ_DETL_SEQ>
      <EMPLID>SR13456</EMPLID>
      <ACAD_CAREER>UGRD</ACAD_CAREER>
      <INSTITUTION>PSUNV</INSTITUTION>
      <STRM>0530</STRM>
      <CLASS_NBR>1237</CLASS_NBR>
      <ENRL_REQ_ACTION>D</ENRL_REQ_ACTION>
      <ENRL_ACTION_REASON/>
      <ENRL_ACTION_DT/>
      <UNT_TAKEN>0</UNT_TAKEN>
      <UNT_EARNED>0</UNT_EARNED>
      <CRSE_COUNT>0</CRSE_COUNT>
      <REPEAT_CODE/>
      <CRSE_GRADE_INPUT/>
      <GRADING_BASIS_ENRL>GRD</GRADING_BASIS_ENRL>
      <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
      <CLASS_NBR_CHG_TO/>
      <DROP_CLASS_IF_ENRL>0</DROP_CLASS_IF_ENRL>
      <CHG_TO_WL_NUM>0</CHG_TO_WL_NUM>
      <RELATE_CLASS_NBR_1>0</RELATE_CLASS_NBR_1>
      <RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
      <OVRD_CLASS_LIMIT/>
      <OVRD_GRADING_BASIS/>
      <OVRD_CLASS_UNITS/>
      <OVRD_UNIT_LOAD/>
      <OVRD_CLASS_LINKS/>
      <OVRD_PREREQS/>
      <OVRD_TIME_CNFLCT/>
      <OVRD_CAREER/>
      <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
      <OVRD_ENRL_ACTN_DT/>
      <OVRD_RQMNT_DESIG/>
      <OVRD_SRVC_INDIC/>
      <OVRD_APPT/>
      <INSTRUCTOR_ID/>
      <ENRL_REQ_DETL_STAT/>
      <RQMNT_DESIGNTN/>
      <RQMNT_DESIGNTN_OPT/>
      <RQMNT_DESIGNTN_GRD/>
      <TSCRPT_NOTE_ID/>
      <TSCRPT_NOTE_EXISTS/>
      <OPRID/>
      <DTTM_STAMP_SEC/>
      <START_DT/>
      <ACAD_PROG>CEOECE</ACAD_PROG>
      <SCC_ENTITY_INST_ID>ENTITY_INST_1</SCC_ENTITY_INST_ID>
  </ENROLL_REQUEST_DETAIL>
</ENROLL_REQUEST_DETAILS>
</SSR_DROP_ENROLLMENT_REQ>
```
Output Message: SSR_DROP_ENROLLMENT_RESP

When the Integration Broker receives the SSR_DROP_ENROLLMENT_REQ message, it responds with the SSR_DROP_ENROLLMENT_RESP message the structure of which is shown below.
Figure 15: SSR_DROP_ENROLLMENT_RESP Message Parameters

The following is an example of the SSR_DROP_ENROLLMENT_RESP message that the SSR_DROP_ENROLLMENT service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_DROP_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <MSGS>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>This class has been removed from your schedule</DESCR>
        <MESSAGE_SEVERITY>M</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_1</SCC_ENTITY_INST_ID>
            <PROPNAME/>
          </PROP>
        </PROPS>
      </MSG>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>Unable to Drop this class</DESCR>
        <MESSAGE_SEVERITY>E</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_2</SCC_ENTITY_INST_ID>
            <PROPNAME/>
          </PROP>
        </PROPS>
      </MSG>
    </MSGS>
  </detail>
</SSR_DROP_ENROLLMENT_RESP>
```

Edit Enrollment

Description
<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_ENROLLMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_EDIT_ENROLLMENT: edit a class</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to edit a class in which the student is enrolled.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation allows the student to edit a class in which the student is enrolled. The enrollment options to be provided will be derived from SSR_STUDENT_ENROLLMENT_OPTIONS service operation. Enrollment request action from service tester should be ‘Normal Maintenance’. If user update involves any related components then it will create a “swap” transaction, otherwise creates a normal maintenance (edit) transaction. Validations performed for Edit Enrollment:</td>
</tr>
<tr>
<td></td>
<td>• Enrollment requests should only be created when:</td>
</tr>
<tr>
<td></td>
<td>• Student is term activated and eligible to enroll.</td>
</tr>
<tr>
<td></td>
<td>• Student is not withdrawn or cancelled.</td>
</tr>
<tr>
<td></td>
<td>• Student has an active program for the term.</td>
</tr>
<tr>
<td></td>
<td>• Users can submit enrollment transactions only if their Userld is associated to the Permission List specified in Setup SACR &gt; Security &gt; Secure Student Administration &gt; Permission List &gt; Self-Serv Enrollment Perm List.</td>
</tr>
<tr>
<td></td>
<td>• If online appointment checking is enforced, check if student has enrollment appointment for the selected class’ session.</td>
</tr>
<tr>
<td></td>
<td>• Self-service: enrollment requests are always built with EMPLID = Userld’s EMPLID.</td>
</tr>
<tr>
<td></td>
<td>• Classes that may be edited are classes that have not been graded, have an enrollment status of enrolled and the enrollment status reason is not WDRW or DROP, or the enrollment status of waitlisted.</td>
</tr>
<tr>
<td></td>
<td>• Classes with updateable attributes –</td>
</tr>
<tr>
<td></td>
<td>• Student-option Grading Basis – If the class is set up to allow students to choose their own grading basis, the student is prompted for their selection at this point.</td>
</tr>
<tr>
<td></td>
<td>• Units:</td>
</tr>
<tr>
<td></td>
<td>• If the class selected is a variable-unit course, then prompt the user to make a selection. Valid range is anything between min and max, with unit increments determined by setup in the self service configuration. Default is the minimum units.</td>
</tr>
</tbody>
</table>
|                  |       • If the class is not a variable unit course, default the cart’s UNT_TAKEN field to Assoc_UnitMin and do not allow input.
Description:

• Instructor Edit – depending on appclass
  SSR_STUDENT_RECORDS:SR_ClassData:ClassInfo

  Assoc_InstrEdit:
  • ‘C’ (Class Instructor) - prompt = INSTR_IND_C_VW
  • ‘I’ (Instructor Advisor) – using
    CLASS_TBL.ACAD_ORG, look up
    ACAD_ORG_TBL.INSTR_EDIT
    • If ‘P’, prompt = INSTR_IND_1_VW
    • If ‘I’, gets complicated: see code in review page
      SSR_CLS_DTLOPT’s Activate code (look for
      the string “/* Instructor Edit */”),
  • ‘N’ – no choice, do not prompt for input

• Requirement Designation - check if the student selected a class
  with an associated Requirement Designation (appclass
  SSR_STUDENT_RECORDS:SR_ClassData:ClassInfo property

  Assoc_ReqDes
  • If the field is not empty, copy its value to the cart’s
    RQMNT_DESIGNTN.
  • Look up the definition for the Requirement Designation in
    RQMNT_DESIG_TBL.
  • If RQMNT_DESIG_TBL .AT_STDNT_OPTION = ‘Y’,
    it means the student can choose if they want to enroll in
    this RD or not. Allow the student to make their election,
    default the cart’s RQMNT_DESIGNTN_OPT to ‘N’.
  • If RQMNT_DESIG_TBL .AT_STDNT_OPTION = ‘N’,
    the student has no choice but to take this RD. Set the
    cart’s RQMNT_DESIGNTN_OPT to ‘Y’ and do not
    prompt the student.

• Classes with related sections.

• Enrollment Request Detail creation:
  • If the user’s update involves selecting a different related
    component section, create a swap transaction
    • Enrollment action = ‘S’
    • Use the current enrollment class nbr for both
      ENRL_REQ_DETAIL.CLASS_NBR and
      ENRL_REQ_DETAIL.CLASS_NBR_CHG_TO
    • Supply
      ENRL_REQ_DETAIL.RELATE_CLASS_NBR_(1 or 2)
    • If the user changed Grading Basis, Units, Instructor or
      Requirement Designation
      • Enrollment action = ‘M’

Users:

Processing:

This service operation performs the following steps:

• Verify required input parameters exist.
• Performs the above mentioned validations on the input parameters.
• Creates an entry in Enrollment Request data modal and invokes the COBOL
  process.
• Responds with the Info/error messages.

Output:

Response message contains Information/error messages occurred in enrollment

Error Conditions:

The service operation results in error in the following conditions.
• Invalid input parameters.
• Any of the validations mentioned above fail.
• COBOL process results in error.
**Input Message: SSR_EDIT_ENROLLMENT_REQ**

The following diagram shows the input message structure:

![Diagram showing input message structure](image)

**Figure 16: SSR_EDIT_ENROLLMENT_REQ Message Structure**

Input parameters:

- *User ID
- *Password
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number
- Related Class Number 1
- Related Class Number 2
- Change to Class nbr
- Drop this class if enrolled
- Wait List Okay
- Grading Basis
- Units Taken
- Course Count
- Academic Program
- Start Date
- Class Permission Nbr
- Instructor ID
- Requirement Designation Option
- *Enrollment request Action

Parameters marked with * are mandatory.

- User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
- EmplID - the ID of the Student.
• Class Number - the Class Number that the student is trying to update.
• Related Class Number 1 & 2 – Classes that are related to the selected Class number.
• Change to class nbr & drop this class if enrolled – are used in case of swap operation.
• Wait list okay – Y/N flag to be set by user if enabled.
• Grading Bases – Can be selected from LOV by user if enabled.
• Units taken – Number of units, can be selected by user if enabled.
• Academic Program - Can be selected from LOV by user if enabled.
• Start Date - if enabled, selected by user.
• Class Permission Nbr – Permission number to be entered by user if enabled.
• Instructor Id - Can be selected from LOV by user if enabled.
• Requirement Designation Option - Y/N flag to be set by user if enabled.
• Enrollment request Action – to be set as “M” (Normal Maintenance).

The following is an example of the SSR_EDIT_ENROLLMENT_REQ message that the SSR_EDIT_ENROLLMENT service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_EDIT_ENROLLMENT_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <ENROLL_REQ_HEADER>
    <ENRL_REQUEST_ID></ENRL_REQUEST_ID>
    <OPRID></OPRID>
    <ENRL_REQ_PROC_ST></ENRL_REQ_PROC_ST>
    <ENRL_REQ_SOURCE></ENRL_REQ_SOURCE>
    <PROCESS_INSTANCE>0</PROCESS_INSTANCE>
    <DTTM_STAMP_SEC></DTTM_STAMP_SEC>
  </ENROLL_REQ_HEADER>
  <ENROLL_REQUEST_DETAILS>
    <ENROLL_REQUEST_DETL>
      <ENRL_REQUEST_ID></ENRL_REQUEST_ID>
      <ENRL_REQ_DETL_SEQ>0</ENRL_REQ_DETL_SEQ>
      <EMPLID>SR13456</EMPLID>
      <ACAD_CAREER>UGRD</ACAD_CAREER>
      <INSTITUTION>PSUNV</INSTITUTION>
      <STRM>0530</STRM>
      <CLASS_NBR>1237</CLASS_NBR>
      <ENRL_REQ_ACTION>M</ENRL_REQ_ACTION>
      <ENRL_ACTION_REASON></ENRL_ACTION_REASON>
      <ENRL_ACTION_DT></ENRL_ACTION_DT>
      <UNT_TAKEN>3</UNT_TAKEN>
      <UNT_EARNED></UNT_EARNED>
      <CRSE_COUNT>1</CRSE_COUNT>
      <REPEAT_CODE></REPEAT_CODE>
      <CRSE_GRADE_INPUT></CRSE_GRADE_INPUT>
      <GRADING_BASIS_ENRL>GRD</GRADING_BASIS_ENRL>
      <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
      <CLASS_NBR_CHG_TO>0</CLASS_NBR_CHG_TO>
      <DROP_CLASS_IF_ENRL>1111</DROP_CLASS_IF_ENRL>
      <CHG_TO_WL_NUM>0</CHG_TO_WL_NUM>
      <RELATE_CLASS_NBR_1>0</RELATE_CLASS_NBR_1>
      <RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
      <OVRD_CLASS_LIMIT></OVRD_CLASS_LIMIT>
      <OVRD_GRADING_BASIS></OVRD_GRADING_BASIS>
      <OVRD_CLASS_UNITS></OVRD_CLASS_UNITS>
      <OVRD_UNIT_LOAD></OVRD_UNIT_LOAD>
      <OVRD_CLASS_LINKS></OVRD_CLASS_LINKS>
      <OVRD_CLASS_PRMSN></OVRD_CLASS_PRMSN>
      <OVRD_REQUISITES></OVRD_REQUISITES>
      <OVRD_TIME_CNFLCT></OVRD_TIME_CNFLCT>
      <OVRD_CAREER></OVRD_CAREER>
      <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
      <OVRD_ENRL_ACTN_DT></OVRD_ENRL_ACTN_DT>
      <OVRD_RQMTNDESIG></OVRD_RQMTNDESIG>
      <OVRD_SRVC_INDIC></OVRD_SRVC_INDIC>
    </ENROLL_REQUEST_DETL>
  </ENROLL_REQUEST_DETAILS>
</SSR_EDIT_ENROLLMENT_REQ>
```
When the Integration Broker receives the SSR_EDIT_ENROLLMENT_REQ message, it responds with the SSR_EDIT_ENROLLMENT_RESP message the structure of which is shown below.

```
<schema>
  targetNamespace=http://xmlns.oracle.com/Enterprise/HCM/services
  <include>
    schemaLocation=SCC_ENTITY_ENROLLMENT_RESP.xsd
  </include>
</schema>
```

**Figure 17: SSR_EDIT_ENROLLMENT_RESP Message Parameters**

The following is an example of the SSR_EDIT_ENROLLMENT_RESP message that the SSR_EDIT_ENROLLMENT service operation transmits to the Enrollment UI:

```
<?xml version="1.0"?>
<SSR_EDIT_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <MSGS>
      <MSG>
        <ID>158-2183</ID>
        <DESCR>This class has been updated.</DESCR>
        <MESSAGE_SEVERITY>M</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ENTITY_INST_1</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
    </MSGS>
  </detail>
</SSR_EDIT_ENROLLMENT_RESP>
```
</MSG>
</MSG>
</detail>
</SSR_EDIT_ENROLLMENT_RESP>
**Swap Enrollment**

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_ENROLLMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_SWAP_ENROLLMENT: allows student to swap classes</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation allows the student to swap a class in which student is enrolled with another class.</td>
</tr>
</tbody>
</table>
| Description: | This Service Operation allows the student to swap a class in which student is enrolled with another class. In this case either the ‘change to class nbr’ or ‘Drop this class if enrolled’ parameter should be passed. The enrollment options to be provided will be derived from SSR_STUDENT_ENROLLMENT_OPTIONS service operation. Enrollment request action from service tester should be ‘Enroll’ or “Swap” depending on the value of ‘ENRL_REQ_ACTION’ (E or S respectively) in ENROLLMENT_REQUEST_DEFAULT.  
  • Classes that may be swapped are classes that have not been graded, have an enrollment status of enrolled and the enrollment status reason is not WDRW or DROP, or the enrollment status of waitlisted  
  • Swap To class selection:  
    • Class Nbr validation – checks if the passed Class Nbr is valid for the Institution and Term.  
    • Check Career Rules – checks if the student is allowed to take this course, based on career rules. For example, an undergraduate student should be prevented from putting Law classes in their cart.  
    • OEE Edits- If the class selected is an Open Entry/Open Exit class, check if the student’s program allows enrollment to this type of class. Setup for this can be found at Setup SACR > Foundation Tables > Academic Structure > Academic Program Table. (See: PeopleSoft Campus Solutions 9.0 Application Fundamentals PeopleBook, Defining Programs, Plans, and Subplans)  
    • Check Last Date to Enroll – if the last date to enroll for this class has passed, don’t allow the class to be added to the cart.  
    • Check duplicate enrollment – if the student is already enrolled in the class, don’t allow the class to be added to the cart.  
  • Enrollment Request Detail creation:  
    • If the Self Service Enrollment Option for Swap is ‘Drop if Enrolled’ and the student is not enrolled in the new class nbr or either related classes, then the enrollment action will be E (Enroll) and the class nbr will be the new class and the original class number will be in the drop class if enrolled field.  
    If the Self Service Enrollment Option for Swap is “Swap” then the enrollment action will be S (Swap) and the class nbr goes in the “original class nbr” field and the new class number goes in the “class number to change to” field. |
| Users: | Student |
Processing: This service operation performs the following steps:

- Verify required input parameters exist.
- Performs the above mentioned validations on the input parameters.
- Creates an entry in Enrollment Request data modal and invokes the COBOL process.
- Responds with the Info/error messages.

Output: Response message contains Information/error messages occurred in enrollment.

Error Conditions: The service operation contains Information/error messages occurred in enrollment.

- Invalid input parameters.
- Any of the validations mentioned above fail.
- COBOL process results in error.

**Input Message: SSR_SWAP_ENROLLMENT_REQ**

The following diagram shows the input message structure:

```
<schema>
  targetNamespace: http://xmlns.oracle.com/Enterprise/HCMessages
  <include>
    schemaLocation: BCC_ENTITY_ENROLL_REQ_HEADER_V1.xsd
  </include>
</schema>
```

**Figure 18: SSR_SWAP_ENROLLMENT_REQ Message Structure**

Input parameters:

- *User ID
- *Password
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number
- Related Class Number 1
- Related Class Number 2
- Change to Class nbr
- Drop this class if enrolled
- Wait List Okay
- Grading Basis
- Units Taken
- Course Count
- Academic Program
• Start Date
• Class Permission Nbr
• Instructor ID
• Requirement Designation Option
• *Enrollment request Action

Parameters marked with * are mandatory.

• User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
• EmpID - the ID of the Student.
• Class Number - the Class Number that the student is trying to swap with another class.
• Related Class Number 1 & 2 – Classes that are related to the class number to be swapped with.
• Change to class nbr – New Class number to be swapped with.
• Drop this class if enrolled – class number to be swapped, if the ‘Drop if enroll’ option is enabled.
• Wait list okay – Y/N flag to be set by user if enabled.
• Grading Bases – Can be selected from LOV by user if enabled.
• Units taken – Number of units, can be selected by user if enabled.
• Academic Program - Can be selected from LOV by user if enabled.
• Start Date - if enabled, selected by user.
• Class Permission Nbr – Permission number to be entered by user if enabled.
• Instructor Id - Can be selected from LOV by user if enabled.
• Requirement Designation Option - Y/N flag to be set by user if enabled
• Enrollment request Action – to be set as “S” (Swap).

The following is an example of the SSR_SWAP_ENROLLMENT_REQ message that the SSR_SWAP_ENROLLMENT service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_SWAP_ENROLLMENT_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
<ENROLL_REQ_HEADER>
<ENRL_REQUEST_ID></ENRL_REQUEST_ID>
<OPRID></OPRID>
<ENRL_REQ_PROC_ST></ENRL_REQ_PROC_ST>
<ENRL_REQ_SOURCE></ENRL_REQ_SOURCE>
<PROCESS_INSTANCE>0</PROCESS_INSTANCE>
<DTTM_STAMP_SEC></DTTM_STAMP_SEC>
<ENROLL_REQUEST_DETAILS>
<ENROLL_REQUEST_DETAIL>
<EMPLID>SR13456</EMPLID>
<ACAD_CAREER>UGRD</ACAD_CAREER>
<INSTITUTION>PSUNV</INSTITUTION>
<STRM>0530</STRM>
<CLASS_NBR>1111</CLASS_NBR>
<ENRL_REQ_ACTION>S</ENRL_REQ_ACTION>
<ENRL_ACTION_REASON></ENRL_ACTION_REASON>
<ENRL_ACTION_DT></ENRL_ACTION_DT>
<UNT_TAKEN>3</UNT_TAKEN>
<UNT_EARNED>0</UNT_EARNED>
<CRSE_COUNT>1</CRSE_COUNT>
<REPEAT_CODE></REPEAT_CODE>
<CRSE_GRADE_INPUT></CRSE_GRADE_INPUT>
<GRADING_BASIS_ENRL>GRD</GRADING_BASIS_ENRL>
<class_PRMSN_NBR>0</class_PRMSN_NBR>
<class_NBR_CHG_TO>1237</class_NBR_CHG_TO>
<DROP_CLASS_IF_ENRL>0</DROP_CLASS_IF_ENRL>
<CHG_TO_WL_NUM>0</CHG_TO_WL_NUM>
<RELATE_CLASS_NBR_1></RELATE_CLASS_NBR_1>
<RELATE_CLASS_NBR_2></RELATE_CLASS_NBR_2>
<OVRD_CLASS_LIMIT></OVRD_CLASS_LIMIT>
```

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
When the Integration Broker receives the SSR_SWAP_ENROLLMENT_REQ message, it responds with the SSR_SWAP_ENROLLMENT_RESP message the structure of which is shown below.

```
<?xml version="1.0"?>
<SSR_SWAP_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <!-- Message parameters here -->
  </detail>
</SSR_SWAP_ENROLLMENT_RESP>
```

**Figure 19: SSR_SWAP_ENROLLMENT_RESP Message Parameters**

The following is an example of the SSR_SWAP_ENROLLMENT_RESP message that the SSR_SWAP_ENROLLMENT service operation transmits to the Enrollment UI:

```
<?xml version="1.0"?>
<SSR_SWAP_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <!-- Message parameters here -->
  </detail>
</SSR_SWAP_ENROLLMENT_RESP>
```
### Enrollment Validation

**Description**

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_ENROLLMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_ENR_VALIDATE: validate shopping cart</td>
</tr>
<tr>
<td>Summary:</td>
<td>This Service Operation is used to validate items before they can be checked out.</td>
</tr>
<tr>
<td>Description:</td>
<td>This Service Operation is used to validate items before they can be checked out of the shopping cart, based on validation mode passed in the request. It can validate multiple items and send back the validation results of all the items. The operation converts the items to be validated into enrollment request, validates the enrollment request based on the mode, then invokes the validate cobol engine. Performs pre cobol validations and invokes the cobol engine only when these validations are successful, whereas the validation service operation under shopping cart service just performs the cobol validations.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>Identifies the cart type (course) and builds course shopping cart entities. Validates user access to perform validation. Converts shopping cart items to enrollment request. Reads the Validation Mode and performs pre cobol validations based on the mode. Invokes cobol validation engine. Sends back the validation results.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response consists of validation results which can be an informational message, for example, ‘Ok to Add’ on successful validation or error messages.</td>
</tr>
</tbody>
</table>
Error Conditions: If the user has no access to perform the validation, an error is sent back conveying the same.

- Service operation has 2 validation modes
  - ‘Y’ Class Number
  - ‘N’ Entity Validation

- Validations performed
  - If validation mode is Class nbr (‘Y’), then the class number validation is performed and then the COBOL engine is triggered.
  - If Validation mode is Entity Validation (full Validation), then following validations are performed:
    - Class Nbr validation.
    - System edits on enrollment request entity hierarchy - Required, Yes/No, Translate, Prompt.
    - Enrollment requests should only be created when:
      - Student is term activated and eligible to enroll.
      - Student is not withdrawn or cancelled.
      - Student has an active program for the term.
    - Users can submit enrollment transactions only if their UserId is associated to the Permission List specified in Setup SACR > Security > Secure Student Administration > Permission List > Self-Service Enrollment Perm List.
    - If online appointment checking is enforced, check if student has enrollment appointment for the selected class’ session.
    - Self-service: Enrollment requests are always built with EMPLID = UserId’s EMPLID.
    - Check Career Rules – checks if the student is allowed to take this course, based on career rules. For example, an undergraduate student should be prevented from putting Law classes in their cart.
    - OEE Edits- If the class selected is an Open Entry/Open Exit class, check if the student’s program allows enrollment to this type of class.
    - Check Last Date to Enroll – if the last date to enroll for this class has passed, don’t allow the class to be added to the cart.
    - Check duplicate enrollment – if the student is already enrolled in the class, don’t allow the class to be added to the cart.

**Input Message: SSR_ENR_VALIDATE_REQ**

The following diagram shows the SSR_ENR_VALIDATE REQ message structure:
The structure of SCC_ENTITY_ENR_VALIDATE.V1.xsd is shown below:

```
<schema>
  targetNamespace=http://xmlns.oracle.com/Enterprise/HCM/services
  <include>
    schemaLocation=SCC_ENTITY_CLS_MEETING_PATTERN.V1.xsd
  </include>
  <complexType name="SCC_ENTITY_INST_ID">
    <simpleType name="string">
      <restriction base="string">
        <maxLength value="64"/>
      </restriction>
    </simpleType>
  </complexType>
  <complexType name="SCC_SHOP_CART_TYPE">
    <simpleType name="string">
      <restriction base="string">
        <maxLength value="64"/>
      </restriction>
    </simpleType>
  </complexType>
  <complexType name="COURSE_SHOP_CART">
    <complexContent>
      <extension base="SCC_SHOP_CART_TYPE">
        <attribute name="SCC_ENTITY_INST_ID" type="string"/>
        <sequence>
          <element name="ITEM" type="SCC_ENTITY_INST_ID"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="SHOPPING_CART">
    <complexContent>
      <extension base="COURSE_SHOP_CART">
        <sequence>
          <element name="SHOPPING_CART" maxOccurs="3"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CLASS_COMPONENT_SECTION">
    <complexContent>
      <extension base="CLASS_COMPONENT">
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CLASS_COMPONENT">
    <complexContent>
      <extension base="CLASS_TEXT_BOOK">
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CLASS_TEXT_BOOK">
    <complexContent>
      <extension base="TEXT_BOOK_DETAILS">
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
  <complexType name="TEXT_BOOK_DETAILS">
    <complexContent>
      <extension base="ENROLLMENT_INFORMATION">
        <attribute name="TEXT_BOOK_REGION" type="string"/>  
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ENROLLMENT_INFORMATION">
    <complexContent>
      <extension base="SSR_CLASS_SECTION">
        <attribute name="SSR_REGION" type="string"/>  
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
  <complexType name="SSR_CLASS_SECTION">
    <complexContent>
      <extension base="ITEM">
        <attribute name="ITEM_REGION" type="string"/>  
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ITEM">
    <complexContent>
      <extension base="SHOPPING_CART">
        <attribute name="ITEM_REGION" type="string"/>  
        <attribute name="CLASS_REGION" type="string"/>  
        <attribute name="CLASS_SECTION" type="string"/>  
      </extension>
    </complexContent>
  </complexType>
</schema>
```
In the above diagram, SSR_CLASS_SECTION has been defined as child of ITEM but this should not be part of the request xml. The request xml should only include elements as detailed in the following section.

Figure 201: SSR_ENR_VALIDATE_REQ Message Structure

The following are the input parameters for the UI to pass to the service operation:

- *User ID
- *Password
- *EMPLID
- *INSTITUTION
- *Academic Career
- *Term
- *Class Number
- *Validation Mode

- Parameters marked with * are mandatory.
- User ID & Password are not the part of actual request payload. These parameters are to be sent through WS Security header as username token.
- EmplID - the ID of the Student.
- Institution – the Institution of the student.
- Academic Career – the Career of the student.
- Term – the Term of the student.
- Class Number - the Class Number for which the enrollment options is requested.
- Validation Mode – can be any of the modes - Class Nbr/Entity Validation.

The following is an example of the SSR_ENR_VALIDATE_REQ message that the SSR_ENR_VALIDATE service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_ENR_VALIDATE_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <SHOPPING_CART>
    <COURSE_SHOP_CART>
      <SCC_SHOP_CART_TYPE>COURSE</SCC_SHOP_CART_TYPE>
      <ITEMS>
        <ITEM>
          <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
          <EMPLID>SR12030</EMPLID>
          <INSTITUTION>PSUNV</INSTITUTION>
          <ACAD_CAREER>UGRD</ACAD_CAREER>
          <STRM>0670</STRM>
          <CLASS_NBR>1718</CLASS_NBR>
          <WAIT_LIST_OKAY>N</WAIT_LIST_OKAY>
          <CLASS_PRMSN_NBR>0</CLASS_PRMSN_NBR>
          <GRADING_BASIS_ENRL></GRADING_BASIS_ENRL>
          <UNT_TAKEN>0</UNT_TAKEN>
          <INSTRUCTOR_ID></INSTRUCTOR_ID>
          <RQMNT_DESIGNTN_OPT></RQMNT_DESIGNTN_OPT>
          <START_DT></START_DT>
          <CRSE_COUNT>0</CRSE_COUNT>
          <ACAD_PRG></ACAD_PRG>
          <RELATE_CLASS_NBR_1>0</RELATE_CLASS_NBR_1>
          <RELATE_CLASS_NBR_2>0</RELATE_CLASS_NBR_2>
          <SSR_VALIDATIN_MODE>N</SSR_VALIDATIN_MODE>
        </ITEM>
      </ITEMS>
    </COURSE_SHOP_CART>
  </SHOPPING_CART>
</SSR_ENR_VALIDATE_REQ>
```
Output Message: SCC_VALIDATE_RESP

When the Integration Broker receives the SSR_ENR_VALIDATE_REQ message, it responds with the SCC_VALIDATE_RESP message.

The following diagram shows the SCC_VALIDATE_RESP message structure:

```
<?xml version="1.0"?>
<SSR_ENR_VALIDATE_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <detail>
    <MSGS>
      <MSG>
        <ID>14732-11</ID>
        <DESCR>OK to Add.</DESCR>
        <MESSAGE_SEVERITY>I</MESSAGE_SEVERITY>
        <PROPS>
          <PROP>
            <SCC_ENTITY_INST_ID>ITEM ROW :1</SCC_ENTITY_INST_ID>
            <PROPNAME></PROPNAME>
          </PROP>
        </PROPS>
      </MSG>
    </MSGS>
  </detail>
</SSR_ENR_VALIDATE_RESP>
```

Figure 12: SCC_VALIDATE_RESP Message Structure

The following is an example of the SCC_VALIDATE_RESP message that the SSR_ENR_VALIDATE service operation transmits to the Enrollment UI:

```
GET StudyList Details

Description
```
Service: SSR_ENROLLMENT
Operation: SSR_GET_ENROLLMENT: Retrieve details of StudyList
Summary: Allows the student to view the study list details.
Description: The Get study list Service can be used to retrieve the student study list detail. Study List detail can be retrieved for the student. The INSTITUTION, ACADEMIC CAREER and TERM are optional input parameters. If no TERM is passed, it returns enrollment details for all terms.
Users: Student
Processing: This service operation performs the following steps:
• Verify required input parameters exist.
• Perform validations on the input parameters.
• Validate if the input term obeys the Term table date settings.
• Use the Study List entity and its child entities to retrieve and populate Study list details.
• Prepare response message containing the study list details.
Output: Response message containing the study list details is sent back to the user.
Error Conditions: The following conditions result in a service error:
• The supplied input parameters are invalid.
• Today’s date is outside the input term’s ‘Display in Self-Service’ range. This is set in Setup SACR > Foundation Tables > Term Setup > Term Session Table. (See: PeopleSoft Campus Solutions 9.0 Application Fundamentals PeopleBook, Establishing Terms and Sessions)

Input Message: SSR_GET_ENROLLMENT_REQ

The following diagram shows the input parameters that the SSR_GET_ENROLLMENT service operation receives from the user interface:

![Figure 21: SSR_GET_ENROLLMENT_REQ Message Parameters](image)

The following are the input parameters for the user interface to pass to the service operation:

- EMPLID
- ACAD_CAREER
- INSTITUTION
- STRM
EMPLID would be overridden with the Logged in User ID’s EMPLID. User ID should be passed using username token. The remaining parameters are optional.

The following is an example of the SSR_GET_ENROLLMENT_REQ message that the SSR_GET_ENROLLMENT service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_GET_ENROLLMENT_REQ xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
    <SCC_ENTITY_INST_ID>ITEM ROW 1</SCC_ENTITY_INST_ID>
    <EMPLID>SR12030</EMPLID>
    <ACAD_CAREER>UGRD</ACAD_CAREER>
    <INSTITUTION>PSUNV</INSTITUTION>
    <STRM>0630</STRM>
</SSR_GET_ENROLLMENT_REQ>
```

Output Message: SSR_GET_ENROLLMENT_RESP

When the Integration Broker receives the SSR_GET_ENROLLMENT_REQ message, it responds with the SSR_GET_ENROLLMENT_RESP message the structure of which is shown below.

![Figure 22: SSR_GET_ENROLLMENT_RESP Message Structure](image)
The following is an example of the SSR_GET_ENROLLMENT_RESP message that the SSR_GET_ENROLLMENT service operation transmits to the Enrollment UI:

<?xml version="1.0"?>
<SSR_GET_ENROLLMENT_RESP xmlns="http://xmlns.oracle.com/Enterprise/HCM/services">
  <EMPLID>SR12030</EMPLID>
  <SS_MESSAGE_LONG/>
  <SSR_ENRL_TERMS>
    <SSR_ENRL_TERM>
      <INSTITUTION>PSUNV</INSTITUTION>
      <INSTITUTION_LOVDescr>PeopleSoft University</INSTITUTION_LOVDescr>
      <ACAD_CAREER>UGRD</ACAD_CAREER>
      <ACAD_CAREER_LOVDescr>Undergraduate</ACAD_CAREER_LOVDescr>
      <STRM>0630</STRM>
      <STRM_LOVDescr>2009 Fall</STRM_LOVDescr>
      <TERM_BEGIN_DT>2009-08-30</TERM_BEGIN_DT>
      <TERM_END_DT>2009-12-12</TERM_END_DT>
      <ENROLLMENT_DETAIL>
        <SUBJECT>WOMEN'S</SUBJECT>
        <CATALOG_NBR>495</CATALOG_NBR>
        <ASSOCIATED_CLASS>1</ASSOCIATED_CLASS>
        <STDNT_ENRL_STATUS>E</STDNT_ENRL_STATUS>
        <ENROLLMENT_CLASS_NAME>WOMEN'S 495 - Spec Topics in Women's Studies</ENROLLMENT_CLASS_NAME>
        <CLASS_DESCRIPTION>WOMEN'S 495 - Spec Topics in Women's Studies</CLASS_DESCRIPTION>
        <WAITLIST_POS>0</WAITLIST_POS>
        <UNT_TAKEN>3</UNT_TAKEN>
        <GRADING_SCHEME_ENR>UGD</GRADING_SCHEME_ENR>
        <GRADE_BASIS_DESCR>Graded</GRADE_BASIS_DESCR>
        <RQMNT_DESIGNTN/>
        <RQMT_DESG_EXPRFORMAL/>
        <CRSE_ID>666791</CRSE_ID>
        <CRSE_DESCR>WOMEN'S 495 - Spec Topics in Women's Studies</CRSE_DESCR>
        <CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
        <SESSION_CODE>1</SESSION_CODE>
        <ACAD_PROG>LAU</ACAD_PROG>
        <ACAD_PROG_DESCR>Liberal Arts Undergraduate</ACAD_PROG_DESCR>
        <ACAD_PROG_DESCR_VISIBLE>Y</ACAD_PROG_DESCR_VISIBLE>
        <ENRL_ADD_DT>2010-02-19</ENRL_ADD_DT>
        <ENRL_DROP_DT/>
        <GRADE_DT/>
        <REPEAT_CODE/>
        <REPEAT_CODE_EXPRFORMAL/>
        <ENRL_CLASS_SECTION>
          <CLASS_SECTION>2</CLASS_SECTION>
          <SSR_COMPONENT>LEC</SSR_COMPONENT>
          <LMS_LINK/>
          <LINK_GRADEBOOK_VISIBLE>N</LINK_GRADEBOOK_VISIBLE>
          <LMS_LINK_VISIBLE>N</LMS_LINK_VISIBLE>
          <START_DT>2009-08-30</START_DT>
          <END_DT>2009-12-12</END_DT>
          <SSR_ENRL_SWAP_ALLOW>True</SSR_ENRL_SWAP_ALLOW>
          <SSR_ENRL_EDIT_ALLOW>False</SSR_ENRL_EDIT_ALLOW>
        </ENRL_CLASS_SECTION>
      </ENROLLMENT_DETAIL>
    </SSR_ENRL_TERM>
  </SSR_ENRL_TERMS>
</SSR_GET_ENROLLMENT_RESP>
<GRADE_DT/>
<REPEAT_CODE/>
<REPEAT_CODE_LOVDescr/>
<ENRL_CLASS_SECTIONS>
  <ENRL_CLASS_SECTION>
    <CLASS_NBR>6638</CLASS_NBR>
    <CLASS_SECTION>2</CLASS_SECTION>
    <SSR_COMPONENT>LEC</SSR_COMPONENT>
    <SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
    <LMS_LINK/>
    <LINK_GRADEBOOK_VISIBLE>N</LINK_GRADEBOOK_VISIBLE>
    <LINK_VISIBLE>N</LINK_VISIBLE>
    <START_DT>2009-08-30</START_DT>
    <END_DT>2009-12-12</END_DT>
    <SSR_ENRL_SWAP_ALLOW>True</SSR_ENRL_SWAP_ALLOW>
    <SSR_ENRL_EDIT_ALLOW>False</SSR_ENRL_EDIT_ALLOW>
    <SSR_ENRL_DROP_ALLOW>True</SSR_ENRL_DROP_ALLOW>
    <CAMPUS>MAIN</CAMPUS>
    <CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
    <CLASS_MEETING_PATTERNS>
      <CLASS_MEETING_PATTERN>
        <SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
        <SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
        <SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
        <SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
        <SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
      </CLASS_MEETING_PATTERN>
    </CLASS_MEETING_PATTERNS>
  </ENRL_CLASS_SECTION>
  <ENRL_CLASS_SECTION>
    <CLASS_NBR>6637</CLASS_NBR>
    <CLASS_SECTION>1</CLASS_SECTION>
    <SSR_COMPONENT>LEC</SSR_COMPONENT>
    <SSR_COMPONENT_LOVDescr>Lecture</SSR_COMPONENT_LOVDescr>
    <LMS_LINK/>
    <LINK_GRADEBOOK_VISIBLE>N</LINK_GRADEBOOK_VISIBLE>
    <LINK_VISIBLE>N</LINK_VISIBLE>
    <START_DT>2009-08-30</START_DT>
    <END_DT>2009-12-12</END_DT>
    <SSR_ENRL_SWAP_ALLOW>True</SSR_ENRL_SWAP_ALLOW>
    <SSR_ENRL_EDIT_ALLOW>False</SSR_ENRL_EDIT_ALLOW>
    <SSR_ENRL_DROP_ALLOW>True</SSR_ENRL_DROP_ALLOW>
    <CAMPUS>MAIN</CAMPUS>
    <CAMPUS_LOVDescr>Main Hacienda Campus</CAMPUS_LOVDescr>
    <CLASS_MEETING_PATTERNS>
      <CLASS_MEETING_PATTERN>
        <SSR_MTG_SCHED_LONG>TBA</SSR_MTG_SCHED_LONG>
        <SSR_MTG_LOC_LONG>TBA</SSR_MTG_LOC_LONG>
        <SSR_INSTR_LONG>Staff</SSR_INSTR_LONG>
        <SSR_MTG_DT_LONG>TBA</SSR_MTG_DT_LONG>
        <SSR_TOPIC_LONG>TBA</SSR_TOPIC_LONG>
      </CLASS_MEETING_PATTERN>
    </CLASS_MEETING_PATTERNS>
  </ENRL_CLASS_SECTION>
</ENRL_CLASS_SECTIONS>
</ENROLLMENT_DETAILS>
</SSR_ENRL_TERM>
</SSR_ENRL_TERMS>
</SSR_ENRL_STUDYLIST>
</SSR_GET_ENROLLMENT_RESP>
GET Enrollment Deadlines

Description

<table>
<thead>
<tr>
<th>Service:</th>
<th>SSR_ENROLLMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation:</td>
<td>SSR_STUDYLIST_DEADLINES: Retrieve details of class deadlines</td>
</tr>
<tr>
<td>Summary:</td>
<td>Allows the student to view class deadlines information</td>
</tr>
<tr>
<td>Description:</td>
<td>After invoking the Get Enrollment service operation, the UI can invoke the Study list Deadlines service operation to retrieve the class deadline details. Input to this service operation comes from the response of the Get Enrollment operation. The operation can process multiple deadlines request and respond with multiple deadlines details.</td>
</tr>
<tr>
<td>Users:</td>
<td>Student</td>
</tr>
<tr>
<td>Processing:</td>
<td>This service operation performs the following steps:</td>
</tr>
<tr>
<td></td>
<td>• Validates input parameters.</td>
</tr>
<tr>
<td></td>
<td>• Use the Deadlines entity to retrieve and populate Class Deadline details.</td>
</tr>
<tr>
<td></td>
<td>• Prepare response message containing the Class Deadline details.</td>
</tr>
<tr>
<td>Output:</td>
<td>Response message containing the Class Deadline details is sent back to the user.</td>
</tr>
<tr>
<td>Error Conditions:</td>
<td>The following conditions result in a service error:</td>
</tr>
<tr>
<td></td>
<td>• Validation error due to invalid request parameters.</td>
</tr>
</tbody>
</table>

Input Message: SSR_STUDYLIST_DEADLINES_REQ

The following diagram shows the input message structure:

```xml
<schema>

targetNamespace http://xmlns.oracle.com/Enterprise/HCM/services

SSR_STUDYLIST_DEADLINES_REQ ➔ STUDYLIST_DEADLINE ➔

Figure 23: SSR_STUDYLIST_DEADLINES_REQ Message Structure

The following input parameters are mandatory for the user interface to pass to the service operation:

• Emplid
• Institution
• Academic Career
• Term
• Academic Program
• Course ID
• Course Offering Nbr
• Session
• Class Section

The following is an example of the SSR_STUDYLIST_DEADLINES_REQ message that the SSR_STUDYLIST_DEADLINES

service operation receives from an Enrollment application:

```xml
<?xml version="1.0"?>
<SSR_STUDYLIST_DEADLINES_REQ xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <STUDYLIST_DEADLINE>
    <SCC_ENTITY_INST_ID>ITEM ROW :0</SCC_ENTITY_INST_ID>
    <EMPLID>SR12030</EMPLID>
    <ACAD_CAREER>UGRD</ACAD_CAREER>
    <INSTITUTION>PSUNV</INSTITUTION>
    <STRM>0665</STRM>
    <ACAD_PROG>LAU</ACAD_PROG>
    <CRSE_ID>003146</CRSE_ID>
    <CRSE_OFFER_NBR>1</CRSE_OFFER_NBR>
    <SESSION_CODE>6W2</SESSION_CODE>
    <CLASS_SECTION>SF</CLASS_SECTION>
  </STUDYLIST_DEADLINE>
</SSR_STUDYLIST_DEADLINES_REQ>
```

Output Message: SSR_STUDYLIST_DEADLINES_RESP

When the Integration Broker receives the SSR_STUDYLIST_DEADLINES_REQ message, it responds with the SSR_STUDYLIST_DEADLINES_RESP message the structure of which is shown below.

```xml
<?xml version="1.0"?>
<SSR_STUDYLIST_DEADLINES_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <STUDYLIST_DEADLINE>
    <START_DT></START_DT>
    <END_DT></END_DT>
    <LST_DROP_DT_DEL></LST_DROP_DT_DEL>
    <LST_DROP_DT_RET></LST_DROP_DT_RET>
    <LST_DROP_DT_PEN></LST_DROP_DT_PEN>
    <LST_DROP_DT_PEN_2></LST_DROP_DT_PEN_2>
    <LST_WD_WO_PEN_DT></LST_WD_WO_PEN_DT>
  </STUDYLIST_DEADLINE>
</SSR_STUDYLIST_DEADLINES_RESP>
```

Figure 24: SSR_STUDYLIST_DEADLINES_RESP Message Parameters

The following is an example of the SSR_STUDYLIST_DEADLINES_RESP message that the SSR_STUDYLIST_DEADLINES service operation transmits to the Enrollment UI:

```xml
<?xml version="1.0"?>
<SSR_STUDYLIST_DEADLINES_RESP xmlns = "http://xmlns.oracle.com/Enterprise/HCM/services">
  <STUDYLIST_DEADLINE>
    <START_DT></START_DT>
    <END_DT></END_DT>
    <LST_DROP_DT_DEL></LST_DROP_DT_DEL>
    <LST_DROP_DT_RET></LST_DROP_DT_RET>
    <LST_DROP_DT_PEN></LST_DROP_DT_PEN>
    <LST_DROP_DT_PEN_2></LST_DROP_DT_PEN_2>
    <LST_WD_WO_PEN_DT></LST_WD_WO_PEN_DT>
  </STUDYLIST_DEADLINE>
</SSR_STUDYLIST_DEADLINES_RESP>
```
<LST_WD_W_PEN_DT/>
<LST_WD_W_PEN2_DT/>
<LST_CANCEL_DT/>
<SSR_DYN_DLRP_DT/>
<LST DROP DT DEL_DESCR/>
<LST DROP DT RET_DESCR/>
<LST DROP DT PEN_DESCR/>
<LST DROP DT PEN_2_DESCR/>
<SSR DYN DLRP DT DESCRIPT/>
<LST_CANCEL_DT_DESCR/>
<LST_WD_WO_PEN_DT_DESCR/>
<LST_WD_W_PEN_DT_DESCR/>
<LST_WD_W_PEN2_DT_DESCR>
<Class_DESCR/>
<Class_NBR>0</Class_NBR>
<Subject/>
<Catalog_NBR/>
</StudyList_DEADLINE>
</SSR_STUDYLIST_DEADLINES_RESP>
Configuration Hints and Tips

This section provides post installation setup recommendations for EWS and additional web service operations. For information about the configuration options on the Campus Solutions pages, refer to the Student Records and Enrollment PeopleBook updates for the Additional Features July 2011 (Enrollment Web Services Documentation Package.pdf posted to My Oracle Support).

For the Integration Broker configuration options, refer to the EWS Developer's Guide.

Entity Registry

Navigation: Set Up SACR > System Administration > Entity > Entity Registry

The entities for all core data records within the Enrollment and Course areas are delivered within the Entity Registry component. The entity data is considered system data and is necessary for the correct functioning of EWS. For a list of EWS delivered entities, please refer to the Appendix in this User Guide.

Recommendations:

• Register new institution-specific extension entities as children to any entity in the registry. For example, if you want to add an Institution specific Course Information entity to inform the student of additional Course detail, you need to add a new entity Course detail and associate this new entity with an existing parent entity, such as the CourseOfferingDetailResult entity.
• After associating the new entity with a parent entity, you must include the new entity into your PeopleSoft system as a PeopleCode application class. You can either write the code yourself or use the Generate Code button to automatically generate a PeopleCode application class template. You can then include your validations in the class template and include the class in your PeopleSoft system using Application Designer.
• When generating code for a new extension entity, select the appropriate code generation options:
  • Base App Class: Select if you want to write the code for a staged entity. In general, all extension entities will be of type Basic Entity. Basic entity represents an entity that captures data into a single core database table.
  • Note that the entities that you create automatically share their parent entities keys with other entities. This is a referential relationship of 1:Many for parent: child.
  • After you create a new entity, select the Generate XSD button to generate an XML schema. The Enrollment user interface must use the XSD to communicate with EWS. Using the XSD for the entity is a mandatory step because your Enrollment UI must validate the XML data for an institution-defined entity using the XSD before passing on the XML data to EWS.

See Also: PeopleSoft Campus Community Fundamentals 9.0 PeopleBook, Setting Up Entity Registry. The Setting Up Entity Registry chapter was posted to My Oracle Support as part of the Campus Community documentation updates for Bundle 22. Please review Bundle 22 documentation updates and any subsequent bundle documentation updates on My Oracle Support to check if there have been updates to this chapter.

Enrollment Configuration

Recommendations:

The EWS service operations depend upon a number of existing Campus Solutions system configuration options. The results that these services deliver are controlled by the following Installation and Student Records configuration options:
• Browse course service obeys the ‘Catalog Print’ setup option available in Requirement Designation Table and Course Attributes setups under Set Up SACR > Product Related > Student Records > Curriculum Management. Hence make sure that the ‘Catalog Print’ option is set if required.

See: PeopleSoft Student Records 9.0 PeopleBook, Preparing for the Course Catalog and Schedule of Classes

• Class Search

1. The Get Classes service operation validates the input Term against the data range specified in Term Values table and sends an error if the input term is not within the date range for the term specified under Self-Service Student/Visitor head in the below mentioned navigation. So if Term error is displayed, check the date range in the below mentioned navigation.

   Home > Setup SACR > Foundation Tables > Term Setup > Term Values Table
<tab: Display in class search>

2. If search result exceeds error threshold, response will not include any class data and a message indicating this will be sent in the response in the tag <ERROR_WARN_TEXT>

3. If search result exceeds warning threshold and the obey_warning_limit in the request payload is set to YES, response will include warning message in the tag <ERROR_WARN_TEXT> along with any class data and is up to the UI either to show or not show the search results.

See: PeopleSoft Campus Solutions 9.0 Application Fundamentals PeopleBook, Establishing Terms and Sessions

• Get StudyList

1. The Get Studylist service operation validates the input Term against the date range specified in Term/Session Table and sends an error back if the input term is not within the date range for the term specified against the “Enrollment & Shopping Cart” head in the “Display in Self Service” group box in the navigation shown below. So if this error is seen, check the date range in the below mentioned navigation.

   Home > Setup SACR > Foundation Tables > Term Setup > Term/Session Table > Term Table
2. If the service operation returns the message "You are not registered for classes in this term," it indicates that the student is not enrolled for any class and hence the message. Once the student is enrolled, the service operation will retrieve the appropriate enrollment details.

   See: PeopleSoft Campus Solutions 9.0 Application Fundamentals PeopleBook, Establishing Terms and Sessions

   - **Enrollment Shopping Cart**
     - The input Term is validated against the Student Active Term. In case an invalid term error is seen in the response while adding item to the cart, ensure that the input term is activated for the student under navigation Home>Records and Enrollment>Student Term Information>Term Activate a student. Also ensure that the student is not withdrawn or cancelled. (See: PeopleSoft Student Records 9.0 PeopleBook, Maintaining Student Career Term Records)
     - Ensure that the student’s user id is associated to the Self Service Permission List. Students can submit enrollment transactions only when their UserId is associated to the Self Serv Permission List under Setup SACR > Security > Secure Student Administration > Permission List > Self-Serv Enrollment Perm List. Else error is sent in response while checking out the item or validating the item. (See: PeopleSoft Campus Solutions 9.0 Application Fundamentals PeopleBook, Securing Student Records, Setting Up Enrollment Security for Self-Service Enrollment)
     - In case of error indicating that user has no access to validate a cart item as part of validate operation, ensure that access is provided to the user for validating under navigation: Setup SACR > Common Definitions > Self Service > Student Records.<Allow Validation check box>  (See: PeopleSoft Campus Self Service 9.0 PeopleBook, Setting Up Student Records Self-Service)
• **Student Enrollment Options**
  • The input Term is validated against the Student Active Term. In case an invalid term error is seen in the response, ensure that the input term is activated for the student Home>Records and Enrollment>Student Term Information>Term Activate a student. Also ensure that the student is not withdrawn or cancelled.
  • Ensure that the student’s user id is associated to the Self Service Permission List. Students can submit enrollment transactions only when their UserId is associated to the Self Serv Permission List Setup SACR > Security > Secure Student Administration > Permission List > Self-Serv Enrollment Perm List.

• **Enrollment Requests (Add/Drop/Swap/Edit)**
  • The input Term is validated against the Student Active Term. In case an invalid term error is seen in the response, ensure that the input term is activated for the student under navigation Home>Records and Enrollment>Student Term Information>Term Activate a student. Also ensure that the student is not withdrawn or cancelled.
  • Ensure that the student’s user id is associated to the Self Service Permission List. Students can submit enrollment transactions only when their UserId is associated to the Self Serv Permission List Setup SACR > Security > Secure Student Administration > Permission List > Self-Serv Enrollment Perm List.
  • In case of swap operation, the ‘Swap Enrollment Action’ must be set as required under navigation: Setup SACR > Common Definitions > Self Service > Student Records.
List of Values

Most self-service user interfaces (UIs) constructed outside of your PeopleSoft Campus Solutions system will contain data fields that have predefined or “prompt” values used to control and streamline data entry by a self-service user. To use list of values that are defined inside your PeopleSoft Campus Community system, use the List of Values Web Service (SCC_GET_LOV, introduced in Feature Pack 3, July 2010). The service recognizes and provides the values that are used to populate drop-down list boxes on a user interface and for validating selections. The required parameters are the record and the field names for which you want a list of values to be returned. The service operation looks, by default, at the prompt table defined for the desired record and field name to retrieve the values for a field. Optionally, if you want to alter the default way in which the operation picks up the list of values for a field, use the List of Values Setup component (under Set Up SACR > System Administration > Utilities > List Of Values).

See Also: PeopleSoft Campus Community Fundamentals 9.0 PeopleBook, Setting Up List of Values. The Setting Up List of Values chapter was posted to My Oracle Support as part of the Campus Community documentation updates for Bundle 22. Please review Bundle 22 documentation updates and any subsequent bundle documentation updates on My Oracle Support to check if there have been updates to this chapter.

The ID for the My Oracle Support Campus Solutions 9.0 Documentation Home Page is 751540.1.

Examples:

**Browse Course Service:**

In the Browse Course UI, it is possible to use LOV web service to populate the list of institutions available, so that the user can select one to pass it to the Get Courses service operation (SSR_GET_COURSES).

For example: User can use the following inputs in the request message to invoke the LOV web service to populate list of institutions.

```
RECORDNAME: SSR_CRS_SRCH_RQ
FIELDNAME: INSTITUTION
```

**Class Search Service:**

In the Class Search UI, it is possible to use LOV web service to populate the list of institutions available, so that the user can select one to pass it to the Get Classes service operation (SSR_GET_CLASSES).

For example: User can use the following inputs in the request message to invoke the LOV web service to populate list of institutions.

```
RECORDNAME: SSR_CLSSRCH_REQ
```
Similarly, the UI can make use of the LOV web service for the following:

**Term**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : STRM

**Subject**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SUBJECT

**Campus**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : CAMPUS

**Course ID**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : CRSE_ID

**Operator for Catalog Nbr**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_EXACT_MATCH1

**Course Career**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : ACAD_CAREER

**Session**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SESSION_CODE

**Course component**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_COMPONENT

**Mode of Instruction**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : INSTRUCTION_MODE

**Location**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : LOCATION

**Meeting Start Time**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_START_TIME_OPR

**Meeting End Time**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_END_TIME_OPR

**Day of Week**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : INCLUDE_CLASS_DAYS

**Instructor Last Name**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_EXACT_MATCH2

**Maximum Units**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_UNITS_MAX_OPR

**Minimum Units**
- RECORDNAME: SSR_CLSSRCH_REQ
- FIELDNAME : SSR_UNITS_MIN_OPR

**Get StudyList Service:**

In the UI, it is possible to use LOV web service to populate the list of institutions available, so that the user can select one to pass it to the Get Studylist service operation (SSR_GET_ENROLLMENT).

For example: User can use the following inputs in the request message to invoke the LOV web service to populate list of institutions.
Similarly, the UI can make use of the LOV web service for the following:

EmplID

RECORDNAME : SSR_STDYLST_WRK
FIELDNAME : EMPLID

Academic Career

RECORDNAME : SSR_LOV_WRK
FIELDNAME : ACAD_CAREER
KEYS: EMPLID & INSTITUTION

Term

RECORDNAME : SSR_LOV_WRK
FIELDNAME : STRM
KEYS: EMPLID, ACAD_CAREER, INSTITUTION

**Student Enrollment Options/Enrollment Request Service:**
The UI can make use of the LOV service to obtain the valid values for a particular input item. The following can be used as inputs to the LOV service:

EmplID

RECORDNAME : SSR_ENRL_REQ
FIELDNAME : EMPLID

Institution

RECORDNAME : SSR_LOV_WRK
FIELDNAME : INSTITUTION
KEYS: EMPLID

Academic Career

RECORDNAME : SSR_LOV_WRK
FIELDNAME : ACAD_CAREER
KEYS: EMPLID & INSTITUTION

Term

RECORDNAME : SSR_LOV_WRK
FIELDNAME : STRM
KEYS: EMPLID, ACAD_CAREER, INSTITUTION

**Enrollment Request Action**

RECORDNAME : ENRL_REQ_DETAIL
FIELDNAME : ENRL_REQ_ACTION

The UI has to build the enrollment options page based on the output of Student Enrollment Options service. LOV service has to be used to obtain valid values for some of these enrollment options. The following can be used as inputs to the LOV service:

Related class Number 1

RECORDNAME : ENRL_REQ_DETAIL
FIELDNAME : RELATE_CLASS_NBR_1

Related class Number 2

RECORDNAME : ENRL_REQ_DETAIL
FIELDNAME : RELATE_CLASS_NBR_2

**Grading Basis**
The UI must invoke the Student Enrollment Options Service Operation to determine the appropriate Grading basis LOV setup to use.
The UI can then use the LOV service as shown below accordingly:

```plaintext
RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_GRD_BASE_LOV1

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_GRD_BASE_LOV2

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_GRD_BASE_LOV3
```

**Academic Program**

```
RECORDNAME : ENRL_REQ_DETAIL
FIELDNAME : ACAD_PROG
```

**Instructor ID**

The UI must invoke the Student Enrollment Options Service Operation to determine the appropriate Grading basis LOV setup to use. The UI can then use the LOV service as shown below accordingly:

```
RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV1

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV2

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV3

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV4

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV5

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV6

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV7

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV8

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV9

RECORDNAME : SSR_LOV_WRK
FIELDNAME : SSR_INSTR_LOV10
```

**Enrollment Shopping Cart Service:**

The UI can make use of the LOV service to obtain the valid values for a particular input item. The following can be used as inputs to the LOV service:

**EmplID**

```
RECORDNAME : SSR_CRS_SC_WRK
FIELDNAME : EMPLID
```

**Institution**

```
RECORDNAME : SSR_LOV_WRK
FIELDNAME : INSTITUTION
```
KEYS: EMPLID

Academic Career
RECORDNAME : SSR_LOV_WRK
FIELDNAME  : ACAD_CAREER
KEYS: EMPLID & INSTITUTION

Term
RECORDNAME : SSR_LOV_WRK
FIELDNAME  : STRM
KEYS: EMPLID, ACAD_CAREER, INSTITUTION

Recommendations:

- List of Values configuration is not mandatory for all prompts. For example, in the external user interface, you display the Institution field as a prompt and want the prompt to show all the Institution values defined inside your Campus Solutions system inside a specific record (the record name you passed as the input parameter). By default the LOV web service will return the description values for all the institutions defined inside that record. If this is what you want, you do not need to configure the List of Value Setup component for the record where the Institution field is located.
- You can override the default list of values that is displayed by using the List of Values Setup component to point to a different database table column or use a different description field. For example, suppose you have stored values such as Online and Distance Education. By default, the LOV web service retrieves the description values by scanning the record columns and returning the values for the first field encountered: In order that is DESCR, DESCRSHORT, DESCR30, DESCR100 or the first non-key field marked as a list box item inside the record. If the record you are using includes the DESCR, DESCRSHORT and DESCRFORMAL fields, by default, the LOV web service will return the DESCR values. Rather than providing the student the values of DESCR, you want the system to look at the table column DESCRFORMAL, - in this case, you must use List of Values configuration to override the default.
- List of Values configuration supports both Self Service and Administrative usage. The EWS service operations delivered for Additional Feature October 2011 support Self Service access only. Administrative access via EWS may be delivered in a future Additional Feature pack.

Enrollment Messages

Self-Service Enrollment Shopping Cart Validation and Enrollment Messages

Oracle delivers a set of error and informational message sets exclusively for the Enrollment Shopping Cart Validation feature and self-service enrollment processing. These message sets enable the enrollment process to return more self-service-friendly messages. You need to modify these messages to meet the specific needs of your students. You can view and make changes to these messages using the PeopleTools Message Catalog component (MESSAGE_CATALOG1) (PeopleTools, Utilities, Administration, Message Catalog).

These message sets are:

- Validation (Message Set 14632): These messages are returned when a student selects Validate on the Enrollment Shopping Cart page. The system returns a validation-specific message rather than a regular enrollment engine message. For example, if the validation process finds a time conflict, the message 14632, 17 is returned rather than the enrollment engine message 14640, 17.
- Enrollment (Message Set 14641): This message set is used for all messages received through self-service enrollment (excluding Enrollment Shopping Cart Validation). While actual enrollment is processed in the same way for self-service enrollment as when it is processed using an administrative user page, such as Quick Enroll, the fact that a request was
submitted through self-service enrollment now causes the system to return a self-service-specific error or informational message. For example, if an enrollment request submitted through the Quick Enroll component (QUICK_ENROLL) returns message 14640, 17 (time conflict), the same request submitted through self-service returns the self-service-specific message 14641,17. This enables more appropriate self-service messaging, without changing the messages used for non-self-service, administrative users.
Appendix

Entity Details for EWS

This section provides details of all entities leveraged by EWS.

Browse Course

Service Operation: SSR_GET_COURSES

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Entity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CourseSearchRequest</td>
<td>SSR_CRS_SRCH_RQ</td>
<td>SSR_GET_COURSES_REQ</td>
</tr>
<tr>
<td>Response Entities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CourseSearchResult</td>
<td>SSR_CRS_SRCH_RE</td>
<td>SSR_GET_COURSES_RESP</td>
</tr>
<tr>
<td>CourseSubject</td>
<td>SUBJECT_VW</td>
<td></td>
</tr>
<tr>
<td>CourseSummary</td>
<td>SSR_CRSE_ID_VW</td>
<td></td>
</tr>
<tr>
<td>CourseOfferingSummary</td>
<td>SSR_CRSE_OFF_VW</td>
<td></td>
</tr>
</tbody>
</table>

Entity Hierarchy Display

- CourseSearchResult
  - CourseSubject (0, Unbounded)
    - CourseSummary (0, Unbounded)
      - CourseOfferingSummary (0, Unbounded)

Service Operation: SSR_GET_COURSE_OFFERING

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Entity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CourseOfferingDetailRequest</td>
<td>SSR_CRS_OFF_RQ</td>
<td>SSR_GET_COURSE_OFFERING_REQ</td>
</tr>
<tr>
<td>Response Entities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CourseOfferingDetailResult</td>
<td>SSR_CRS_OFF_RE</td>
<td>SSR_GET_COURSE_OFFERING_RESP</td>
</tr>
<tr>
<td>CourseOfferingDetail</td>
<td>SSR_CRSE_OFF_VW</td>
<td></td>
</tr>
<tr>
<td>CourseOfferingDerived</td>
<td>SSR_CRSE_OFF_WRK</td>
<td></td>
</tr>
<tr>
<td>CourseComponentDetail</td>
<td>CRSE_COMP_VW3</td>
<td></td>
</tr>
<tr>
<td>CourseAttributesDetail</td>
<td>CRSE_ATTRIB_VW</td>
<td></td>
</tr>
<tr>
<td>CourseOfferingTerm</td>
<td>SSS_CRSETERM_VW</td>
<td></td>
</tr>
</tbody>
</table>
Entity Hierarchy Display

- **CourseOfferingDetailResult**
  - **CourseOfferingDetail** (Unbounded)
    - **CourseOfferingDerived** (Embedded)
    - **CourseComponentDetail** (Unbounded)
    - **CourseAttributesDetail** (Unbounded)
    - **CourseOfferingTerm** (Unbounded)

---

Class Search

**Service Operation: SSR_GET_CLASSES**

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassSearchRequest</td>
<td>SSR_CLSSRCH_RE</td>
<td>SSR_GET_CLASSES_REQ</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response Entities</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassSearchResult</td>
<td>SSR_CLSSRCH_RES</td>
<td>SSR_GET_CLASSES_RESP</td>
<td></td>
</tr>
<tr>
<td>ClassSearchSubject</td>
<td>SSR_CLSSRCH_SUB</td>
<td>SSR_CLSSRCH_SUB</td>
<td></td>
</tr>
<tr>
<td>ClassSearchSummary</td>
<td>SSR_CLSSRCH_SUM</td>
<td>SSR_CLSSRCH_SUM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
</tbody>
</table>

Entity Hierarchy Display

- **ClassSearchResult**
  - **ClassSearchSubject** (Unbounded)
    - **ClassSearchSummary** (Unbounded)
      - **ClassSectionMeetingPattern** (Unbounded)
Service Operation: SSR_GET_CLASS_SECTION

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Entity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionRequest</td>
<td>SSR_CLSDTL_RE</td>
<td>SSR_GET_CLASS_SECTION_REQ</td>
</tr>
<tr>
<td>Response Entities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionResult</td>
<td>SSR_CRS_OFF_RE</td>
<td>SSR_GET_CLASS_SECTION_RESP</td>
</tr>
<tr>
<td>ClassSection</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td>ClassSectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td>ClassComponent</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
</tbody>
</table>

Entity Hierarchy Display

- **ClassSectionResult**
  - **ClassSection** (0, Unbounded)
    - **ClassSectionCombined** (0, Unbounded)
    - **ClassComponent** (0, Unbounded)
      - **ClassSectionComponent** (0, Unbounded)
      - **ClassSectionMeetingPattern** (0, Unbounded)
    - **ClassSectionMeetingPattern** (0, Unbounded)
    - **ClassSectionTextbook** (0, Unbounded)
    - **ClassSectionTextbookDetails** (0, Unbounded)
    - **ClassSectionEnrlInfo** (0, Unbounded)

Study List

Service Operation: SSR_GET_ENROLLMENT

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Entity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StudyList</td>
<td>SSR_STDYLST_WRK</td>
<td>SSR_GET_ENROLLMENT_REQ</td>
</tr>
<tr>
<td>Response Entities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StudyList</td>
<td>SSR_STDYLST_WRK</td>
<td>SSR_GET_ENROLLMENT_RESP</td>
</tr>
<tr>
<td>EnrollmentTerm</td>
<td>SSR_ENRL_TERMMVW</td>
<td></td>
</tr>
<tr>
<td>EnrollmentDetail</td>
<td>SSR_ENRL_DTL_VW</td>
<td></td>
</tr>
<tr>
<td>EnrollmentClassSection</td>
<td>STDNT_ENRL_SSVW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
</tbody>
</table>

Entity Hierarchy Display

- **StudyList**
  - **EnrollmentTerm** (0, Unbounded)
    - **EnrollmentDetail** (0, Unbounded)
      - **EnrollmentClassSection** (0, Unbounded)
      - **ClassSectionMeetingPattern** (0, Unbounded)
**Service Operation: SSR_STUDYLIST_DEADLINES**

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StudyListDeadlines</td>
<td>SSR_STDYLST_WRK</td>
<td>SSR_STUDYLIST_DEADLINES_REQ</td>
</tr>
</tbody>
</table>

**Response Entities**  
StudyListDeadlines  
SSR_STDYLST_WRK  
SSR_STUDYLIST_DEADLINES_RESP

**Entity Hierarchy Display**

- **StudyListDeadlines**

---

**Student Enrollment Options**

**Service Operation: SSR_STUDENT_ENROLLMENT_OPTIONS**

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StudentEnrollmentOptionRequest</td>
<td>SSR_ENRL_REQ</td>
<td>SSR_STDNT_ENRL_OPTIONS_REQ</td>
</tr>
</tbody>
</table>

**Response Entities**  
EnrollmentRequestDefault  
ClassSectionOptions  
SSR_CLS_DTL_OPT  
SSR_STDNT_ENRL_OPTIONS_RESP

**Entity Hierarchy Display**

- **EnrollmentRequestDefault**
  - **ClassSectionOptions** (0, Unbounded)

---

**Enrollment Shopping Cart**

**SCC_SC_ADDITEM**

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_ADDITEM_REQ</td>
</tr>
<tr>
<td></td>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTBG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
</tbody>
</table>

---

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Response Entities

- CourseShoppingCart
- CourseShoppingCartItem
- Class Section
- Class SectionCombined
- Class Component
- ClassSectionComponent
- ClassSectionMeetingPattern
- ClassSectionTextbook
- ClassSectionTextbookDetails
- ClassSectionEnrlInfo

The Request and Response entities of SCC_SC_ADDITEM service operation will have the same structure since both represent the items on course shopping cart. In the above entity hierarchy, ClassSection entity has been defined as the child of CourseShoppingCartItem so this will reflect both in request as well as in the response entity structure. The request xml need not include ClassSection at all since it represents the class section details of the CourseShoppingCartItem and hence it will only be part of the response xml.

**Service Operation: SCC_SC_GETITEM**

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_GETITEM_REQ</td>
</tr>
</tbody>
</table>

Response Entities

- CourseShoppingCart
- CourseShoppingCartItem
- Class Section
- Class SectionCombined
- Class Component
- ClassSectionComponent
- ClassSectionMeetingPattern
- ClassSectionTextbook
- ClassSectionTextbookDetails
- ClassSectionEnrlInfo

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Entity Hierarchy Display

- **CourseShoppingCart**
  - **CourseShoppingCartItem** (0 Unbounded)
  - **ClassSection** (0 Unbounded)
    - **ClassSectionCombined** (0 Unbounded)
    - **ClassComponent** (0 Unbounded)
      - **ClassSectionComponent** (0 Unbounded)
        - **ClassSectionMeetingPattern** (0 Unbounded)
    - **ClassSectionMeetingPattern** (0 Unbounded)
    - **ClassSectionTextbook** (0 Unbounded)
      - **ClassSectionTextbookDetails** (0 Unbounded)
    - **ClassSectionEnrlInfo** (0 Unbounded)

**Service Operation: SCC_SC_GETCART**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Entity</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Request Entity</td>
<td>NA</td>
<td>SCC_SC_GETCART_REQ</td>
</tr>
</tbody>
</table>

**Response Entities**

- CourseShoppingCart
- CourseShoppingCartItem
- ClassSection
- ClassSectionCombined
- ClassComponent
- ClassSectionComponent
- ClassSectionMeetingPattern
- ClassSectionTextbook
- ClassSectionTextbookDetails
- ClassSectionEnrlInfo

**Entity Hierarchy Display**

- **CourseShoppingCart**
  - **CourseShoppingCartItem** (0 Unbounded)
  - **ClassSection** (0 Unbounded)
    - **ClassSectionCombined** (0 Unbounded)
    - **ClassComponent** (0 Unbounded)
      - **ClassSectionComponent** (0 Unbounded)
        - **ClassSectionMeetingPattern** (0 Unbounded)
    - **ClassSectionMeetingPattern** (0 Unbounded)
    - **ClassSectionTextbook** (0 Unbounded)
      - **ClassSectionTextbookDetails** (0 Unbounded)
    - **ClassSectionEnrlInfo** (0 Unbounded)

**Service Operation: SCC_SC_CLEARCART**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Entity</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Request Entity</td>
<td>NA</td>
<td>SCC_SC_CLEARCART_REQ</td>
</tr>
</tbody>
</table>

**Response Entities**

- NA
- NA
- SCC_SC_CLEARCART_RESP
Service Operation: SCC_SC_REMOVEITEM

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_REMOVEITEM_REQ</td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response Entities</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_REMOVEITEM_RESP</td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
</tbody>
</table>

Entity Hierarchy Display

- **CourseShoppingCart**
  - **CourseShoppingCartItem** (0,Unbouded)
    - **ClassSection** (0,Unbouded)
      - **ClassSectionCombined** (0,Unbouded)
      - **ClassComponent** (0,Unbouded)
        - **ClassSectionComponent** (0,Unbouded)
          - **ClassSectionMeetingPattern** (0,Unbouded)
      - **ClassSectionMeetingPattern** (0,Unbouded)
    - **ClassSectionTextbook** (0,Unbouded)
      - **ClassSectionTextbookDetails** (0,Unbouded)
    - **ClassSectionEnrlInfo** (0,Unbouded)

The Request and Response entities of SCC_SC_REMOVEITEM service operation will have the same structure since both represent the Items on course shopping cart. In the above entity hierarchy, ClassSection entity has been defined as the child of CourseShoppingCartItem so this will reflect both in request as well as in the response entity structure. The request xml need not include ClassSection at all since it represents the class section details of the CourseShoppingCartItem and hence it will only be part of the response xml.
Service Operation: SCC_SC_VALIDATEITEM

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_VALIDATE_REQ</td>
<td></td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Response Entities: NA

Entity Hierarchy Display:

- **CourseShoppingCart**
  - **CourseShoppingCartItem** (0 Unbounded)
    - **ClassSection** (0 Unbounded)
      - **ClassSectionCombined** (0 Unbounded)
      - **ClassComponent** (0 Unbounded)
        - **ClassSectionComponent** (0 Unbounded)
          - **ClassSectionMeetingPattern** (0 Unbounded)
          - **ClassSectionTextbook** (0 Unbounded)
          - **ClassSectionTextbookDetails** (0 Unbounded)
          - **ClassSectionEnrlInfo** (0 Unbounded)

In the above diagram, ClassSection has been defined as a child of CourseShoppingCartItem, but the request xml should not include this and should only include the elements of CourseShoppingCart and CourseShoppingCartItem.

Service Operation: SCC_SC_CHECKOUT

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_CHECKOUT_REQ</td>
<td></td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
In the above diagram, ClassSection has been defined as a child of CourseShoppingCartItem, but the request xml should not include this and should only include the elements of CourseShoppingCart and CourseShoppingCartItem.

**Service Operation: SCC_SC_SAVECART**

<table>
<thead>
<tr>
<th>Request Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_SAVECART_REQ</td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response Entities</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SCC_SC_SAVE_CART_RESP</td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLASS_TXB</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrlInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
</tbody>
</table>
The Request and Response entities of SCC_SC_SAVECART service operation will have the same structure since both represent the Items on course shopping cart. In the above entity hierarchy, ClassSection entity has been defined as the child of CourseShoppingCartItem so this will reflect both in request as well as in the response entity structure. The request xml need not include ClassSection at all since it represents the class section details of the CourseShoppingCartItem and hence it will only be part of the response xml.

**Enrollment Operations**

**Service Operation: SSR_ENR_VALIDATE**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseShoppingCart</td>
<td>SCC_SHOPCRT_WRK</td>
<td>SSR_ENR_VALIDATE_REQ</td>
</tr>
<tr>
<td>CourseShoppingCartItem</td>
<td>SSR_REGFORM</td>
<td></td>
</tr>
<tr>
<td>Class Section</td>
<td>SSR_CLSSRCH_INF</td>
<td></td>
</tr>
<tr>
<td>Class SectionCombined</td>
<td>SSR_CLSSRCH_CMB</td>
<td></td>
</tr>
<tr>
<td>Class Component</td>
<td>SSR_CLSSRCH_COM</td>
<td></td>
</tr>
<tr>
<td>ClassSectionComponent</td>
<td>SSR_CLS_TBL_SEC</td>
<td></td>
</tr>
<tr>
<td>ClassSectionMeetingPattern</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbook</td>
<td>SSR_CLSSRCH_MTG</td>
<td></td>
</tr>
<tr>
<td>ClassSectionTextbookDetails</td>
<td>SSR_CLTXBDTL_VW</td>
<td></td>
</tr>
<tr>
<td>ClassSectionEnrInfo</td>
<td>SSR_CLSRCH_ENRL</td>
<td></td>
</tr>
</tbody>
</table>

Response Entities: NA

**Entity Hierarchy Display**

- **CourseShoppingCart**
  - **CourseShoppingCartItem** (0, Unbounded)
    - **ClassSection** (0, Unbounded)
      - **ClassSectionCombined** (0, Unbounded)
      - **ClassComponent** (0, Unbounded)
        - **ClassSectionComponent** (0, Unbounded)
          - **ClassSectionMeetingPattern** (0, Unbounded)
      - **ClassSectionMeetingPattern** (0, Unbounded)
      - **ClassSectionTextbook** (0, Unbounded)
        - **ClassSectionTextbookDetails** (0, Unbounded)
      - **ClassSectionEnrInfo** (0, Unbounded)

In the above diagram, ClassSection has been defined as a child of CourseShoppingCartItem, but the request xml should not include
this and should only include the elements of CourseShoppingCart and CourseShoppingCartItem.

**Service Operation: SSR_ADD_ENROLLMENT**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnrollmentRequestHdr</td>
<td>ENRL_REQ_HEADER</td>
<td>SSR_ADD_ENROLLMENT_REQ</td>
</tr>
<tr>
<td>EnrollmentRequestDetail</td>
<td>ENRL_REQ_DETAIL</td>
<td></td>
</tr>
</tbody>
</table>

**Service Operation: SSR_DROP_ENROLLMENT**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnrollmentRequestHdr</td>
<td>ENRL_REQ_HEADER</td>
<td>SSR_DROP_ENROLLMENT_REQ</td>
</tr>
<tr>
<td>EnrollmentRequestDetail</td>
<td>ENRL_REQ_DETAIL</td>
<td></td>
</tr>
</tbody>
</table>

**Service Operation: SSR_SWAP_ENROLLMENT**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnrollmentRequestHdr</td>
<td>ENRL_REQ_HEADER</td>
<td>SSR_SWAP_ENROLLMENT_REQ</td>
</tr>
<tr>
<td>EnrollmentRequestDetail</td>
<td>ENRL_REQ_DETAIL</td>
<td></td>
</tr>
</tbody>
</table>

**Service Operation: SSR_EDIT_ENROLLMENT**

<table>
<thead>
<tr>
<th>Entity</th>
<th>DB Record</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnrollmentRequestHdr</td>
<td>ENRL_REQ_HEADER</td>
<td>SSR_EDIT_ENROLLMENT_REQ</td>
</tr>
<tr>
<td>EnrollmentRequestDetail</td>
<td>ENRL_REQ_DETAIL</td>
<td></td>
</tr>
</tbody>
</table>

**Entity Hierarchy Display**

- EnrollmentRequestHdr
  - EnrollmentRequestDetail (0..Unbounded)