1 Document Information .......................................................... 3
2 Introduction ........................................................................... 4
  2.1 Purpose of this document .................................................. 4
  2.2 Out of Scope ...................................................................... 4
  2.3 Target Audience ............................................................... 4
  2.4 Disclaimer ......................................................................... 4
  2.5 Version Control ................................................................. 4
3 HEIMS Technical Architecture .................................................. 5
  3.1 Web Services Architecture Overview .................................. 5
  3.2 Technical Requirements ..................................................... 5
  3.3 Compression ..................................................................... 6
  3.4 Security ............................................................................. 7
  3.5 Transactions Overview ...................................................... 7
    3.5.1 Real-Time Transactions ................................................. 7
    3.5.2 Batch Transactions ...................................................... 9
  3.6 Service Agreements and Availability .................................. 10
  3.7 Request and Response Definitions ..................................... 11
    3.7.1 Request Identifiers ...................................................... 11
    3.7.2 Request Object Schema .............................................. 12
    3.7.3 Response Schema ..................................................... 12
    3.7.4 Control Data .............................................................. 13
    3.7.5 Transaction Data ....................................................... 15
    3.7.6 Messages .................................................................. 16
  3.8 Error Handling ................................................................. 17
    3.8.1 Schema Errors .......................................................... 17
    3.8.2 Business Errors ......................................................... 18
4 HEIMS Web Service Methods .................................................. 19
  4.1 General Information .......................................................... 19
  4.2 Real-Time Student CHESSN Allocation .............................. 20
  4.3 Student CHESSN Allocation (Batch) ................................... 27
  4.4 Student CHESSN Allocation Results (Batch)....................... 30
  4.5 Retrieving Batch Call Status ............................................. 32
  4.6 Real-Time Student Entitlement Details ................................. 34
  4.7 Student Entitlement Details (Batch) .................................... 38
  4.8 Student Entitlement Detail Results (Batch) ......................... 40
  4.9 Security – Changing Password .......................................... 43
  4.10 Ping ............................................................................... 44
5 Environments .......................................................................... 45
  5.1 Production ......................................................................... 45
  5.2 Next Production Release .................................................... 45
6 Certificate .............................................................................. 46
7 Troubleshooting ...................................................................... 46
A. Appendix A – References and Other Relevant Documents ........ 47
  A.1 References ....................................................................... 47
  A.2 Other Relevant Documents .............................................. 48
B. Appendix B – Glossary ............................................................ 49
C. Appendix C – List of Business Messages ................................. 50
  C.1 Request Error Messages .................................................. 50
  C.2 Security Related ............................................................ 50
  C.3 Business Messages .......................................................... 51
D. Appendix D – Web Service Schemas ....................................... 53
  D.1 Business.xsd .................................................................... 53
  D.2 AllocateChessn.xsd ........................................................ 80
  D.3 AllocateChessnBatch.xsd .................................................. 82
  D.4 AllocateChessnRealTime.xsd ............................................. 83
  D.5 GetEntitlement.xsd .......................................................... 84
  D.6 GetEntitlementBatch.xsd ................................................... 85
  D.7 GetEntitlementRealTime.xsd ............................................ 86
E. Appendix E – CHESSN Check Digit Algorithm ......................... 87
F. Appendix F – Business Rules ................................................... 89
G. Appendix G – Business Fields ............................................... 90
1 Document Information
This document approved for public release.
2 Introduction

2.1 Purpose of this document
The purpose of this document is to provide technical information required to make web service calls between the Higher Education Information Management System (HEIMS) and Higher Education Providers (HEPs) / Tertiary Admission Centres (TACs) / VET Providers. The scope of the document includes technical specifications of the web services, description of the web methods, the schemas used and troubleshooting information.

Due to the flexibility of web services and the large number of systems they can run on, it is not feasible to provide detailed implementation instructions for specific platforms and systems. This document will however, provide what technical information is needed in order to configure and use a web services toolkit.

2.2 Out of Scope
The following are out of scope for this document:

- Detailed business rules on data matching on student information; and
- Technical specification for any of the W3C standards used in HEIMS.

A list of documentation that addresses W3C standards and context information for HEIMS Web services (such as business requirements) can be found in Appendix A.

2.3 Target Audience
This is a technical document aimed at:

- Higher Education Providers
- Tertiary Admission Centres
- VET Providers
- Third-Party software developers for HEPs, TACs and VET Providers.

2.4 Disclaimer
These specifications give information about how to use the HEIMS Web Services. These specifications are not intended to provide implementation instructions for individual systems. The department accepts no responsibility for any loss or damage to any system resulting from the use of these specifications.

These specifications may be changed from time to time. It is the responsibility of HEPs, TACS, VET Providers and others using these specifications to ensure they are using the latest version.

2.5 Version Control
Documenting changes to this document will be managed through controlled versioning. Versioning will be done at two levels: the document level and the XML Schema Level. It should be noted that minor revisions to the XML Schema will not result in a reissuing of this document.
3 HEIMS Technical Architecture

3.1 Web Services Architecture Overview
XML Web Services provide a mechanism for applications to exchange information over a network. By providing a standard interface and communicating using international standard protocols, all Web service implementations operate in the same manner making communicating using web services a simple, open and platform independent process.

Web service interfaces are described by the Web Services Definition Language (WSDL) and web service communication relies on protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP), Hypertext Transfer Protocol (HTTP), and Simple Object Access Protocol (SOAP). These standards are developed and maintained by the World Wide Web Consortium (W3C), a member organisation consisting of leading technology vendors, corporate users, standards bodies and government organisations. As such, this provides a set of rich standards which are not tied to a specific vendor or system. For these reasons, XML Web Services are suitable as a means for HEIMS to communicate with external systems.

3.2 Technical Requirements
HEIMS Web services will only cater for requests using SOAP 1.1. Web service calls using HTTP-GET and HTTP-POST will not be supported. This is because using SOAP provides XML schema support for more complex data types. The transport method supported is Transport Layer Security (TLS1.2 and above).

From 01 October 2018 onwards, communications through transport methods SSL3.0, TLS1.0 and TLS1.1 is no longer supported.

In order to call the HEIMS Web services, an application or SDK capable of calling XML web services is required. The application or SDK must support the following:

- XML 1.0;
- SOAP 1.1;
- HTTP 1.1;
- Basic Authentication; and
- Server Name Identification.

When implementing a responsive application, bandwidth requirements must be taken into consideration. Bandwidth requirements depend on many factors. These include:

- Size of the payload sent to HEIMS for processing;
- Frequency of the requests; and
- Data compression.

HEIMS web services have been designed to minimise the network traffic payload as much as possible. The services provided by HEIMS are therefore not bandwidth intensive. However, to ensure best performance, the department recommends a broadband connection for both upstream and downstream traffic. Minimum bandwidth recommendations will be confirmed during performance testing.
3.3 Compression

Compression is an optional feature for most HEIMS Web services that can reduce the bandwidth requirements. To find out which methods support compression and which ones don't, refer to individual method descriptions in section 4. For clients with small bandwidth capabilities or those that process large volumes of requests, this may be a useful option as the bandwidth savings can be quite large. Typical compression ratios of 6:1 can be achieved\(^1\). Using compression reduces not only bandwidth requirements, but other side effects of large requests such as timeouts. The compression algorithm supported by HEIMS Web Services is *gzip*.

In order to use compression, the tools used to call HEIMS Web services must be able to control how and when SOAP messages are created and sent. Compression of the stream must take place between the SOAP layer and the HTTP layer. That is, after the SOAP envelope has been properly constructed, the stream must be compressed and then sent over HTTP to the server. To inform the server that the call is a compressed one, the following HTTP headers must be added:

<table>
<thead>
<tr>
<th>HTTP Header</th>
<th>Value Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Encoding</td>
<td>gzip</td>
<td>Set this HTTP header if the request stream is compressed.</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>gzip</td>
<td>Set this HTTP header if you wish the server to return a compressed stream as its response.</td>
</tr>
</tbody>
</table>

\(^1\) Based on a call to the *AllocateStudentCHESSN* web method containing transaction data for 10,000 students. Data for each student was non-repeating. The SOAP envelope in the requesting call was compressed using *gzip*. Compression ratios are data dependent and may vary.
If the server detects that the `Content-Encoding` HTTP header has a value of ‘gzip’, it will treat the request stream as being compressed. The server will then uncompress the stream into the original SOAP message and process it as normal. If the server also detects that the `Accept-Encoding` HTTP header has a value of ‘gzip’, the SOAP response will be compressed before it is sent back to the client.

### 3.4 Security

HEIMS Web services use Basic authentication over SSL for secure communication. Basic authentication is widely supported and part of the HTTP 1.1 standard. In Basic Authentication, the client provides the username and password, which is Base64 encoded and sent directly to server. Encryption of the communication channel is then handled by the SSL protocol.

The following security rules apply to login passwords for HEIMS Web services:

- Passwords will expire within 30 days;
- The last 10 passwords are recorded so that passwords cannot be reused;
- Strong passwords will be used – They must contain a combination of upper and lowercase characters, numbers and special characters (eg. #, @, $);
- Password must be between 7 and 15 characters long;
- After a password has been successfully changed by a user, it cannot be changed again within a 24-hour period except by the HEIMS IT Liaison officer.
- The maximum number of failed logon attempts before the account will be locked is 3;
- If the account is locked, the HEIMS IT Liaison officer (HEIMS.datacollections@education.gov.au) must be contacted to reset the password; and
- After a password has been reset, it must be changed by calling the `ChangePassword` method before any other HEIMS Web methods are called. The number of days elapsed since the password was changed is then reset to zero.

Change of password can be done via the `ChangePassword` web method described in section 4.9.

### 3.5 Transactions Overview

HEIMS provides two types of Web services: real-time and batch. The main difference is that real-time Web services are processed immediately by the server, whereas batch Web services are queued on the server to be processed later. Some methods (such as CHESSN allocation) will have a real-time and a batch version provided. Both versions perform exactly the same business function. Where both types are provided for a particular business function, which version to use is entirely up to the client and will depend on the clients business requirements.

#### 3.5.1 Real-Time Transactions

Real-time transactions are the simpler of the two types of transactions. A real-time method will perform the requested action on the server immediately and return the results as part of the call. Real-time requests can only send one transaction for processing at a time. Because of this, real-time calls should only be used when a single result is required immediately. For instance, when an application, assessment and offer is made at the university on the spot, the system may be required to allocate a CHESSN immediately. In this case the real-time method would be the appropriate one to call.
Figure 1: Message Flow in Real-Time Transactions

(1a) HEP/TAC/VET Provider calls HEIMS
Passing a unique RequestId

(1b) HEIMS replies acknowledgement, processes
3.5.2 Batch Transactions

Batch transactions are those that, due to the volume of data being sent, or the amount of processing required will be placed in a queue for later processing. They are used primarily where an immediate response is not required. Batch methods can contain requests for multiple transactions and therefore can be very large in size.

For instance, when an HEP or TAC or VET Provider wants to allocate CHESSNs for all continuing students, the volume of data send to the server will be large. This will take some time for the server to process and in addition, the resulting CHESSNs are not urgently required because the students are already enrolled. In such a case, the use of batch transactions would be ideal.

An invocation of a batch transaction can actually comprise of three method calls: The first call contains any request-specific information plus any transaction data. The server accepts the request and marks it for processing. The second (and optional) call polls the server and checks if processing for this request is complete. The third and final call queries the server for the results of the job. If the job has completed, the results are returned back to the caller. Otherwise, the server will inform the client that the original request is still being processed.

![Message Flow in Batch Transactions](image)

Figure 2: Message Flow in Batch Transactions
3.6 Service Agreements and Availability

The department is committed to providing a high performance, reliable, available and fully supported HEIMS production environment. The department will provide the following environment to all approved HEIMS users:

- **Batch Transactions** – 95% of Batches will be processed within a maximum of 24 hours normal working days, from receipt by the department provided no more than 500,000 transactions in total are submitted by the sector per day. 10,000 compressed/1,000 uncompressed transactions will be allowed.
- **Real Time transactions** – 95% of transactions within 5 seconds turnaround measured from receipt by the department to the time the response leaves the department’s firewall.

These response figures do not include network transport times. In other words, the response times given are from the time the department servers receive the request to the time the results are sent back.

It is intended that HEIMS Web Services will be available 24 hours a day, 7 days a week with the following possible exceptions:

- A weekly infrastructure maintenance window on Thursday evenings, 7.00pm-12.00am AEST/AEDT. HEIMS Web Services will generally be available during this period but interruptions to server availability may occur;
- A production migration window on Wednesday evenings, 8.00pm-11.00pm AEST/AEDT with 1 week notification by the department. System availability in this period will depend on the number and type of production migrations required; and
- A HEIMS system maintenance window as required but on Saturdays with 1 week notification by the department. System availability in this period will depend on the amount of production maintenance required.

The department will endeavour to minimise all system maintenance during peak enrolment periods and keep unscheduled down-times to an absolute minimum.
3.7 Request and Response Definitions

The following object types are used generally in all HEIMS Web service calls:

- Response objects and request objects;
- Response control tables and Request control tables; and
- Response transaction data and request transaction data.

The following table outlines where and when each structure is used:

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Input Type</th>
<th>Return Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time Call</td>
<td>Request object</td>
<td>Response object</td>
</tr>
<tr>
<td>Batch Call</td>
<td>Request object</td>
<td>ResponseControlTable</td>
</tr>
<tr>
<td>Batch Status Call</td>
<td>RequestControlTable</td>
<td>ResponseControlTable</td>
</tr>
<tr>
<td>Batch Results Call</td>
<td>RequestControlTable</td>
<td>Response object</td>
</tr>
</tbody>
</table>

Note: In this and following sections, element fields will be defined in terms of XML data types. If an XML data type has a prefix of xs, it refers to a built-in XML data type that is part of the http://www.w3.org/2001/XMLSchema namespace. If the data type has a heims prefix, it refers to elements that are part of the http://dest.gov.au/Heims/ namespace. Full schema definitions for all HEIMS data types can be found in Appendix D.

3.7.1 Request Identifiers

Request identifiers (RequestId) are an important part of HEIMS web services. They are included in every web method call to identify an individual request. The length of the RequestId must be between 1 and 36 characters, and can be of any format. An obvious choice is to use a Globally Unique Identifier (GUID), but using an integer starting from 0 and incrementing by 1 with each web method call is also valid.

The value of the RequestId must only be unique within a particular HEP or TAC or VET Provider and it must retain uniqueness over time. It is the responsibility of the HEP or TAC or VET Provider to ensure that all its client systems internally keep track of all previous RequestIds and to synchronise generation of unique RequestIds for new calls.

Every time a method is called, a new RequestId must be sent to the server. When returning the processing results of the request, the server response will include the same RequestId. In order to guarantee that the server does not process the same request twice due to a communication failure, calls to a web method providing an old RequestId will return the results of the old request. No new processing will be performed.

Please note that request results will only be kept for 30 days after processing. After that, the request results will be archived. Any call to retrieve results of an archived request will return no results.
3.7.2 Request Object Schema
The request object is a generic container that holds all the information a client needs to send to a HEIMS Web service. It contains request control data (one RequestControlTable) plus multiple request transaction data elements. If the request is being sent for a real-time method, only one transaction data element is allowed. For batch methods, there is a limit on the number of transactions allowed per request. The limit is subject to service agreements in section 3.6.

```
+RequestControlData[1] : RequestControlTable
+TransactionData[*] : RequestTransactionData
```

Request

```
+RequestId : string
+ClientOrganisationCode : string
+RequestLocalDateTime : DateTime
```

RequestControlTable

```
+RequestField1
+RequestField2
```

Figure 3: General Request Class Diagram

3.7.3 Response Schema
Similarly, the response schema is a container for all information a HEIMS Web service sends to a client. It contains response control data (one ResponseControlTable) plus multiple response transaction data elements.

```
+ResponseControlData[1] : ResponseControlTable
+TransactionData[*] : ResponseTransactionData
```

Response

```
+ResponseField1
+ResponseField2
```

ResponseControlTable

```
+RequestId : string
+ClientOrganisationCode : string
+RequestLocalDateTime : DateTime
```

ResponseTransactionData

```
+Messages[*] : Message
```

```
+Code : int
+Description : string
+Severity : MessageSeverity
```

```
+TransactionStatusCode : TransactionStatusCode
+Messages[*] : Message
```

```
+Status : string
```

```
+TransactionStatus : TransactionStatusCode
```

```
+MessageSeverity : string
```

```
+ERROR
+WARNING
+INFORMATION
```

```
+SUCCESS
+FAILURE
+WARRANTING
```

```
+PROCESS
+FAILURE
+WARRANTING
```

Figure 4: General Response Class Diagram
3.7.4 Control Data

Control data is present in all communication between clients and HEIMS Web services. It provides information concerned with the HEIMS Web service infrastructure and as such, is essential to every HEIMS Web method. There are two control data structures: RequestControlTable and ResponseControlTable.

RequestControlTable

RequestControlTable is a container for all request control data. For every call to a HEIMS web service, a RequestControlTable must always be sent. Depending on the type of method being called (real-time or batch), it can be sent by itself or wrapped within a request object. The RequestControlTable contains the following fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestId</td>
<td>heims:RequestId</td>
<td>The request identifier.</td>
</tr>
<tr>
<td>ClientOrganisationCode</td>
<td>heims:ClientOrganisationCode</td>
<td>The client’s 4-digit identifying code.</td>
</tr>
<tr>
<td>RequestLocalDateTime</td>
<td>xs:dateTime</td>
<td>Date and time on the client at the time of the call.</td>
</tr>
</tbody>
</table>

ResponseControlTable

Similarly, the ResponseControlTable is a container for all response control data. When a client receives a response from a HEIMS Web service, a ResponseControlTable is always present. Again, depending on the method being called (real-time or batch), it can be sent by itself or wrapped in a response object. A ResponseControlTable object contains the following fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestId</td>
<td>heims:RequestId</td>
<td>The same request identifier of the original request.</td>
</tr>
<tr>
<td>ClientOrganisationCode</td>
<td>heims:ClientOrganisationCode</td>
<td>The 4-digit identifying code of the client that issued the original request.</td>
</tr>
<tr>
<td>CurrentDateTime</td>
<td>xs:dateTime</td>
<td>Date and time on the server at the time the response was sent.</td>
</tr>
<tr>
<td>ReceivedDateTime</td>
<td>xs:dateTime</td>
<td>Date and time on the server when the server first received the request.</td>
</tr>
<tr>
<td>RequestStatus</td>
<td>heims:RequestStatus</td>
<td>Contains a RequestStatusCode – the current status of the request, and Messages – a list of messages associated with the request.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>heims:SchemaVersion</td>
<td>The Web services schema version the incoming request was validated against.</td>
</tr>
</tbody>
</table>
The `RequestStatus` element contains two fields, `RequestStatusCode` and `Messages`. `RequestStatusCode` contains information on the status of the request itself. It is not related to the status of the individual transactions. For example, a request can have a `RequestStatusCode` equal to ‘SUCCESS’, but all of its individual transactions can fail business processing. `RequestStatusCode` can take the following values:

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPLICATE</td>
<td>The <code>RequestId</code> of the current request has already been used. Request transaction data will contain the results of the original request. No new processing was performed on the server.</td>
</tr>
<tr>
<td>FAILURE</td>
<td>An error occurred with the request. Request transaction data will not contain results.</td>
</tr>
<tr>
<td>PROCESS</td>
<td>The request is currently being processed or is queued for processing. Request transaction data will not contain results.</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The <code>RequestId</code> of the current request has already been used more than 30 days ago. The server has since then archived the processing results. Request transaction data will not contain results.</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>The request has been processed successfully. Request transaction data will contain processing results.</td>
</tr>
</tbody>
</table>

The `Messages` element contains a list of messages concerned with the request. For example, if `RequestStatusCode` has a value of 'FAILURE', `Messages` will contain descriptions as to why the original request was invalid.
### 3.7.5 Transaction Data

Transaction data contains business information specific to the method being called. Request transaction data contains data required to perform the required business service, while response transaction data contains the results of processing the request. For example, in a method that returns student entitlement information, request transaction data will contain the student’s CHESSN, while response transaction data will contain entitlement information such as Ordinary Student Learning Entitlement usage and FEE-HELP loan balance.

Because each method has different input and output fields, each method defines different request and response transaction data schemas. There are however, two fields that are common across transaction schemas, RecordId and TransactionStatus.

#### Record Identifier

The record identifier (RecordId) is an element common to both request and response transaction data elements. When a batch request contains more than one transaction, the RecordId is used to distinguish between each transaction data element.

The RecordId can be any string up to 20 characters and can take on any format chosen by the client. For batch requests, the value of the RecordId must be unique across all transactions in the request. Because real-time requests contain at most only one transaction, the RecordId will always be unique regardless of the RecordId value.

In the context of a response transaction data element, the RecordId will match that of the corresponding request transaction data element. This allows the client to identify which output record matches up with which input record.

#### TransactionStatus

TransactionStatus is an element common only to response transaction schemas. It provides information about the results of business processing for an individual transaction. The definition for TransactionStatus in Appendix D.1 defines it as having two fields: TransactionStatusCode and Messages. TransactionStatusCode indicates the processing status of the transaction, and Messages is an array of Message elements. Within the context of TransactionStatus, each Message element holds business validation or business error messages for that transaction.

TransactionStatusCode can take on the following values:

<table>
<thead>
<tr>
<th>TransactionStatusCode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESS</td>
<td>The individual transaction processed correctly according to its respective business rules. Other response transaction data fields will contain valid output.</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The individual transaction failed processing according to business rules. Other response transaction data fields will not contain valid output.</td>
</tr>
<tr>
<td>WARNING</td>
<td>The individual transaction processed correctly but with warnings. Other response transaction data fields will contain valid output; however the Messages element will contain warning messages which should be investigated by the client HEP/TAC/VET Provider in order to maintain data and processing integrity.</td>
</tr>
</tbody>
</table>
3.7.6 Messages
As mentioned in the previous sections, both the RequestStatus and TransactionStatus fields contain the Messages element. This element is used to hold any messages concerned with either the request or its constituent transactions. The Messages element itself only contains an array of individual Message elements. Each Message element contains three fields: Code, Description and MessageSeverity. The Code and Description elements will match those listed in Appendix C. MessageSeverity can take on the following values:

<table>
<thead>
<tr>
<th>MessageSeverity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFORMATION</td>
<td>The message is of an informational nature.</td>
</tr>
<tr>
<td>WARNING</td>
<td>The message provides a warning which should be investigated. When a warning message is produced, it usually implies that the request or transaction was successful, but with caveats.</td>
</tr>
<tr>
<td>ERROR</td>
<td>The message is produced when the request or transaction has failed. It contains reasons as to why it failed and what caused the error.</td>
</tr>
</tbody>
</table>
### 3.8 Error Handling

#### 3.8.1 Schema Errors

Every request received by a HEIMS Web service is validated against the respective schema for that method. The validation performs checks to do with the format and structure of the request fields. If the request does not comply with the schema, the returning SOAP message will not contain any response data, only a SOAP fault. An example of a server response to an invalid request would look like:

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>Schema Validation Error(s)</faultstring>
      <detail>
        <ValidationError>
          <RecordId>211</RecordId>
          <Element>PostCode</Element>
          <Line>10</Line>
          <Column>3</Column>
          <Description>Invalid value according to its data type</Description>
        </ValidationError>
        <ValidationError>
          <RecordId>211</RecordId>
          <Element>FamilyName</Element>
          <Line>23</Line>
          <Column>5</Column>
          <Description>Element content is incomplete according to the DTD/Schema.</Description>
        </ValidationError>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Within the `detail` element, HEIMS will place a set of `ValidationError` elements, one for each error encountered. The `detail` element contains the following fields:

<table>
<thead>
<tr>
<th>SOAP Fault Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordId</td>
<td>This refers to the <code>RecordId</code> of the transaction data element that contains the error. This field applies to errors with transaction data only. Errors concerned with control data or errors to do with the entire request (eg. duplicate <code>RecordId</code> values) will result in an empty <code>RecordId</code> value.</td>
</tr>
<tr>
<td>Element</td>
<td>The element in which the error occurred.</td>
</tr>
<tr>
<td>Line</td>
<td>The line number of the incoming request where the error occurred.</td>
</tr>
<tr>
<td>Column</td>
<td>The column number of the incoming request where the error occurred.</td>
</tr>
<tr>
<td>Description</td>
<td>A brief description of why the element was invalid.</td>
</tr>
</tbody>
</table>

Line and column number will only be useful if the tools used to call Web services allow access to view the raw SOAP request sent over the wire. In any case, the other fields (`RecordId`, `Element` and `Description`) should be sufficient to pinpoint where the error occurred in the request.
Errors in the request control data (*RequestControlTable*) will return a SOAP fault immediately and no input transaction data will be validated. If any schema validation errors are detected, HEIMS will not store the request which means the same *RequestId* can be re-used.

### 3.8.2 Business Errors

Once the incoming request has validated successfully against the schema, the server will process each transaction according to the respective business rules. Any errors or messages for a transaction data element arising from this validation are stored in the *TransactionStatus* field. A list of business rules and messages for each method can be found in Appendix C.
4 HEIMS Web Service Methods

4.1 General Information
This section provides a list of all available web services and the methods associated with them. It also lists data types used and relevant business rules.

Note: Before calling a HEIMS Web method for the first time, the password must be changed by invoking the ChangePassword method described in section 4.9. If not, authentication for the method will fail with a message indicating a change of password is required. In addition, every time a password is reset manually the ChangePassword method will need to be called before calling any other HEIMS Web methods.

The following sections describe the web methods provided by HEIMS web services:

Student CHESSN Allocation
Section 4.2 Real-Time Student CHESSN Allocation
Section 4.3 Batch Student CHESSN Allocation
Section 4.4 Batch Student CHESSN Allocation Results

All Batch Methods
Section 4.5 Retrieving Batch Call Status

Retrieving Student Entitlement
Section 4.6 Real-Time Student Entitlement
Section 4.7 Batch Student Entitlement
Section 4.8 Batch Student Entitlement Results

Security
Section 4.9 Change Password

Other
Section 4.10 Ping
4.2 Real-Time Student CHESSN Allocation

**End point URL:** https://app.heim.education.gov.au/WebServices.CHESSN/service.asmx

**WSDL Location:** https://app.heim.education.gov.au/WebServices.CHESSN/service.asmx?WSDL

**Method Signature**

AllocateChessnResponse AllocateStudentChessn (AllocateChessnRequest chessnRequest)

**Description**

This method provides a real-time service to allocate a CHESSN for a new or continuing student. The server will perform data matching on the given student fields and immediately assign a new CHESSN or return the existing CHESSN if the student already exists within HEIMS.

The following information is retrieved for the student as the result of a successful CHESSN allocation:

- Commonwealth Higher Education Student Support Number (CHESSN) (G61).
- Ordinary Student Learning Entitlement (SLE) limit (G898)
- Ordinary Student Learning Entitlement (SLE) usage. (G746)
- Ordinary Student Learning Entitlement (SLE) balance (G747)
- Ordinary Student Learning Entitlement (SLE) ‘As at’ date. (G738)
- Student FEE-HELP limit (G669)
- Student FEE-HELP usage (G749)
- Student FEE-HELP Loan Balance (G129, G768, G757)
- Student FEE-HELP’As at’ date. (G738)
- Student Ordinary limit (G985)
- Student Ordinary usage (G1035)
- Student Ordinary balance (G989)
- Student Associate limit (G986)
- Student Associate usage (G1036)
- Student Associate balance (G990)
- Student Enabling limit (G987)
- Student Enabling usage (G991)
- Student Enabling balance (G991)
- Student Indigenous limit (G988)
- Student Indigenous usage (G999)
- Student Indigenous balance (G992)
- Student Commonwealth Scholarships ‘As at’ date. (G738)
- Student OS-HELP limit (G945)
- Student OS-HELP usage (G944)
- Student OS-HELP balance (G946)
- Student OS-HELP ‘As at’ date. (G738)

Please consult Appendix G – Business Fields for a detailed description of the meaning and usage of these fields.

**Note:** If two real-time CHESSN requests containing the same student data are sent within a very short time period of each other, there is a chance that after the first request is processed, the data matching component of the system will not have enough time to update its search repository before the second request is received. This situation could result in both requests being returned different CHESSNs even though the student data was exactly the same. To avoid this situation, two real-time CHESSN requests containing the same student data should not be sent within a short time span of each other.

**Supports Compression**

Yes
Parameter List

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| chessnRequest   | heims:AllocateChessnRequest | The real-time request object. The AllocateChessnRequest schema defines one RequestControlTable and one AllocateChessnIn element. Each AllocateChessnIn element contains student information required for CHESSN allocation.  
Please refer to:  
Appendix D.4 – Schema definition for AllocateChessnRequest  
Appendix D.2 – Schema definition for AllocateChessnIn  
Appendix D.1 – Schema definition for RequestControlTable |

Business Rules

According to the AllocateChessnIn schema in Appendix D.2, the RecordId, FamilyName, BirthDate and SexCode fields are always mandatory while the GivenName, OtherGivenName and PreviousNames fields are always optional. For the rest of the fields, the following table lists which are mandatory and under which conditions:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Rule for New Applicants</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CitizenshipCode</td>
<td>Element must be present.</td>
<td>10201</td>
</tr>
<tr>
<td>PostalAddress</td>
<td>Element must be present.</td>
<td>10201</td>
</tr>
<tr>
<td>AttendedYear12Code</td>
<td>Element must be present.</td>
<td>10201</td>
</tr>
<tr>
<td>AttendedPreviousHepCode</td>
<td>Element must be present.</td>
<td>10201</td>
</tr>
</tbody>
</table>

Although the schema definition for AllocateChessnIn specifies the fields in the above table as optional, they will be checked by the business rules engine. In addition to rules on which element must present for which type of student, the following rules apply to the value of individual AllocateChessnIn elements:

**NOTE:** Please consult Business Rule 389 contained in Appendix F – Business Rules for additional business rules pertaining to this method.
<table>
<thead>
<tr>
<th>Data Element</th>
<th>Business Rule</th>
<th>Applicable Message Codes</th>
<th>Applicable to</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Continuing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Students</td>
</tr>
<tr>
<td>BirthDate</td>
<td>The value of the BirthDate must be greater than the current server date minus</td>
<td>10203</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>110 years AND the value of the BirthDate must be less than the current server</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>date.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SexCode</td>
<td>A student record must have a SexCode equal to ‘M’, ‘F’ or ‘X’.</td>
<td>10259</td>
<td>Yes</td>
</tr>
<tr>
<td>FamilyName</td>
<td>Within each PreviousName element, FamilyName is mandatory.</td>
<td>10201</td>
<td>Yes</td>
</tr>
<tr>
<td>CitizenshipStatus</td>
<td>A student must have a CitizenshipStatusCode equal to ‘1’, ‘2’, ‘3’, ‘4’,</td>
<td>10205</td>
<td>Yes</td>
</tr>
<tr>
<td>Code</td>
<td>‘5’ or ‘8’.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year12Details</td>
<td>If the value of AttendedYear12Code is equal to ‘AttendedYear12’, THEN</td>
<td>10224</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Year12Details must be present.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year12State</td>
<td>If the value of AttendedYear12Code is equal to ‘AttendedYear12’ THEN</td>
<td>10207</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>For each record supplied in Year12Details, Year12State must be supplied.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year12State</td>
<td>If the value of AttendedYear12Code is equal to ‘AttendedYear12’, THEN</td>
<td>10208</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>For each record supplied in Year12Details, Year12State must be a valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Australian state according to the Code Value Source on the CHESSN structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year12Year</td>
<td>If the value of AttendedYear12Code is equal to ‘AttendedYear12’, THEN</td>
<td>10209</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>For each record supplied in Year12Details, Year12Year must be supplied.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Element</td>
<td>Business Rule</td>
<td>Applicable Message Codes</td>
<td>Applicable to Continuing Students</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| Year12Year            | If the value of `AttendedYear12Code` is equal to 'AttendedYear12',  

**THEN**  

For each record supplied in `Year12Details`,  

`Year12Year` must be greater than the current server date minus 100 years  

**AND**  

`Year12Year` must be less than or equal to the current server date year. | 10225 | Yes |
| Year12StudentNumber   | For TACs only. If the value of `AttendedYear12Code` is equal to 'AttendedYear12',  

**THEN**  

For each record supplied in `Year12Details`,  

`Year12StudentNumber` and `Year12SchoolName` cannot both be omitted. One of these data elements must be specified. | 10210 | Yes |
| Year12SchoolName      | For TACs only. If the value of `AttendedYear12Code` is equal to 'AttendedYear12',  

**THEN**  

For each record supplied in `Year12Details`,  

`Year12StudentNumber` and `Year12SchoolName` cannot both be omitted. One of these data elements must be specified. | 10226 | Yes |
| PreviousHepDetails    | If the value of `AttendedPreviousHepCode` is equal to 'AttendedPreviousHep',  

**THEN**  

PreviousHepDetails must be present. | 10228 | Yes |
| HepCode               | If the value of `AttendedPreviousHepCode` is equal to 'AttendedPreviousHep',  

**THEN**  

For each record supplied in `PreviousHepDetails`, the value of `HepCode` must be valid according to Appendix A on HEIMSHELP. | 10214 | Yes |
| HepName               | If the value of `AttendedPreviousHepCode` is equal to 'AttendedPreviousHep'  

**THEN**  

For each record supplied in `PreviousHepDetails`, `HepName` and `HepStudentNumber` cannot both be omitted. One of those data elements must be specified. | 10214 | Yes |
<table>
<thead>
<tr>
<th>Data Element</th>
<th>Business Rule</th>
<th>Applicable Message Codes</th>
<th>Applicable to Continuing Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>HepYear</td>
<td>If the value of <code>AttendedPreviousHepCode</code> is equal to 'AttendedPreviousHep', THEN For each record supplied in <code>PreviousHepDetails</code>, <code>HepYear</code> must be present.</td>
<td>10215</td>
<td>Yes</td>
</tr>
<tr>
<td>HepYear</td>
<td>If the value of <code>AttendedPreviousHepCode</code> is equal to 'AttendedPreviousHep', THEN For each record supplied in <code>PreviousHepDetails</code>, <code>HepYear</code> must be less than or equal to the current year AND <code>HepYear</code> must be greater than the current year minus 100 years.</td>
<td>10229</td>
<td>Yes</td>
</tr>
<tr>
<td>ClientOrganisationCode</td>
<td>Only a valid Higher Education Provider can request a CHESSN for a continuing student.</td>
<td>10260</td>
<td>Yes</td>
</tr>
<tr>
<td>PostCode</td>
<td>Within the <code>PostalAddress</code> element, if <code>CountryCode</code> is equal to ‘1100’ (Australia) THEN <code>PostCode</code> must be present.</td>
<td>10218</td>
<td>No</td>
</tr>
<tr>
<td>PostCode</td>
<td>Within the <code>PostalAddress</code> element, if <code>CountryCode</code> is provided AND <code>CountryCode</code> is equal to ‘1100’ (Australia), THEN <code>PostCode</code> must be a value in the range of ‘0001’ to ‘9999’.</td>
<td>10230</td>
<td>Yes</td>
</tr>
<tr>
<td>CountryCode</td>
<td>Within the <code>PostalAddress</code> element, if the <code>CountryCode</code> is provided THEN <code>CountryCode</code> must be valid according to Appendix C on HEIMSHELP.</td>
<td>10216</td>
<td>Yes</td>
</tr>
</tbody>
</table>
CHESSN Duplicate Handling

With Release 1.1.0 of the HEIMS CHESSN Interface software, the department is able to administer Type 1 CHESSN duplicates identified through a manual form-based procedure for TAC, HEP and VET Provider notification of CHESSN duplicates to the department.

Business rules are as follows:

<table>
<thead>
<tr>
<th>Entitlement information returned for CHESSNs duplicates</th>
<th>Business Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>The entitlement information returned for the CHESSN Allocation process will include entitlement data for the master CHESSN as well as any entitlement data that was attached to related duplicate CHESSNs.</td>
<td></td>
</tr>
</tbody>
</table>

Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>heims:AllocateChessnResponse</td>
<td>The real-time response object. The AllocateChessnResponse schema defines one ResponseControlTable and one AllocateChessnOut element. The AllocateChessnOut element will contain processing results including the allocated CHESSN and entitlement details. Please refer to: Appendix D.4 – Schema definition for AllocateChessnResponse Appendix D.2 – Schema definition for AllocateChessnOut Appendix D.1 – Schema definition for ResponseControlTable</td>
</tr>
</tbody>
</table>

Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the ClientOrganisationCode provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>FAILURE</td>
<td>A duplicate RequestId was provided, but the RequestId was not for a real-time AllocateStudentChessn method.</td>
<td>10</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestDateTime is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>DUPLICATE</td>
<td>The RequestId matches a previous real-time AllocateStudentChessn request. The same results will now be returned. No new processing was performed.</td>
<td>8</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. Processing results are not returned.</td>
<td>9</td>
</tr>
<tr>
<td>RequestStatusCode</td>
<td>Condition</td>
<td>Applicable Message Codes</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Processing for this request is complete and will be returned.</td>
<td></td>
</tr>
</tbody>
</table>
| SUCCESS           | Condition: All active CHESSNs.  
Message: ‘The available SLE for this active CHESSN is subject to any enrolments and variations that may be applicable subsequent to the ‘As at’ date.’ | 10036                   |
| SUCCESS           | Condition: For active CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa).  
Message: ‘The FEE-HELP balance for this active CHESSN is subject to any enrolments, variations, and indexation that may be applicable subsequent to the ‘As at’ date.’ | 10033                   |
| SUCCESS           | Condition: For provisional CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa) or ‘3’ (Permanent visa holder other than humanitarian visa) or ‘X’ or NULL.  
Message: ‘The FEE-HELP balance may be subject to usage not yet reported to the department and should be checked with the student.’ | 10034                   |
4.3 Student CHESSN Allocation (Batch)
This method submits a batch request for allocating student CHESSNs.

**End point URL:** https://app.heim.education.gov.au/WebServices.CHESSN/Batch/service.asmx


**Method Signature**
ResponseControlTable AllocateStudentChessn (AllocateChessnBatchRequest chessnRequest)

**Description**
This method sends a request to allocate CHESSNs for a set of students. The set of students can be new, continuing or a mixture. The server will perform data matching on the given student fields and assign a new CHESSN or return the existing CHESSN if the student already exists within HEIMS.

The following information is retrieved for the student as the result of a successful CHESSN allocation:

- Commonwealth Higher Education Student Support Number (CHESSN) (G61).
- Ordinary Student Learning Entitlement (SLE) limit (G898)
- Ordinary Student Learning Entitlement (SLE) usage. (G746)
- Ordinary Student Learning Entitlement (SLE) balance (G747)
- Ordinary Student Learning Entitlement (SLE) ‘As at’ date. (G738)
- Student FEE-HELP limit (G669)
- Student FEE-HELP usage (G749)
- Student FEE-HELP Loan Balance (G129, G768, G757)
- Student FEE-HELP ‘As at’ date. (G738)
- Student Ordinary limit (G985)
- Student Ordinary usage (G1035)
- Student Ordinary balance (G989)
- Student Associate limit (G986)
- Student Associate usage (G1036)
- Student Associate balance (G990)
- Student Enabling limit (G987)
- Student Enabling usage (G998)
- Student Enabling balance (G991)
- Student Indigenous limit (G988)
- Student Indigenous usage (G999)
- Student Indigenous balance (G992)
- Student Commonwealth Scholarships ‘As at’ date. (G738)
- Student OS-HELP limit (G945)
- Student OS-HELP usage (G944)
- Student OS-HELP balance (G946)
- Student OS-HELP ‘As at’ date. (G738)

Please consult Appendix G – Business Fields for a detailed description of the meaning and usage of these fields.

This method only queues a job on the server to be processed at a later time. To retrieve the results of processing, refer to section 4.4 – Batch Student CHESSN Allocation Results. To poll the server for the status of this job (without returning the results), use the GetBatchCallStatus method described in section 4.5.

**Note:** When a batch request has two transactions containing the same student data, there is a chance that after the first transaction is processed; the data matching component of the system will not have enough time to update its search repository before the second transaction is processed. This situation could result in different CHESSNs being returned for each transaction, even though the student data was exactly the same. To avoid this situation, transactions containing the same student data should not be sent in the same request.

**Supports Compression**
Yes

**Parameter List**
<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chessn Request</td>
<td>heims: AllocateChessnBatchRequest</td>
<td>The batch request object. The AllocateChessnBatchRequest schema defines one RequestControlTable and can contain many AllocateChessnIn elements. Each AllocateChessnIn element contains details of a student for which a CHESSN is required. Please refer to: Appendix D.2 – Schema definition for AllocateChessnIn Appendix D.3 – Schema definition for AllocateChessnBatchRequest Appendix D.1 – Schema definition for RequestControlTable</td>
</tr>
</tbody>
</table>

**Business Rules**

Business Rules for batch student CHESSN allocation and duplicate handling are the same as those for real-time (see section 4.2)

**Return Type**

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>heims:ResponseControlTable</td>
<td>The ResponseControlTable will contain information on the status of the request. Please refer to: Appendix D.1 – Schema definition for ResponseControlTable</td>
</tr>
</tbody>
</table>
## Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the <code>ClientOrganisationCode</code> provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>FAILURE</td>
<td>A duplicate <code>RequestId</code> was provided, but it was not for an existing batch <code>AllocateStudentChessn</code> method.</td>
<td>10</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The <code>RequestDateTime</code> is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>DUPLICATE</td>
<td>A duplicate <code>RequestId</code> was provided for an existing batch <code>AllocateChessn</code> method. A subsequent call to the <code>AllocateStudentChessnResults</code> method will return results of the original request. No new processing job was queued on the server.</td>
<td>8</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. A subsequent call to the <code>AllocateStudentChessnResults</code> method will not return results of the original request.</td>
<td>9</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Request was received and is queued for processing.</td>
<td>None</td>
</tr>
</tbody>
</table>
4.4 Student CHESSN Allocation Results (Batch)

End point URL: https://app.heims.education.gov.au/WebServices.CHESSN/Batch/service.asmx


Method Signature:

AllocateChessnBatchResponse AllocateStudentChessnResults (RequestControlTable controlRequest)

Description

This method gets the results of the of a previous batch call to AllocateStudentChessn described in section 4.3. To poll the server for the status of a previous batch request (without returning the results), use the GetBatchCallStatus method described in section 4.5.

Supports Compression

Yes

Parameter List

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlRequest</td>
<td>heims:RequestControlTable</td>
<td>The RequestControlTable should contain the RequestId of the original batch AllocateStudentChessn method call. Please refer to: Appendix D.1 – Schema definition for RequestControlTable</td>
</tr>
</tbody>
</table>

Business Rules

Because a call to AllocateStudentChessnResults contains no input transaction data, no business rules are applicable for this method. For business rules relating to student CHESSN allocation in general, please see section 4.2.

Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>heims:AllocateChessnBatchResponse</td>
<td>The batch response object. The AllocateChessnBatchResponse schema defines one ResponseControlTable element and can contain multiple AllocateChessnOut elements. Each AllocateChessnOut element will contain the CHESSN allocated for a student. Please refer to: Appendix D.3 – Schema definition for AllocateChessnBatchResponse Appendix D.2 – Schema definition for AllocateChessnOut Appendix D.1 – Schema definition for ResponseControlTable</td>
</tr>
</tbody>
</table>
## Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestId given does not match an existing request.</td>
<td>7</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the ClientOrganisationCode provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>FAILURE</td>
<td>An existing RequestId was provided, but the RequestId was not for a batch AllocateStudentChessn method.</td>
<td>10</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestDateTime is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>PROCESS</td>
<td>Processing for this request has not finished.</td>
<td>1 – 6</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. Processing results (ie, the AllocateChessnIn element) will not be present.</td>
<td>9</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Processing has completed and results returned.</td>
<td></td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: All active CHESSNs.</td>
<td>10036</td>
</tr>
<tr>
<td></td>
<td>Message: 'The available SLE for this active CHESSN is subject to any enrolments and variations that may be applicable subsequent to the 'As at' date.'</td>
<td></td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: For active CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa).</td>
<td>10033</td>
</tr>
<tr>
<td></td>
<td>Message: 'The FEE-HELP balance for this active CHESSN is subject to any enrolments, variations, and indexation that may be applicable subsequent to the 'As at' date.'</td>
<td></td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: For provisional CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa) or ‘3’ (Permanent visa holder other than humanitarian visa) or ‘X’ or NULL.</td>
<td>10034</td>
</tr>
<tr>
<td></td>
<td>Message: 'The FEE-HELP balance may be subject to usage not yet reported to the department and should be checked with the student.'</td>
<td></td>
</tr>
</tbody>
</table>
4.5 Retrieving Batch Call Status

End point URL: https://app.heim.s.education.gov.au/WebServices.CHESSN/Batch/service.asmx


Method Signature
ResponseControlTable GetBatchCallStatus (RequestControlTable controlRequest)

Description
This method returns the processing status for any previous batch request.

Supports Compression
Yes

Parameter List

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>control</td>
<td>heims:RequestControlTable</td>
<td>The RequestControlTable should contain the RequestId of the original batch method for which the status is required. Please refer to: Appendix D.1 – Schema definition for RequestControlTable</td>
</tr>
</tbody>
</table>

Business Rules
Because a call to GetBatchCallStatus does not contain any input transaction data, no business rules are applicable.

Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>heims:ResponseControlTable</td>
<td>The ResponseControlTable contains information on the status of the request including an indication of how many transactions have been processed so far. Please refer to: Appendix D.1 – Schema definition for ResponseControlTable</td>
</tr>
</tbody>
</table>

Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestId given does not match an existing request.</td>
<td>7</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the ClientOrganisationCode provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>RequestStatusCode</td>
<td>Condition</td>
<td>Applicable Message Codes</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The <em>RequestId</em> given was not for a batch method.</td>
<td>13</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The <em>RequestDateTime</em> is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>PROCESS</td>
<td>Processing for the request with this <em>RequestId</em> has not finished.</td>
<td>1 – 6</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. A subsequent call to retrieve batch results will not return any processing data.</td>
<td>9</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Processing for the request with this <em>RequestId</em> has completed.</td>
<td>None</td>
</tr>
</tbody>
</table>
4.6 Real-Time Student Entitlement Details

End point URL: https://app.heims.education.gov.au/WebServices.CHESSN/service.asmx


Method Signature

GetEntitlementResponse GetStudentEntitlement (GetEntitlementRequest entitlementRequest)

Description

This method provides a real-time service to retrieve entitlement details for a particular student. Given the combination of a student CHESSN, Family Name and Date of Birth, the following information is retrieved for the student:

- Ordinary Student Learning Entitlement (SLE) limit (G898)
- Ordinary Student Learning Entitlement (SLE) usage. (G746)
- Ordinary Student Learning Entitlement (SLE) balance (G747)
- Ordinary Student Learning Entitlement (SLE) ‘As at’ date. (G738)
- Student FEE-HELP limit (G669)
- Student FEE-HELP usage (G749)
- Student FEE-HELP Loan Balance (G129, G768, G757)
- Student FEE-HELP ‘As at’ date. (G738)
- Student Ordinary limit (G985)
- Student Ordinary usage (G1035)
- Student Ordinary balance (G989)
- Student Associate limit (G986)
- Student Associate usage (G1036)
- Student Associate balance (G990)
- Student Enabling limit (G987)
- Student Enabling usage (G998)
- Student Enabling balance (G991)
- Student Indigenous limit (G988)
- Student Indigenous usage (G999)
- Student Indigenous balance (G992)
- Student Commonwealth Scholarships ‘As at’ date. (G738)
- Student OS-HELP limit (G945)
- Student OS-HELP usage (G944)
- Student OS-HELP balance (G946)
- Student OS-HELP ‘As at’ date. (G738)

Please consult Appendix G – Business Fields for a detailed description of the meaning and usage of these fields.

Supports Compression

Yes
Parameter List

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>entitlement Request</td>
<td>heims:GetEntitlementRequest</td>
<td>The real-time request object. The GetEntitlementRequest schema defines one RequestControlTable and one GetEntitlementIn element. Each GetEntitlementIn element contains student information required to return entitlement details. Please refer to: Appendix D.7 – Schema definition for GetEntitlementRequest Appendix D.5 – Schema definition for GetEntitlementIn Appendix D.1 – Schema definition for RequestControlTable</td>
</tr>
</tbody>
</table>

Business Rules

<table>
<thead>
<tr>
<th>Data Element</th>
<th>Business Rule</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chessn</td>
<td>The value of Chessn is invalid according to the check digit algorithm. (See Appendix E)</td>
<td>10223</td>
</tr>
<tr>
<td>Chessn</td>
<td>The value of Chessn must be a valid CHESSN which HEIMS has previously allocated.</td>
<td>10221</td>
</tr>
<tr>
<td>BirthDate</td>
<td>The value of the BirthDate must be greater than the current server date – 110 years AND The value of the DateofBirth must be less than the current server date.</td>
<td>10203</td>
</tr>
<tr>
<td>BirthDate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FamilyName</td>
<td>The value of BirthDate and FamilyName must be an acceptable match with that currently recorded for the student.</td>
<td>10222</td>
</tr>
</tbody>
</table>

CHESSN Duplicate Handling

With Release 1.1.0 of the HEIMS CHESSN Interface software the department is able to administer Type 1 CHESSN duplicates identified through a manual form-based procedure for TAC, HEP and VET Provider notification of CHESSN duplicates to the department.

Business rules are as follows:

<table>
<thead>
<tr>
<th>Business Rule</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entitlement information returned for CHESSNs duplicates</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Business Rule

<table>
<thead>
<tr>
<th>Business Rule</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entitlement request using a duplicate CHESSN</td>
<td>10312</td>
</tr>
</tbody>
</table>

If an entitlement information request is made where the supplied candidate CHESSN is recorded as a duplicate CHESSN in HEIMS, then information message 10312 will be returned. The message supplies the master CHESSN details.

### Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| heims:GetEntitlementResponse | The real-time response object. The GetEntitlementResponse schema defines one ResponseControlTable and one GetEntitlementOut element. The GetEntitlementOut element will contain the student entitlement details such as ordinary SLE usage and FEE-HELP Loan Balance.  
Please refer to:  
Appendix D.7 – Schema definition for GetEntitlementResponse  
Appendix D.5 – Schema definition for GetEntitlementOut  
Appendix D.1 – Schema definition for ResponseControlTable |
## Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the ClientOrganisationCode provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>FAILURE</td>
<td>A duplicate RequestId was provided, but the RequestId was not for a real-time GetStudentEntitlement method.</td>
<td>10</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestDateTime is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>DUPLICATE</td>
<td>The RequestId matches a previous real-time GetStudentEntitlement request. The same results will now be returned. No new processing was performed.</td>
<td>8</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. Processing results are not returned.</td>
<td>9</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Processing for this request is complete and will be returned.</td>
<td></td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: All active CHESSNs. Message: ‘The available SLE for this active CHESSN is subject to any enrolments and variations that may be applicable subsequent to the ‘As at’ date.’</td>
<td>10036</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: For active CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa). Message: ‘The FEE-HELP balance for this active CHESSN is subject to any enrolments, variations, and indexation that may be applicable subsequent to the ‘As at’ date.’</td>
<td>10033</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: For provisional CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa) or ‘3’ (Permanent visa holder other than humanitarian visa) or ‘X’ or NULL. Message: ‘The FEE-HELP balance may be subject to usage not yet reported to the department and should be checked with the student.’</td>
<td>10034</td>
</tr>
</tbody>
</table>
4.7 Student Entitlement Details (Batch)

End point URL: https://app.heim.education.gov.au/WebServices.CHESSN/Batch/service.asmx


Method Signature
ResponseControlTable GetStudentEntitlement (GetEntitlementBatchRequest entitlementRequest)

Description
This method submits a batch request to get entitlement details for a set of students. It queues a job on the server to be processed at a later time. To retrieve the results of processing, refer to section 5.8 – Batch Student Entitlement Results. To poll the server for the status of this job (without returning the results), use the GetBatchCallStatus method described in section 5.5.

Supports Compression
Yes

Parameter List

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| entitlementRequest | heims: GetEntitlementBatchRequest | The batch request object. The GetEntitlementBatchRequest schema defines one RequestControlTable and can contain many GetEntitlementIn elements. Each GetEntitlementIn element contains student information required to return entitlement details.  

Please refer to:
Appendix D.5 – Schema definition for GetEntitlementIn
Appendix D.6 – Schema definition for GetEntitlementBatchRequest
Appendix D.1 – Schema definition for RequestControlTable |

Business Rules
The business rules and rules for duplicate handling are the same as the real-time version of this method. See section 5.6.

Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| heims:ResponseControlTable | The ResponseControlTable will contain information on the status of the request.  

Please refer to:
Appendix D.1 – Schema definition for ResponseControlTable |
### Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the <code>ClientOrganisationCode</code> provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>FAILURE</td>
<td>A duplicate <code>RequestId</code> was provided, but it was not for an existing batch <code>GetStudentEntitlement</code> method.</td>
<td>10</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The <code>RequestDateTime</code> is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>DUPLICATE</td>
<td>A duplicate <code>RequestId</code> was provided for an existing batch <code>GetStudentEntitlement</code> method. A subsequent call to the <code>GetStudentEntitlementResults</code> method will return results of the original request. No new processing job was queued on the server.</td>
<td>8</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. A subsequent call to the <code>GetStudentEntitlementResults</code> method will not return results of the original request.</td>
<td>9</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Request was received and is queued for processing.</td>
<td>None</td>
</tr>
</tbody>
</table>
4.8 Student Entitlement Detail Results (Batch)

**End point URL:** https://app.heim.s.education.gov.au/WebServices.CHESSN/Batch/service.asmx

**WSDL Location:** https://app.heim.s.education.gov.au/WebServices.CHESSN/Batch/service.asmx?WSDL

**Method Signature**

GetEntitlementBatchResponse GetStudentEntitlementResults (RequestControlTable controlRequest)

**Description**

This method gets the results of a previous batch request for student entitlement. To poll the server for the status of a previous batch request (without returning the results), use the GetBatchCallStatus method described in section 5.5.

Given a student CHESSN, the following information is retrieved for the student:

- Ordinary Student Learning Entitlement (SLE) limit (G898)
- Ordinary Student Learning Entitlement (SLE) usage. (G746)
- Ordinary Student Learning Entitlement (SLE) balance (G747)
- Ordinary Student Learning Entitlement (SLE) ‘As at’ date. (G738)
- Student FEE-HELP limit (G669)
- Student FEE-HELP usage (G749)
- Student FEE-HELP Loan Balance (G129, G768, G757)
- Student FEE-HELP ‘As at’ date. (G738)
- Student Ordinary limit (G985)
- Student Ordinary usage (G1035)
- Student Ordinary balance (G989)
- Student Associate limit (G986)
- Student Associate usage (G1036)
- Student Associate balance (G990)
- Student Enabling limit (G987)
- Student Enabling usage (G998)
- Student Enabling balance (G991)
- Student Indigenous limit (G988)
- Student Indigenous usage (G999)
- Student Indigenous balance (G992)
- Student Commonwealth Scholarships ‘As at’ date. (G738)
- Student OS-HELP limit (G945)
- Student OS-HELP usage (G944)
- Student OS-HELP balance (G946)
- Student OS-HELP ‘As at’ date. (G738)

Please consult Appendix G – Business Fields for a detailed description of the meaning and usage of these fields.

**Supports Compression**

Yes

**Parameter List**

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlRequest</td>
<td>heims:RequestControlTable</td>
<td>The RequestControlTable should contain the RequestId of the original batch GetStudentEntitlement method call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Please refer to:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Appendix D.1 – Schema definition for RequestControlTable</td>
</tr>
</tbody>
</table>
Business Rules
Because a call to GetStudentEntitlementResults contains no input transaction data, no business rules are applicable for this method. For business rules relating to the retrieval of student entitlement details in general, please see section 5.6.

Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>heims:GetEntitlementBatchResponse</td>
<td>The batch response object. The GetEntitlementBatchResponse schema defines one ResponseControlTable element and can contain multiple GetEntitlementOut elements. Each GetEntitlementOut element will contain student entitlement information such as Ordinary SLE usage and FEE-HELP Loan balance. Please refer to: Appendix D.6 – Schema definition for GetEntitlementBatchResponse Appendix D.5 – Schema definition for GetEntitlementOut Appendix D.1 – Schema definition for ResponseControlTable</td>
</tr>
</tbody>
</table>
## Request Messages

<table>
<thead>
<tr>
<th>RequestStatusCode</th>
<th>Condition</th>
<th>Applicable Message Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILURE</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
<td>10007</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
<td>10008</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestId given does not match an existing request.</td>
<td>7</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Invalid ClientOrganisationCode provided.</td>
<td>12</td>
</tr>
<tr>
<td>FAILURE</td>
<td>Access to send or view results for the ClientOrganisationCode provided is denied.</td>
<td>11</td>
</tr>
<tr>
<td>FAILURE</td>
<td>An existing RequestId was provided, but the RequestId was not for a batch GetStudentEntitlement method.</td>
<td>10</td>
</tr>
<tr>
<td>FAILURE</td>
<td>The RequestDateTime is not valid.</td>
<td>14</td>
</tr>
<tr>
<td>PROCESS</td>
<td>Processing for this request has not finished.</td>
<td>1 – 6</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>The results for this request have been archived. Processing results (ie, the GetEntitlementIn element) will not be present.</td>
<td>9</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Processing has completed and results returned.</td>
<td></td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: All active CHESSNs. Message: 'The available SLE for this active CHESSN is subject to any enrolments and variations that may be applicable subsequent to the 'As at' date.'</td>
<td>10036</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: For active CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa). Message: 'The FEE-HELP balance for this active CHESSN is subject to any enrolments, variations, and indexation that may be applicable subsequent to the 'As at' date.'</td>
<td>10033</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Condition: For provisional CHESSNs having Citizenship Status Code of ‘1’ (Australian Citizen) or ‘2’ (New Zealand) or ‘8’ (Permanent humanitarian visa) or ‘3’ (Permanent visa holder other than humanitarian visa) or ‘X’ or NULL. Message: 'The FEE-HELP balance may be subject to usage not yet reported to the department and should be checked with the student.'</td>
<td>10034</td>
</tr>
</tbody>
</table>
4.9 Security – Changing Password


Method Signature

void ChangePassword (string logonId, string currentPassword, string newPassword)

Description

This method allows a user to change their password used to authenticate themselves on HEIMS web services. The following rules apply when changing passwords:

- The logonId given as the first parameter to the method must match the logon ID used to authenticate with HEIMS.
- The currentPassword must be valid for the logon ID.
- The newPassword must conform to the HEIMS password security rules.

Soap envelope for change password should look like:

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   soap:Action="ChangePassword">
  <soap:Body>
    <ChangePassword>
      <logonId>testuser</logonId>
      <currentPassword>Password1</currentPassword>
      <newPassword>Password2</newPassword>
    </ChangePassword>
  </soap:Body>
</soap:Envelope>
```

Refer to section 4.4 for information related to security and passwords. If any of the above rules are not met, a SOAP exception will be thrown. Within the SOAP fault, the faultstring tag will contain the error message. For example:

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   soap:Action="ChangePassword">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>New Password must be between 7 and 15 characters long.</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Supports Compression

No

Parameter List

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logonId</td>
<td>xs:string</td>
<td>This must match the logon ID used to authenticate with the system.</td>
</tr>
<tr>
<td>currentPassword</td>
<td>xs:string</td>
<td>Current Password</td>
</tr>
<tr>
<td>newPassword</td>
<td>xs:string</td>
<td>New Password</td>
</tr>
</tbody>
</table>

Business Rules

Not applicable.
Return Type
Not applicable.

Request Messages
Not applicable.

4.10 Ping
End point URLs

WSDL Locations

Method Signature
string Ping ()

Description
This method allows a client application to ping the server. It is available on all web services provided by HEIMS. It returns a string containing the date and time on the server.

Supports Compression
No

Parameter List
None

Business Rules
Not applicable.

Return Type

<table>
<thead>
<tr>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xs:string</td>
<td>String containing date and time on the server.</td>
</tr>
</tbody>
</table>

Request Messages
Not applicable.
5 Environments
There are two environments available to users of HEIMS web services

- Production (PROD).
- Next Production Release (NPR).

As parent URL for both Production and Next Production Release environments are the same (app.heits.education.gov.au), we recommend using End Point URLs mentioned in this document for each environment.

5.1 Production
HEIMS live production environment.

See Section 4 of this document for functionality available.

5.2 Next Production Release
The Next Production Release testing environment has been constructed in order to allow updated versions of the HEIMS CHESSN Interface software to be tested by the members of the sector interfacing to the department's web services. This testing environment will allow future releases of the department's functionality to be accessed by the sector prior to production implementation to ensure that changes do not affect the functionality calling the service.

Functionality Available in Next Production Release environment:

<table>
<thead>
<tr>
<th>Function</th>
<th>Reference</th>
<th>End Point URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Time Student CHESSN Allocation</td>
<td>Section 4.2</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/service.asmx</a></td>
</tr>
<tr>
<td>Batch Student CHESSN Allocation</td>
<td>Section 4.3</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx</a></td>
</tr>
<tr>
<td>Batch Student CHESSN Allocation Results</td>
<td>Section 4.4</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx</a></td>
</tr>
<tr>
<td>Retrieving Batch Call Status</td>
<td>Section 4.5</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx</a></td>
</tr>
<tr>
<td>Real Time Student Entitlement</td>
<td>Section 4.6</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/service.asmx</a></td>
</tr>
<tr>
<td>Batch Student Entitlement</td>
<td>Section 4.7</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx</a></td>
</tr>
<tr>
<td>Batch Student Entitlement Results</td>
<td>Section 4.8</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/batch/service.asmx</a></td>
</tr>
<tr>
<td>Change Password</td>
<td>Section 4.9</td>
<td><a href="https://app.heits.education.gov.au/heimtest/webservices.chessn/service.asmx">https://app.heits.education.gov.au/heimtest/webservices.chessn/service.asmx</a></td>
</tr>
<tr>
<td>Ping</td>
<td>Section 4.10</td>
<td>Available for each Web Service end point URL</td>
</tr>
</tbody>
</table>
6 Certificate

The CHESSN web service Certificate Authority (CA) has been changed from sanExternalSite.ssc.gov.au to san.education.gov.au (Baltimore CyberTrust Root). The update in certificate will encrypt all transmitted data between client machine and HEIMS web service server. The change in certificate has been implemented from July 2014.


Additional information to assist IT administrator with install the CA onto client system can be found at the following links.

- Exporting Certificate Authorities (CAs) from a Website
- How to call a Web service by using a client certificate for authentication in an ASP.NET Web application

7 Troubleshooting

If any problems are encountered with HEIMS web services please contact HEIMS Client Support at HEIMS.datacollections@education.gov.au or phone (02) 6240 7487.
## A. Appendix A – References and Other Relevant Documents

### A.1 References

This sub-section lists all external documents, web sites and other information sources that are referenced in this document or have been used in the gathering of requirements.

<table>
<thead>
<tr>
<th>Id</th>
<th>References</th>
</tr>
</thead>
</table>
| 1  | Web Services Architecture  
W3C Working Group Note 11 February 2004  
http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/ |
| 2  | Extensible Markup Language (XML) 1.0 (Third Edition)  
W3C Recommendation 04 February 2004  
http://www.w3.org/TR/2004/REC-xml-20040204/ |
| 3  | XML Schema Part 1: Structures  
W3C Recommendation 2 May 2001  
http://www.w3.org/TR/2001/REC-xmlschema-1/ |
| 4  | XML Schema Part 2: DataTypes  
W3C Recommendation 02 May 2001  
http://www.w3.org/TR/2001/REC-xmlschema-2/ |
| 5  | SOAP Version 1.2 Part 1: Messaging Framework  
W3C Recommendation 24 June 2003  
http://www.w3.org/TR/2003/REC-soap12-part1-20030624/ |
| 6  | AS 4590-1999 Interchange of Client Information  
Standards Australia |
| 7  | ISO/IEC 11179 Information Technology – Specification and Standardization of Data Elements |
| 8  | ISO/IEC 7064 Information Technology – Security techniques – Check character systems |
## A.2 Other Relevant Documents

This section lists all documents that relate to this interface.

<table>
<thead>
<tr>
<th>Other Relevant Documents</th>
<th>Reason for Association</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequently Asked Questions</td>
<td>Provides information on common questions about the HEIMS Technical Architecture and HEIMS Web Services.</td>
<td>HEIMS Developer Network</td>
</tr>
<tr>
<td>CHESSN Interface with HEP Business Requirements</td>
<td>HEIMS Business requirement for interfacing with HEPs.</td>
<td>HEIMS Developer Network</td>
</tr>
<tr>
<td>CHESSN Interface with TACS Business Requirements</td>
<td>HEIMS Business requirements for interfacing with TACs.</td>
<td>HEIMS Developer Network</td>
</tr>
<tr>
<td>HEIMS External Development and Testing Strategy</td>
<td>Outlines the HEIMS Proof of Concept, pilot testing strategy and sign off requirements.</td>
<td>HEIMS Developer Network</td>
</tr>
<tr>
<td>HEIMS Findings and Recommendations from the IT Questionnaire</td>
<td>Results of the IT Questionnaire circulated to HEPs and TACs.</td>
<td>HEIMS Developer Network</td>
</tr>
<tr>
<td>HEIMS Production Credentials Procedures</td>
<td>Arrangements for gaining access to the HEIMS production environment.</td>
<td>HEIMS Developer Network</td>
</tr>
</tbody>
</table>
B. Appendix B – Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL</td>
<td>Web Services Definition Language</td>
</tr>
<tr>
<td></td>
<td>The standard format for describing a Web service. Written in XML, it defines how to access a web service and what operations it will perform.</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol</td>
</tr>
<tr>
<td></td>
<td>A suite of protocols that computers use to exchange information over the Internet.</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Objects Access Protocol</td>
</tr>
<tr>
<td></td>
<td>An XML based lightweight protocol for exchange of information in a decentralised, distributed environment.</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td></td>
<td>An international consortium of companies involved with developing standards for the Internet and the Web.</td>
</tr>
<tr>
<td>Basic Authentication</td>
<td>An authentication protocol supported by most browsers where username and password is Base-64 encoded and sent. Sometimes called plaintext authentication because anybody can decode it. Note that encoding is not the same as encryption.</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td></td>
<td>The protocol for moving hypertext files across the Internet.</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td></td>
<td>A commonly used protocol for managing the security of a message transmission on the Internet.</td>
</tr>
<tr>
<td>HTTPS</td>
<td>HyperText Transfer Protocol (Secure)</td>
</tr>
<tr>
<td></td>
<td>HTTP exchanged over an SSL encrypted session.</td>
</tr>
<tr>
<td>Type 1 CHESSN duplicate</td>
<td>A single student recorded in HEIMS with multiple CHESSNs.</td>
</tr>
<tr>
<td>Master CHESSN</td>
<td>When multiple CHESSNs are confirmed as Type 1 CHESSN duplicates one of the CHESSNs is flagged in the HEIMS data stores as a duplicate CHESSN and the other will remain as the master CHESSN. Note that the set of master CHESSNs includes all CHESSNs that are not identified as a duplicate CHESSN. Only master CHESSNs are valid for use by students, client organisations, and the department.</td>
</tr>
<tr>
<td>SSC</td>
<td>Shared Services Centre</td>
</tr>
<tr>
<td>HEPCAT</td>
<td>Higher Education Client Assistance Tool</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td></td>
<td>Is a cryptographic protocol that provides communication security over a computer network. It aims to provide privacy and data integrity between two or more communicating computer applications.</td>
</tr>
</tbody>
</table>
C. Appendix C – List of Business Messages

C.1 Request Error Messages
These are request level messages relating to all Web service calls:

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[number] total transactions in request</td>
</tr>
<tr>
<td>2</td>
<td>[number] transactions waiting to start</td>
</tr>
<tr>
<td>3</td>
<td>[number] transactions currently executing</td>
</tr>
<tr>
<td>4</td>
<td>[number] transactions completed successfully</td>
</tr>
<tr>
<td>5</td>
<td>[number] transactions completed with warnings</td>
</tr>
<tr>
<td>6</td>
<td>[number] transactions completed in error</td>
</tr>
<tr>
<td>7</td>
<td>The RequestId given does not match an existing request</td>
</tr>
<tr>
<td>8</td>
<td>A duplicate RequestId was provided for an existing method. No new processing will be performed</td>
</tr>
<tr>
<td>9</td>
<td>The results for this request have been archived</td>
</tr>
<tr>
<td>10</td>
<td>An existing RequestId was provided, but the RequestId was not for the same method</td>
</tr>
<tr>
<td>11</td>
<td>Access to send or view results for the ClientOrganisationCode provided is denied</td>
</tr>
<tr>
<td>12</td>
<td>Invalid ClientOrganisationCode provided</td>
</tr>
<tr>
<td>13</td>
<td>The RequestId given was not for a batch method.</td>
</tr>
<tr>
<td>14</td>
<td>The RequestDateTime is not valid.</td>
</tr>
</tbody>
</table>

C.2 Security Related
These are security messages related to Web service access.

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10007</td>
<td>There appears to be a problem with your access. Please contact the HEIMS IT Liaison officer.</td>
</tr>
<tr>
<td>10008</td>
<td>Access to this particular function is denied. Please contact the HEIMS IT Liaison officer.</td>
</tr>
</tbody>
</table>
C.3 **Business Messages**
These are the business messages relating to all Web service calls.

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10036</td>
<td>The available SLE for this active CHESSN is subject to any enrolments and variations that may be applicable subsequent to the ‘As at’ date.</td>
</tr>
<tr>
<td>10033</td>
<td>The FEE-HELP balance for this active CHESSN is subject to any enrolments, variations, and indexation that may be applicable subsequent to the ‘As at’ date.</td>
</tr>
<tr>
<td>10034</td>
<td>The FEE-HELP balance may be subject to usage not yet reported to the department and should be checked with the student.</td>
</tr>
<tr>
<td>10039</td>
<td>The student details submitted closely match an existing CHESSN record. The student details associated with the existing record have been returned. Please check that these are correct and contact the Help Desk if you are experiencing difficulties.</td>
</tr>
<tr>
<td>10040</td>
<td>The available OS-HELP data for this active CHESSN is as per the latest submission from the HEP. As OS-HELP data is reported twice yearly, there may be a discrepancy between the remaining available OS-HELP loan entitlement and that displayed here.</td>
</tr>
<tr>
<td>10201</td>
<td>All data fields identified as mandatory must not contain a null value.</td>
</tr>
<tr>
<td>10203</td>
<td>Invalid BirthDate.</td>
</tr>
<tr>
<td>10205</td>
<td>Invalid CitizenshipStatusCode.</td>
</tr>
<tr>
<td>10207</td>
<td>If <code>AttendedYear12Code</code> is equal to ‘AttendedYear12’, then <code>Year12State</code> must not be empty.</td>
</tr>
<tr>
<td>10208</td>
<td>Year12State must be a valid Australian state.</td>
</tr>
<tr>
<td>10209</td>
<td>If <code>AttendedYear12Code</code> is equal to ‘AttendedYear12’, then <code>Year12Year</code> must not be empty.</td>
</tr>
<tr>
<td>10210</td>
<td>If <code>AttendedYear12Code</code> is equal to ‘AttendedYear12’, then <code>Year12Number</code> then <code>Year12SchoolName</code> must not be blank.</td>
</tr>
<tr>
<td>10214</td>
<td>If <code>AttendedPreviousHEPCode</code> is equal to ‘AttendedPreviousHep’, then <code>HepName</code> and <code>HepStudentNumber</code> both must not be blank (ie at least one of these 2 data elements must contain a value).</td>
</tr>
<tr>
<td>10215</td>
<td>If <code>AttendedPreviousHEPCode</code> is equal to ‘AttendedPreviousHep’ then <code>HepYear</code> must not be blank.</td>
</tr>
<tr>
<td>10216</td>
<td><code>PostalCountryCode</code> must be a valid code.</td>
</tr>
<tr>
<td>10218</td>
<td>If <code>PostalCountryCode</code> is equal to ‘1100’ (code for Australia), then <code>PostalPostCode</code> must not be empty.</td>
</tr>
<tr>
<td>10221</td>
<td>The supplied CHESSN does not match an existing CHESSN in HEIMS.</td>
</tr>
<tr>
<td>10222</td>
<td><code>FamilyName</code> and <code>BirthDate</code> are not an acceptable match with that currently recorded for the student.</td>
</tr>
<tr>
<td>10223</td>
<td>Invalid CHESSN</td>
</tr>
<tr>
<td>10224</td>
<td><code>Year12Details</code> must be present.</td>
</tr>
<tr>
<td>10225</td>
<td>Invalid year for <code>Year12Year</code>.</td>
</tr>
<tr>
<td>Message Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>10226</td>
<td>If the value of <code>AttendedPreviousHepCode</code> is equal to ‘AttendedPreviousHep’, then <code>PreviousHepDetails</code> must be present.</td>
</tr>
<tr>
<td>10228</td>
<td>If the value of <code>AttendedPreviousHepCode</code> is equal to ‘AttendedPreviousHep’ then <code>HepCode</code> must be valid according to Appendix A on HEIMSHELP.</td>
</tr>
<tr>
<td>10229</td>
<td>Invalid year for <code>HepYear</code>.</td>
</tr>
<tr>
<td>10230</td>
<td><code>PostCode</code> must be in the range of ‘0001’ to ‘9999’.</td>
</tr>
<tr>
<td>10259</td>
<td><code>SexCode</code> is invalid.</td>
</tr>
<tr>
<td>10295</td>
<td>The confirmation data entered for <code>Family Name and Date of Birth</code> is not an acceptable match with the <code>Family Name and Date of Birth</code> currently recorded against the CHESSN.</td>
</tr>
<tr>
<td>10260</td>
<td>Only a valid Higher Education Provider can assign a CHESSN for a continuing student.</td>
</tr>
<tr>
<td>10262</td>
<td>CHESSN nnnnnnnnnn is not valid. This CHESSN does not exist in the HEIMS database.</td>
</tr>
</tbody>
</table>
| 10312       | The CHESSN that you have supplied has been identified as a duplicate CHESSN for this student. The entitlement information for its associated Master CHESSN has been returned. 

Correct CHESSN: `<related Master CHESSN>`

Family Name: `<Family Name for related Master CHESSN>`

Date of Birth: `<Date of birth for related Master CHESSN>`
D. Appendix D – Web Service Schemas

For XML elements that are concerned with the exchange of information between clients, the format and names of data elements are based on the Australian Standard 4590-1999 and ISO Standard 11179. Because of this, field names and lengths in the interface are not necessarily the same as those in HEPCAT. However, the change in field format need not impact HEPs, TACs and VET Providers as the length of the HEIMS interface fields are longer than HEPCAT fields.

HEIMS Web services will use HEPCAT code values where applicable. The equivalent HEPCAT element number is noted in the XML schema annotation. Whilst code values are available in HEPCAT, it is intended that the department will also provide XML files describing every schema element and its associated code values on the HEIMS Developers network. Codes with only a small number of static values are enumerated in the XML schema.

The latest version of the schema is always available from the HEIMS Developers Network.

D.1 Business.xsd

This is a file containing all base types and elements.

```xml
<?xml version="1.0" encoding="utf-8" ?>
   elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.1">
   <!-- simple types start here -->
   <xs:simpleType name="GivenName">
     <xs:annotation>
       <xs:documentation>The first name of the person, also known as Christian name. The format is based on Clause 3.3 in AS4590-1999.</xs:documentation>
     </xs:annotation>
     <xs:restriction base="xs:string">
       <xs:minLength value="1" />  
       <xs:maxLength value="40" />
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="OtherGivenName">
     <xs:annotation>
       <xs:documentation>Represents any other names a person has such as middle names. The format is based on Clause 3.3 in AS4590-1999.</xs:documentation>
     </xs:annotation>
     <xs:restriction base="xs:string">
       <xs:minLength value="1" />  
       <xs:maxLength value="40" />
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="PersonName">
     <xs:annotation>
       <xs:documentation>
       </xs:documentation>
     </xs:annotation>
   </xs:simpleType>
</xs:schema>
```
<xs:documentation>Represents a persons full name.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
  <xs:minLength value="1" />
  <xs:maxLength value="100" />
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="FamilyName">
  <xs:annotation>
    <xs:documentation>Represents a persons family name or surname. If a person has only one name, it should go in this field. The format is based on Clause 3.4 in AS4590-1999.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="40" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="BirthDate">
  <xs:annotation>
    <xs:documentation>Represents a persons date of birth.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:date">
    <xs:minInclusive value="1879-01-01" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="AddressLine">
  <xs:annotation>
    <xs:documentation>Defines an address line used for capturing addresses.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="50" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CountryName">
  <xs:annotation>
    <xs:documentation>The name of a country.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CountryCode">
  <xs:annotation>
    <xs:documentation>4-digit country code. See appendix G in DESTPAC.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PostCode">
  <xs:annotation>
    <xs:documentation>Address postal code based on clause 8.14 in AS4590-1999.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:maxLength value="12"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StateCode">
  <xs:annotation>
    <xs:documentation>Australian address state or territory code based on DESTPAC element 467.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="NSW"/>
    <xs:enumeration value="QLD"/>
    <xs:enumeration value="VIC"/>
    <xs:enumeration value="TAS"/>
    <xs:enumeration value="ACT"/>
    <xs:enumeration value="NT"/>
    <xs:enumeration value="SA"/>
    <xs:enumeration value="WA"/>
    <xs:enumeration value="AAT"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SexCode">
  <xs:annotation>
    <xs:documentation>A code indicating the biological distinction between male and female. Based on clause 5.3 in AS4590-1999.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:enumeration value="M" />
<xs:enumeration value="F" />
<xs:enumeration value="X" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="CitizenshipStatusCode">
<xs:annotation>
<xs:documentation>Indicates the citizenship or residency of a person. These values are based on DESTPAC element 358.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:pattern value="[1234589]" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="RevisionReasonCode">
<xs:annotation>
<xs:documentation>A 1-digit code representing reason for revision.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:pattern value="[12345]" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="HepCode">
<xs:annotation>
<xs:documentation>A 4-digit code representing a HEP (Higher Education Provider). The values are based on DESTPAC element 306.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:pattern value="%d\d\d\d" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="SchoolName">
<xs:annotation>
<xs:documentation>The name of a school.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:maxLength value="200" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="HepName">
<xs:annotation>
<xs:documentation>The name of a HEP (Higher Education Provider).</xs:documentation>
</xs:annotation>
</xs:simpleType>
Clause 8.11 AS-4590. The full name of the general locality containing the specific address. This will normally be the name of a town or suburb.
<xs:simpleType name="StudentYear12Year">
  <xs:annotation>
    <xs:documentation>The year in which a person attended year 12.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:gYear" />
</xs:simpleType>

<xs:simpleType name="StudentAttendedPreviousHepCode">
  <xs:annotation>
    <xs:documentation>An indicator as to whether a person attended a previous HEP (Higher Education Provider) or not.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="AttendedPreviousHep" />  
    <xs:enumeration value="DidNotAttendPreviousHep" />  
    <xs:enumeration value="Unknown" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StudentHepYear">
  <xs:annotation>
    <xs:documentation>The year in which a person attended a HEP (Higher Education Provider).</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:gYear" />
</xs:simpleType>

<xs:simpleType name="StudentHepNumber">
  <xs:annotation>
    <xs:documentation>The student number allocated to a person when they attended a HEP (Higher Education Provider).</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="10" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="LocationTypeCode">
  <xs:annotation>
    <xs:documentation>A code which identifies whether or not the student location code is in Australia or other countries.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[XA90]" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StudentStatusCode">
  <xs:annotation>
    <xs:documentation>A code which indicates the student status for a unit of study.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SummerSchoolCode">
  <xs:annotation>
    <xs:documentation>A code indicating if a unit of study is a full-fee summer school unit of study.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="SubmissionNumber">
  <xs:annotation>
    <xs:documentation>Submission Number</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="RevisionNumber">
  <xs:annotation>
    <xs:documentation>Revision Number</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="ReportingPeriodYear">
  <xs:attribute name="ReportingYear" type="ReportingYear" use="required" />
  <xs:attribute name="ReportingPeriod" type="ReportingPeriod" use="required" />
</xs:complexType>

<xs:simpleType name="ReportingPeriod">
  <xs:annotation>
    <xs:documentation>Reporting period</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType>
  <xs:complexContent>
    <xs:restriction basis="xs:string">
      <xs:minLength value="1" />
      <xs:maxLength value="72" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="CourseOfStudyLoad">
  <xs:annotation>
    <xs:documentation>Course of Study Load.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CourseCode">
  <xs:annotation>
    <xs:documentation>Hep Course code value.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="10" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CourseName">
  <xs:annotation>
    <xs:documentation>Hep Course description.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="72" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CourseSpecialTypeCode">
  <xs:annotation>
    <xs:documentation>Special course type code value.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CampusLocation">
  <xs:annotation>
    <xs:documentation>Hep Campus Location description.</xs:documentation>
  </xs:annotation>
</xs:simpleType>
<xs:documentation source="E200">Rank/QLD Cut Off Score.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
  <xs:pattern value="\d{2}" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="TacOffer">
  <xs:annotation>
    <xs:documentation>Field of Education code value.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="5" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CampusOPTypeCode">
  <xs:annotation>
    <xs:documentation>Code which identifies the type of arrangement through which an offshore course is being delivered.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{2}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="OffShoreModeCode">
  <xs:annotation>
    <xs:documentation>Code identifying the main mode of delivery for a course that is offered offshore.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{2}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="OffShoreDeliveryCode">
  <xs:annotation>
    <xs:documentation>Code identifying whether a course is delivered in Australia, partially offshore or fully offshore.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{2}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="AreasOfStudy">
Areas of study.

Course search keywords.

Defines available status codes CHESSN.

A structure for generic addresses based on the AS4590 standard. The standard splits up addresses into 13 clauses, but suggests they can be combined into several address lines. Please use the LocalityName, StateCode, PostCode and CountryName elements if possible and avoid combining these fields into the AddressLine2 element.
<xs:element name="CountryName" type="CountryName" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="PreviousName">
<xs:annotation>
     <xs:documentation>A container for previous names.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="1" maxOccurs="1">
     <xs:element name="GivenName" type="GivenName" minOccurs="0" maxOccurs="1" />
     <xs:element name="OtherGivenName" type="OtherGivenName" minOccurs="0" maxOccurs="unbounded" />
     <xs:element name="FamilyName" type="FamilyName" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="HighestEducationalAttainment">
<xs:annotation>
     <xs:documentation>A structure for the student's highest educational attainment and year of attainment prior to the first enrolment in the course.</xs:documentation>
</xs:annotation>
<xs:attribute name="AttainmentCode" type="StudentHighestEducationalAttainmentCode" use="optional" />
<xs:attribute name="AttainmentYear" type="StudentHighestEducationalAttainmentYear" use="optional" />
</xs:complexType>
<xs:simpleType name="HighestEducationalAttainmentOfParentOrGuardian">
<xs:annotation>
     <xs:documentation>A structure for the student's highest educational attainment and year of attainment prior to the first enrolment in the course.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
     <xs:length value="2" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="LocationDetails">
<xs:annotation>
     <xs:documentation>A structure for location details.</xs:documentation>
</xs:annotation>
<xs:attribute name="LocationTypeCode" type="LocationTypeCode" use="required" />
<xs:attribute name="LocationCode" type="LocationCode" use="required" />
</xs:complexType>
<!-- Types used in Student Load -->
<xs:simpleType name="ATORunTypeCode">
<xs:annotation>
     <xs:documentation>A code used to distinguish whether data in a file relates to test data or production data.</xs:documentation>
</xs:annotation>
<xs:simpleType name="LocationTypeCode">
<xs:annotation>
     <xs:documentation>A code used to distinguish whether data in a file relates to test data or production data.</xs:documentation>
</xs:annotation>
<xs:simpleType name="LocationCode">
<xs:annotation>
     <xs:documentation>A code used to distinguish whether data in a file relates to test data or production data.</xs:documentation>
</xs:annotation>
<xs:complexType name="CountryName">
<xs:annotation>
     <xs:documentation>A container for country names.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0" maxOccurs="1">
     <xs:element name="CountryName" type="CountryName" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
<!-- Types used in Student Load -->
<xs:simpleType name="ATORunTypeCode">
<xs:annotation>
     <xs:documentation>A code used to distinguish whether data in a file relates to test data or production data.</xs:documentation>
</xs:annotation>
<xs:simpleType name="LocationTypeCode">
<xs:annotation>
     <xs:documentation>A code used to distinguish whether data in a file relates to test data or production data.</xs:documentation>
</xs:annotation>
<xs:simpleType name="LocationCode">
<xs:annotation>
     <xs:documentation>A code used to distinguish whether data in a file relates to test data or production data.</xs:documentation>
</xs:annotation>
<xs:complexType name="CountryName">
<xs:annotation>
     <xs:documentation>A container for country names.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0" maxOccurs="1">
     <xs:element name="CountryName" type="CountryName" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
A code assigned by the Higher Education Provider which uniquely identifies the academic organisational unit providing a unit of study or part of a unit of study.

The EFTSL value representing the student load for a unit of study.

Course of Study Type Code value.

ASCED code value.

A code which identifies the mode of attendance by which the student undertakes a unit of study.
<xs:simpleType name="StudentMaximumContributionCode">
  <xs:annotation>
    <xs:documentation>A code that indicates which maximum student contribution was used in calculating the student contribution amount.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="StudentLiabilityStatusCode">
  <xs:annotation>
    <xs:documentation>A code which identifies, for an enrolment in a course, in a semester, the exempt status; the liability status for the Higher Education Contribution Scheme (HECS); the status for the Open Learning Deferred Payments Scheme (OLDPS); the status for the Postgraduate Education Loans Scheme (PELS); the status for Bridging for Overseas Trained Professionals Loan Scheme (BOTPLS).</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{3}" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="WEICode">
  <xs:annotation>
    <xs:documentation>An indicator of whether a unit of study consists wholly of work experience in industry.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[012]" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="EligibilityForHelpCode">
  <xs:annotation>
    <xs:documentation>A code which identifies details about citizenship, Permanent Resident Status and Permanent Visa status.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[0123]" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SummerWinterSchoolCode">
  <xs:annotation>
    <xs:documentation>A code indicating if a unit of study is a full-fee summer or winter school unit of study.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[123]" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RPLIndicatorCode">

  <xs:annotation>
    <xs:documentation>A code indicating whether a unit of study is a Recognition of Prior Learning unit of study</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StudentAdmissionBasisCode">

  <xs:annotation>
    <xs:documentation>A code which identifies the main criterion used by the Higher Education Provider in granting the student admission to the current course</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PriorExemptionStatusCode">

  <xs:annotation>
    <xs:documentation>A code which identifies the extent to which exemption or status from the course’s total requirements over all its stages were granted by the reference date of the first year of enrolment in the course</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ExemptionTypeCode">

  <xs:annotation>
    <xs:documentation>A code which identifies the exception type or Provider Code. Destpac 368</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TertiaryEntranceScore">

  <xs:annotation>
    <xs:documentation>A code indicating the tertiary entrance score obtained by a student commencing an undergraduate award course</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StudentSeparationStatusCode">
  <xs:annotation>
    <xs:documentation>The Separation status of a student enrolled in a Higher Degree Research course.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StudentRTSEFTSL">
  <xs:annotation>
    <xs:documentation>The total EFTSL value that has been consumed at previous Higher Education Provider by a student who was enrolled in those Higher Education Provider under the Research Training Scheme (RTS) and who is enrolled at the current Higher Education Provider under the same scheme.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ScholarshipCode">
  <xs:annotation>
    <xs:documentation>A code which identifies the scholarship type code.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="StudentHighestEducationalAttainmentCode">
  <xs:annotation>
    <xs:documentation>A code which indicates the student's highest educational attainment</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[01][0-9]" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="HighestEducationalAttainmentCodeOfParentOrGuardian">
  <xs:annotation>
    <xs:documentation>A code which indicates the student's highest educational attainment</xs:documentation>
  </xs:annotation>
</xs:simpleType>
<xs:annotation><xs:restriction base="xs:string">
  <!--xs:pattern value="[0-9]*" -->
  <xs:length value="2" />
</xs:restriction></xs:simpleType>
<xs:simpleType name="StudentHighestEducationalAttainmentYear">
  <xs:annotation>
    <xs:documentation>A year of attainment prior to the first enrolment in the course.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="AttendanceTypeCode">
  <xs:annotation>
    <xs:documentation>A code which identifies whether a student is classified as being a full-time or part-time.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[0129]" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="LanguageCode">
  <xs:annotation>
    <xs:documentation>A code indicating use of a language other than English.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ATSICode">
  <xs:annotation>
    <xs:documentation>A code which identifies whether or not the student identifies herself or himself as being of Aboriginal and/or Torres Strait Islander descent.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[23459]" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PersonTitle">
  <xs:annotation>
</xs:annotation>
The title used with a person's name.

A 4-digit code representing a Cohort Year or Code.

The tax file number for a student.

The tax file number for a student.

Arrival Australia Code or Year

Year Left School Code
<xs:annotation>
  <xs:documentation>Year Left School Code or Year</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
  <xs:length value="4"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="LabourForceStatusCode">
  <xs:annotation>
    <xs:documentation>Labour Force Status Code</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:length value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StudyReasonCode">
  <xs:annotation>
    <xs:documentation>Study Reason Code</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:length value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="USI">
  <xs:annotation>
    <xs:documentation>Unique Student Identifier</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:length value="10"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="AuthorisationDetail">
  <xs:annotation>
    <xs:documentation>A structure for authorisation details.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AuthorisationReason" type="AuthorisationReason" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="AuthoriseLogonName" type="xs:string" use="required"/>
</xs:complexType>
<xs:attribute name="AuthoriseDateTime" type="xs:dateTime" use="required" />
</xs:complexType>
<xs:simpleType name="AuthorisationReason">
    <xs:annotation>
        <xs:documentation>The reason for authorising a validation error submission.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:minLength value="1" />
        <xs:maxLength value="1000" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CreditOfferedValue">
    <xs:annotation>
        <xs:documentation></xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:length value="4" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CreditOfferedProviderCode">
    <xs:annotation>
        <xs:documentation></xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:length value="4" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CreditUsedValue">
    <xs:annotation>
        <xs:documentation></xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:length value="4" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PriorStudyCreditBasisCode">
    <xs:annotation>
        <xs:documentation></xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
    </xs:restriction>
</xs:simpleType>
A structure containing generic entitlement values

Specifies entitlements for all scholarship categories
<xs:annotation>
  <xs:sequence>
    <xs:element name="IsCommencing" type="xs:boolean" minOccurs="0" maxOccurs="1" /></xs:sequence>
  <xs:element name="OrdinaryEntitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
  <xs:element name="AssociateEntitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
  <xs:element name="EnablingEntitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
  <xs:element name="IndigenousEntitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
  <xs:element name="AsAtDate" type="xs:date" minOccurs="0" maxOccurs="1" />
</xs:complexType>
</xs:annotation>

<xs:complexType name="OrdinarySle">
  <xs:annotation>
    <xs:documentation>Specifies Entitlement for OrdinarySle</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Entitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
    <xs:element name="AsAtDate" type="xs:date" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FeeHelpLoan">
  <xs:annotation>
    <xs:documentation>Specifies Entitlement for FeeHelpLoan</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Entitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
    <xs:element name="AsAtDate" type="xs:date" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OsHelp">
  <xs:annotation>
    <xs:documentation>Specifies Entitlement for OsHelp</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Entitlement" type="EntitlementValues" minOccurs="0" maxOccurs="1" />
    <xs:element name="AsAtDate" type="xs:date" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Entitlements">
  <xs:annotation>
    <xs:documentation>A structure containing generic entitlement values</xs:documentation>
  </xs:annotation>
</xs:complexType>
<xs:sequence>
  <xs:element name="OrdinarySle" type="OrdinarySle" minOccurs="0" maxOccurs="1" />
  <xs:element name="FeeHelpLoan" type="FeeHelpLoan" minOccurs="0" maxOccurs="1" />
  <xs:element name="OsHelp" type="OsHelp" minOccurs="0" maxOccurs="1" />
  <xs:element name="CommonwealthScholarships" type="CommonwealthScholarships" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>

<xs:complexType name="EntitlementsExtension">
  <xs:annotation>
    <xs:documentation>A structure containing other entitlements values</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="CalculationAsAtDate" type="xs:date" minOccurs="0" maxOccurs="1" />
    <xs:element name="FeeHelpUsageAmountIncLoanFee" type="xs:decimal" minOccurs="1" maxOccurs="1" />
    <xs:element name="FeeHelpOverLimitAmount" type="xs:decimal" minOccurs="1" maxOccurs="1" />
    <xs:element name="FeeHelpMaximumLimitAmount" type="xs:decimal" minOccurs="0" maxOccurs="1" />
    <xs:element name="OSHelpLoanAmount" type="xs:decimal" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ScholarshipDetails">
  <xs:annotation>
    <xs:documentation>A structure containing scholarship details used by Heims Load tool</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ReportingYear" type="ReportingYear" minOccurs="0" maxOccurs="1" />
    <xs:element name="ReportingPeriod" type="ReportingPeriodCode" minOccurs="0" maxOccurs="1" />
    <xs:element name="ClientOrganisationCode" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ClientOrganisationDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ScholarshipTypeCode" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ScholarshipTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ActionReasonTypeCode" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ActionReasonTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ScholarshipStatusTypeCode" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ScholarshipStatusTypeDescription" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="ScholarshipPeriodActionReasonTypeDate" type="xs:dateTime" minOccurs="0" maxOccurs="1" />
    <xs:element name="ScholarshipTerminationReasonTypeCode" type="xs:string" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="ThreeDigits">
  <xs:annotation>
    <xs:documentation>A 3-digit string.</xs:documentation>
  </xs:annotation>
</xs:simpleType>
<xs:restriction base="xs:string">
  <xs:pattern value="\d\d\d" />
  <xs:minLength value="3"/>
  <xs:maxLength value="3"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
D.2 AllocateChessn.xsd
This file contains definitions for transaction data elements to do with the AllocateChessn web method.

```xml
<?xml version="1.0" encoding="utf-8"?>
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" version="1.1">
<xs:include schemaLocation="../../DataElements/Business.xsd" />
<xs:include schemaLocation="../../DataElements/Entities/Student.xsd" />
<xs:element name="AllocateChessnIn" id="AllocateChessnIn">
  <xs:annotation>
    <xs:documentation>Input transaction data required in the AllocateChessn method.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="RecordId" type="RecordId" minOccurs="1" maxOccurs="1" />
      <xs:element name="GivenName" type="GivenName" minOccurs="0" maxOccurs="1" />
      <xs:element name="OtherGivenName" type="OtherGivenName" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="FamilyName" type="FamilyName" minOccurs="1" maxOccurs="1" />
      <xs:element name="PreviousName" type="PreviousName" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="BirthDate" type="BirthDate" minOccurs="1" maxOccurs="1" />
      <xs:element name="SexCode" type="SexCode" minOccurs="1" maxOccurs="1" />
      <xs:element name="PostalAddress" type="Address" minOccurs="0" maxOccurs="1" />
      <xs:element name="CitizenshipStatusCode" type="CitizenshipStatusCode" minOccurs="0" maxOccurs="1" />
      <xs:element name="AttendedYear12Code" type="StudentAttendedYear12Code" minOccurs="0" maxOccurs="1" />
      <xs:element name="Year12Details" type="StudentYear12Details" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="AttendedPreviousHepCode" type="StudentAttendedPreviousHepCode" minOccurs="0" maxOccurs="1" />
      <xs:element name="PreviousHepDetails" type="StudentHepDetails" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AllocateChessnOut">
  <xs:annotation>
    <xs:documentation>Output transaction data returned in the AllocateChessn method.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
  </xs:sequence>
</xs:element>
</xs:element>
</xs:complexType>
</xs:element>
</xs:annotation>
</xs:complexType>
```
<xs:sequence>
  <xs:element name="RecordId" type="RecordId" minOccurs="1" maxOccurs="1" />
  <xs:element name="Chessn" type="StudentChessn" minOccurs="0" maxOccurs="1" />
  <xs:element name="TransactionStatus" type="TransactionStatus" minOccurs="1" maxOccurs="1" />
  <xs:element name="Entitlements" type="Entitlements" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
D.3 AllocateChessnBatch.xsd

This file describes the Request and Response types for the batch version of the AllocateChessn method.

```xml
<?xml version="1.0" encoding="utf-8" ?>
    xmlns="http://dest.gov.au/Heims/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" version="1.3">
    <xs:include schemaLocation="../structures/AllocateChessn.xsd" />
    <xs:include schemaLocation="../structures/Technical.xsd" />
    <xs:element name="AllocateChessnBatchRequest">
        <xs:annotation>
            <xs:documentation>A batch request to the AllocateChessn method.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="RequestControlTable" minOccurs="1" maxOccurs="1" />
                <xs:element ref="AllocateChessnIn" minOccurs="1" maxOccurs="10000" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="AllocateChessnBatchResponse">
        <xs:annotation>
            <xs:documentation>A batch response from the AllocateChessn method.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ResponseControlTable" minOccurs="1" maxOccurs="1" />
                <xs:element ref="AllocateChessnOut" minOccurs="0" maxOccurs="10000" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```
D.4 AllocateChessnRealTime.xsd

This file describes the Request and Response types for the real time version of the AllocateChessn method.

```xml
<?xml version="1.0" encoding="utf-8" ?>
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" version="1.3">
  <xs:include schemaLocation="../structures/AllocateChessn.xsd" />
  <xs:include schemaLocation="../structures/Technical.xsd" />
  <xs:element name="AllocateChessnRequest">
    <xs:annotation>
      <xs:documentation>A real-time request to the AllocateChessn Method.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="RequestControlTable" minOccurs="1" maxOccurs="1" />
        <xs:element ref="AllocateChessnIn" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="AllocateChessnResponse">
    <xs:annotation>
      <xs:documentation>A real-time request to the AllocateChessn Method.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ResponseControlTable" minOccurs="1" maxOccurs="1" />
        <xs:element ref="AllocateChessnOut" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
D.5  GetEntitlement.xsd

This file contains definitions for transaction data elements to do with the GetStudentEntitlement web method.

```xml
<?xml version="1.0" encoding="utf-8" ?>
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" version="1.2">
  <xs:include schemaLocation="../DataElements/Business.xsd" />
  <xs:include schemaLocation="../DataElements/Entities/Student.xsd" />

  <xs:element name="GetEntitlementIn">
    <xs:annotation>
      <xs:documentation>Input transaction data returned in the GetEntitlement method.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RecordId" type="RecordId" minOccurs="1" maxOccurs="1" />
        <xs:element name="Chessn" type="StudentChessn" minOccurs="1" maxOccurs="1" />
        <xs:element name="FamilyName" type="FamilyName" minOccurs="1" maxOccurs="1" />
        <xs:element name="BirthDate" type="BirthDate" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="GetEntitlementOut">
    <xs:annotation>
      <xs:documentation>Output transaction data returned in the GetEntitlement method.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RecordId" type="RecordId" minOccurs="1" maxOccurs="1" />
        <xs:element name="TransactionStatus" type="TransactionStatus" minOccurs="1" maxOccurs="1" />
        <xs:element name="Entitlements" type="Entitlements" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
D.6 GetEntitlementBatch.xsd

This file describes the Request and Response types for the batch version of the GetEntitlement method.

```xml
<?xml version="1.0" encoding="utf-8" ?>
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.3">
<xs:include schemaLocation="../structures/GetEntitlement.xsd" />
<xs:include schemaLocation="../structures/Technical.xsd" />
<xs:element name="GetEntitlementBatchRequest">
  <xs:annotation>
    <xs:documentation>A batch request to the GetStudentEntitlement method.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="RequestControlTable" minOccurs="1" maxOccurs="1" />
      <xs:element ref="GetEntitlementIn" minOccurs="1" maxOccurs="10000" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="GetEntitlementBatchResponse">
  <xs:annotation>
    <xs:documentation>A batch response from the GetStudentEntitlement method.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ResponseControlTable" minOccurs="1" maxOccurs="1" />
      <xs:element ref="GetEntitlementOut" minOccurs="0" maxOccurs="10000" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```
D.7  GetEntitlementRealTime.xsd

This file describes the Request and Response types for the real time version of the GetEntitlement method.

```xml
<?xml version="1.0" encoding="utf-8" ?>
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.3">
<xs:include schemaLocation="../structures/GetEntitlement.xsd" />
<xs:include schemaLocation="../structures/Technical.xsd" />
<xs:element name="GetEntitlementRequest">
  <xs:annotation>
    <xs:documentation>A real-time request to the GetStudentEntitlement method.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="RequestControlTable" minOccurs="1" maxOccurs="1" />
      <xs:element ref="GetEntitlementIn" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetEntitlementResponse">
  <xs:annotation>
    <xs:documentation>A real-time response from GetStudentEntitlement method.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ResponseControlTable" minOccurs="1" maxOccurs="1" />
      <xs:element ref="GetEntitlementOut" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```
**E. Appendix E – CHESSN Check Digit Algorithm**

To detect errors in the CHESSN introduced during manual data entry, a check digit scheme was necessary. The rightmost digit of a CHESSN is the check digit, which verifies that the first 9 CHESSN digits are correct. The algorithm used follows the ISO 7064 standard – Check character systems.

A 10-digit CHESSN can be represented as \(a_{10} a_9 a_8 \ldots a_1\), where \(a_1\) represents the check digit. Example CHESSN – 7804045871

<table>
<thead>
<tr>
<th></th>
<th>a10</th>
<th>a9</th>
<th>a8</th>
<th>a7</th>
<th>a6</th>
<th>a5</th>
<th>a4</th>
<th>a3</th>
<th>a2</th>
<th>a1</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The algorithm for verifying the check digit \(a_1\) can be described as follows:

With the index \(j=1\) to \(n\), and defining \(P_j = M\) for \(j = 1\) calculate:

\[
S_j := P_j \mod M + a_{n-j+1}
\]

\[
P_{j+1} := S_j \mod 2
\]

The CHESSN is assumed to be correct if \(S_n - 1 \mod M\) is equal to zero.

where

\(n\) is the number of characters in the string including the check-digit.

\(M\) is the number of characters in the character set.

\(\mod M\) is the remainder after dividing by \(M\); if this is zero, then the value of \(M\) shall be substituted.

\(\mod M+1\) is the remainder after dividing by \(M+1\); the remainder is never zero after this operation.

\(a_{n-j+1}\) is the character value.

\(x \mod y\) is the remainder after dividing \(x\) by \(y\).

For verifying a CHESSN, \(n = 10\) and \(M = 10\).

Example, take the CHESSN example as above 7804045871.

We iterate through \(j = 1\) to 10 and start off with \(P_j = 10\) for \(j = 1\).

<table>
<thead>
<tr>
<th>Step</th>
<th>Product carried forward (P_j)</th>
<th>Adjusted producted (P_j \mod 11)</th>
<th>Next character value (a_{n-j+1})</th>
<th>Intermedeiate sum (S_j)</th>
<th>Adjusted intermediate sum (S_j \mod 10)</th>
<th>(x \times 2)</th>
<th>Adjusted (P_{j+1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>+ 7</td>
<td>= 17</td>
<td>7</td>
<td>(7 \times 2)</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>3</td>
<td>+ 8</td>
<td>= 11</td>
<td>1</td>
<td>(1 \times 2)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>+ 0</td>
<td>= 2</td>
<td>2</td>
<td>(2 \times 2)</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>+ 4</td>
<td>= 8</td>
<td>8</td>
<td>(8 \times 2)</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>5</td>
<td>+ 0</td>
<td>= 5</td>
<td>5</td>
<td>(5 \times 2)</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
<td>+ 4</td>
<td>= 14</td>
<td>4</td>
<td>(4 \times 2)</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>+ 5</td>
<td>= 13</td>
<td>3</td>
<td>(3 \times 2)</td>
<td>6</td>
</tr>
<tr>
<td>Step $j$</td>
<td>Product carried forward $P_j$</td>
<td>Adjusted producted $P_j \mid_{11}$</td>
<td>Next character value $a_{c,j+1}$</td>
<td>Intermediate sum $S_j$</td>
<td>Adjusted intermediate sum $S_j \mid_{10}$</td>
<td>$\times 2 = P_{j+1}$</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------------</td>
<td>------------------------</td>
<td>--------------------------------------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6 + 8 = 14</td>
<td></td>
<td>4</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8 + 7 = 15</td>
<td></td>
<td>5</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10 + 1 = 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The final value $S_{10} = 11$.

$S_{10} - 1 \mod 10 = 0$, therefore the CHESSN is correct.

For more information on the algorithm, refer to the ISO 7064 document (reference [8] in Appendix A.1)
F. Appendix F – Business Rules

BR389 Return messages for CHESSN allocation and entitlement information requests

Messages are required for provisional and active CHESSNs to alert users to caveats related to the SLE balance and FEE-HELP balance. These messages will be supplied in the web services and the CHESSN browser interface.

Case 1 - All active CHESSNs

Subcategory 1 - SLE

The available SLE for this active CHESSN is subject to any enrolments and variations that may be applicable subsequent to the 'As at' date.

Subcategory 2 - Scholarships. This message will be displayed only if one or more scholarship categories have been consumed.

Commonwealth Scholarship (CS) data is calculated and displayed for data reported by providers by the 'As at' date.

Subcategory 3 - OS-HELP. This message will be displayed only if there is OS-HELP consumption data.

The available OS-HELP data for this active CHESSN is as per the latest submission from the HEP. As OS-HELP data is reported twice yearly, there may be a discrepancy between the remaining available OS-HELP loan entitlement and that displayed here.

Case 2 - All active CHESSNs for Australian citizen (E358 = 1) Or Permanent humanitarian visa holder (E358 = 8) Or New Zealand (E358 = 2) Or (Permanent visa - other than humanitarian visa (E358 = 3) And (BOTPLS (E490=233) Or Pre-2005 PELS student (E490=220)))

Message for FEE-HELP:

The FEE-HELP balance for this active CHESSN is subject to any enrolments, variations and indexation that may be applicable subsequent to the 'As at' date.

Case 3 - Provisional CHESSNs for Australian citizen (E358 = 1) Or Permanent humanitarian visa holder (E358 = 8) Or New Zealand (E358 = 2) Or 'X'

Message for FEE-HELP:

The FEE-HELP balance may be subject to usage not yet reported to DEST and should be checked with the student.

Case 4 - Provisional CHESSNs for Permanent visa holder - other than humanitarian visa (E358 = 3)

Message for FEE-HELP:

This student may use FEE-HELP only if they are a pre-2005 PELS student or undertaking bridging study for overseas trained professionals.
# G. Appendix G – Business Fields

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>G61</td>
<td>Commonwealth Higher Education Student Support Number (CHESSN)</td>
<td>A unique identifier for each student who accesses Commonwealth assistance for higher education.</td>
</tr>
<tr>
<td>G129</td>
<td>FEE-HELP balance</td>
<td>The amount of the FEE-HELP limit that a student has not used.</td>
</tr>
<tr>
<td>G422</td>
<td>Active CHESSN</td>
<td>A CHESSN that has a record of Commonwealth assistance at a census date for a student enrolment at a HEP. HEIMS will identify a student's CHESSN as active when data representing Commonwealth assistance for a student is first uploaded to HEIMS. That is, a student's CHESSN is identified as active when data is first loaded for a: 1. Commonwealth supported unit of study (G624); 2. HELP loan; or 3. Commonwealth Scholarship (CS). Specifically HEIMS will identify a student's CHESSN as active when: 1. one or more related current student unit of study records, with a census date up to the Approval period 'As At' date, are defined as a: a) Commonwealth supported unit of study (G624); b) FEE-HELP assisted unit of study (G633); or 2. there are one or more current OS-HELP records; or 3. there are one or more current scholarship records.</td>
</tr>
<tr>
<td>G423</td>
<td>Provisional CHESSN</td>
<td>A CHESSN that has been allocated to a student that is not an Active CHESSN (G422). The CHESSN status, for CHESSNs that have been allocated to a student, is limited to the following two statuses; 1. Provisional CHESSN 2. Active CHESSN A CHESSN allocated to a student through the CHESSN allocation process is initially set as a provisional CHESSN and will retain this status until updated to a status of active. Note that a CHESSN may retain the status of provisional indefinitely.</td>
</tr>
<tr>
<td>G607</td>
<td>Ordinary SLE allocated</td>
<td>Defines the process for allocating ordinary SLE to a student. All eligible students will be allocated ordinary SLE on 1 January 2005. All students who become eligible after 1 January 2005 will be allocated ordinary SLE on the day they become eligible. In HEIMS, students will be allocated ordinary SLE with a value of 7 EFTSL on receiving a provisional CHESSN.</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
</tbody>
</table>
| G624 | Commonwealth supported student unit of study (CSU)  
Defines a unit of study undertaken by a student occupying a Commonwealth supported place.  
A Commonwealth supported unit is reported via the Student Data Collection with a student status code on E490 equal to one of the following codes.  
110 Pre-2005 deferred all or part through HECS-HELP  
111 Pre-2005 paid full contribution with HECS-HELP discount  
112 Pre-2005 paid full contribution up-front without HECS-HELP discount  
201 Deferred all or part through HECS-HELP  
202 Paid full contribution with HECS-HELP discount  
203 Paid full contribution up-front without HECS-HELP discount  
250 Pre-2005 merit based undergrad HECS exemption scholarship  
260 Commonwealth supported place with exemption scholarship  
261 A domestic student enrolled in an enabling course  
262 Student undertaking Work Experience in Industry (WEI) where learning and performance is not directed by, and support is not received from, the provider and for which a student contribution cannot be charged. |
| G626 | Enabling course of study  
Is a course of study type that enables a person to subsequently undertake a course of study leading to a higher education award. An enabling course of study does not itself lead to a higher education award or include any course of study that the Minister determines is not an enabling course of study under HESA.  
Reported in the Student Data Collection as E310 = 30. |
| G630 | SLE exempt due to work experience in industry  
A student unit of study does not consume SLE if it is identified as SLE exempt due to work experience in industry.  
That is, a student unit of study does not consume SLE if it represents a:  
1) student undertaking work experience in industry where learning and performance is not directed by the provider, but support is received from the HEP; or  
2) student undertaking work experience in industry where learning and performance is not directed by, and support is not received from, the HEP.  
A student unit of study is defined as SLE exempt work experience in industry if:  
1) Work experience in industry (E337) = 1; or  
2) Work experience in industry (E337) = 2. |
| G633 | FEE-HELP assisted student unit of study  
Defines a student unit of study undertaken by a student that consumes FEE-HELP assistance.  
A FEE-HELP assisted unit of study is reported via the Student Data Collection with student status code E490 equal to one of the following codes.  
230 Deferred all or part of Award or Enabling course tuition fee through FEE-HELP  
231 Deferred all or part of Employee reserved place tuition fee through FEE-HELP  
232 Deferred all or part of Open Universities Australia tuition fee through FEE-HELP  
233 Deferred all or part of BOTP tuition fee through FEE-HELP  
220 Deferred all or part of the tuition fee through FEE-HELP (Pre-2005 PELS student) |
| G669 | FEE-HELP limit  
Identifies the maximum amount of FEE-HELP usage a student can consume over their lifetime (1-Jan-2005 onwards). The FEE-HELP limit is indexed annually from 1-January-2006. |
<table>
<thead>
<tr>
<th>G738</th>
<th>‘As at’ date</th>
<th>Defines the date for which outputted information applies. SLE, FEE-HELP and OS-HELP outputs from HEIMS have an ‘As At’ date equal to the time period end date according to the Approval period (G763) from which the source data was derived. CS outputs from HEIMS are derived from the ‘As At’ date equal to &quot;HEIMS System date - 1&quot; as Commonwealth Scholarship data is processed and updated nightly.</th>
</tr>
</thead>
<tbody>
<tr>
<td>G740</td>
<td>FEE-HELP limit to ‘As at’ date - for output</td>
<td>Identifies the applicable FEE-HELP limit for active CHESSNs for a specified Approval period ‘As at’ date. The value is dependant on: 1) CHESSN status (Active CHESSN (G422)) 2) Citizenship status for active CHESSNs to ‘As at’ date (G781) 3) Student Liability and Status Code (E490) The value for FEE-HELP limit to ‘As at’ date is based on the following conditions. <strong>Case 1</strong> - Australian citizen (E358 = 1) Or Permanent humanitarian visa holder (E358 = 8) - FEE-HELP limit recorded for the calendar year within which the supplied ‘As at’ date falls (Refer to BR240) <strong>Case 2</strong> - New Zealand citizen (E358 = 2) - FEE-HELP limit recorded for the calendar year within which the supplied ‘As at’ date falls (Refer to BR240) <strong>Case 3</strong> - Permanent visa - other than humanitarian visa (E358 = 3) And (BOTPLS student (G782) Or Pre-2005 PELS student (G783)) - FEE-HELP limit recorded for the calendar year within which the supplied ‘As at’ date falls (Refer to BR240) Case Else - Zero (0) Notes 1) If a case statement evaluates to True, the result is set to the value for that case, and the following case statements are not considered. 2) The Case Else statement will include students with Citizenship Status Codes equal to 4, 5, or 9 and (Citizenship Status Code = 3 And which are Not (BOTPLS student (G782) Or Pre-2005 PELS student (G783)).</td>
</tr>
<tr>
<td>G741</td>
<td>FEE-HELP limit for current date</td>
<td>The applicable FEE-HELP limit taking into account indexation (refer to G753, BR240), recorded in a code table for the calendar year within which the supplied current date falls.</td>
</tr>
<tr>
<td>G744</td>
<td>SLE consumption unit of study</td>
<td>Defines a student unit of study undertaken by a student that consumes SLE. A student's allocated SLE is consumed at the end of the census date for all Commonwealth supported units of study (G624), except where the Commonwealth supported unit of study is: 1) reported as part of an Enabling course of study (G626); 2) reported as SLE exempt due to work experience in industry (G630); or 3) NOT a Current student unit of study record (G786).</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Definition</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| G746  | Student Ordinary SLE usage                                                 | The total ordinary SLE usage for a student is derived from a set of current student unit of study records. The value is derived for data within an Approval period. The 'As at' date (G738) defines the end date for an Approval period.  
The ordinary SLE usage is derived for each master CHESSN (representing a single student) from student units of study:  
1) are defined as SLE consumption units of study (G744)  
2) are defined as current student units of study (G786)  
3) have a census date in the Approval period  
The student unit of study EFTSL value reported as E339 in the Student Data Collection is summed for all student unit of study records in this defined data set.  
Student ordinary SLE usage has a maximum value of 7 EFTSL. |
| G747  | Student Ordinary SLE balance                                               | The student ordinary SLE balance is derived from the difference between:  
1) Ordinary SLE allocated (G607)  
2) Student ordinary SLE usage (G746)  
Student ordinary SLE balance = Ordinary SLE allocated (G607) - Student ordinary SLE usage (G746)                                           |
| G749  | Student FEE-HELP usage                                                     | The total student FEE-HELP usage is derived from a set of current student unit of study records. The student FEE-HELP usage is aggregated for each master CHESSN (representing a single student) where student units of study:  
1) have a census date in the Approval period  
2) are defined as FEE-HELP assisted units of study (G633)  
3) are current student unit of study records  
The Student unit of study FEE-HELP usage (G632) reported as E558 in the Student Data Collection is summed for all current student unit of study records in this defined data set.  
If a master CHESSN has no related current FEE-HELP units of study in this defined data set, however is defined as an Active CHESSN (G422) (due to CLS or OS-HELP related data), the Student FEE-HELP usage = 0. |
| G750  | Limited Student FEE-HELP usage to ‘As at’ date                           | The value derived for Student FEE-HELP usage to ‘As at’ date must be limited to the value of FEE-HELP limit for ‘As at’ date. If the ‘As at’ date is in 2005 the limit will be $50,000 and the value of FEE-HELP usage for a student can not be reported externally with a value that exceeds $50,000. |
| G751  | Student FEE-HELP balance to ‘As at’ date                                  | This value is derived by subtracting the FEE-HELP usage for a student (truncated to the whole dollar value) from the derived FEE-HELP limit. Note that the derived FEE-HELP limit will always be a whole dollar value and therefore the derived FEE-HELP balance for a student will result in a whole dollar value.  
Derived from the difference between:  
1) Limited Student FEE-HELP usage to ‘As at’ date (G750) truncated to a whole dollar value.  
2) FEE-HELP limit to ‘As at’ date (G740)  
If the difference is a negative value, FEE-HELP balance to ‘As at’ date is set to zero (0).  
FEE-HELP balance to ‘As at’ date = FEE-HELP limit to ‘As at’ date - for output (G740) - Truncated (Limited Student FEE-HELP usage to ‘As at’ date (G750)) |
| G757  | FEE-HELP balance for provisional CHESSN                                  | Defines the supplied value of FEE-HELP balance when outputted for provisional CHESSNs.  
FEE-HELP balance for provisional CHESSNs is dependant on:  
1. CHESSN status  
2. Citizenship Status Code |
3. Date of output
4. FEE-HELP limit value for current date (G741)

**Period 1** Date range for production release 1 October 2005 to 29 November 2005

**Case 1** - Citizenship status = Australian citizen (E358 = 1) Or Permanent humanitarian visa holder (E358 = 8) Or Permanent visa - other than humanitarian visa (E358 = 3) Or 'X' Or Null
- Null

**Case 2** - Citizenship status = New Zealand citizen (E358 = 2)
- 0 (zero)

**Period 2** Date range for production release 30 November 2005 onwards

**Case 1** - Citizenship status = Australian citizen (E358 = 1) Or Permanent humanitarian visa holder (E358 = 8) Or Permanent visa - other than humanitarian visa (E358 = 3) Or 'X' Or Null
- FEE-HELP limit for current date (G741)

**Case 2** - New Zealand citizen (E358 = 2)
- FEE-HELP limit for current date (G741)

<table>
<thead>
<tr>
<th>G761</th>
<th>Student Ordinary SLE balance for provisional CHESSNs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HEPs, TACs and VET Providers are returned ordinary SLE balance equal to the ordinary SLE allocated (G607) for provisional CHESSNs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>G763</th>
<th>Approval Period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A date period that defines a set of data requiring approval.</td>
</tr>
</tbody>
</table>

When data is approved for a specified purpose, the data must relate to a period of time referred to as an Approval period.

When a HEIMS Data Event is approved it must be in relation to an Approval period.

Approval for updating myUniAssist, CHESSN allocation, and entitlement information requests with new data must be related to an Approval period that has a start date of 1 January 2005. The Approval periods for entitlement outputs are as follows.

1 January to 31 March
1 April to 30 June
1 July to 31 August
1 September to 31 December

Note that approval for data in an Approval period may occur more than once. In this case a record of the multiple approval events will be maintained.

<table>
<thead>
<tr>
<th>G768</th>
<th>FEE-HELP balance for active CHESSNs</th>
</tr>
</thead>
</table>
|      | This value is derived by subtracting the FEE-HELP usage for a student (truncated to the whole dollar value) from the derived FEE-HELP limit. Derived from the difference between:
1) Limited Student FEE-HELP usage to ‘As at’ date (G750) truncated to a whole dollar value.
2) FEE-HELP limit to ‘As at’ date (G740)
If the difference is a negative value, FEE-HELP balance to ‘As at’ date is set to zero (0).
FEE-HELP balance to ‘As at’ date = FEE-HELP limit to ‘As at’ date - for output (G740) - Truncated (Limited Student FEE-HELP usage to ‘As at’ date (G750)) |

<table>
<thead>
<tr>
<th>G782</th>
<th>BOTP student</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A student is defined as a BOTP student if:</td>
</tr>
</tbody>
</table>
1) Load Liability file records have been reported for the student (StudentUnitForCourse table) with a census date up to Approval period ‘As at’ date and are stored as current student unit of study records; and
2) Any student unit of study record, from the set of all current student unit of study records with a census date up to Approval period ‘As at’ date, has a Student Liability and Status Type for BOTPLS (E490 = 233).

| G783 | Pre-2005 PELS student | A student is defined as a Pre-2005 PELS student if:
| 1) Load Liability file records have been reported for the student (StudentUnitForCourse table) with a census date up to Approval period ‘As at’ date and are stored as current records; and
| 2) Any student unit of study record, from the set of all current student unit of study records with a census date up to Approval period ‘As at’ date, has a Student Liability and Status Type for Pre-2005 PELS student (E490 = 220). |

| G785 | Ordinary SLE usage for provisional CHESSNs | Students are returned ordinary SLE usage equal to zero (0) for provisional CHESSNs. |

| G898 | Student Ordinary SLE limit | Defines the maximum ordinary SLE allocated to a student. Students are returned ordinary SLE limit equal to the ordinary SLE allocated (G607) for provisional CHESSNs. |

| G944 | OS-HELP loans used | The OS-HELP loans used is the number of OS-HELP loans a student has used. It is derived from the data records reported in the OS-HELP (OS) file and the OS-HELP Revision file (RO). |

| G945 | OS-HELP loan entitlement | The OS-HELP loan entitlement is the maximum of 2 OS-HELP loans that a student may receive over their lifetime. |

| G946 | OS-HELP loans available | The OS-HELP loans available shows the number of OS-HELP periods that a student has not used and hence are still available. |

| G985 | Maximum CS entitlement for Ordinary Scholarship Category (Maximum ORD) | The maximum entitlement for Ordinary Scholarship Category is 8. See “RefCSEntitlement” table for the maximum entitlement period for each Scholarship Category. |

| G986 | Maximum CS entitlement for Associate Scholarship Category (Maximum ASC) | The maximum entitlement for Associate Scholarship Category is 4. |

| G987 | Maximum CS entitlement for Enabling Scholarship Category (Maximum ENB) | The maximum entitlement for Indigenous Access Scholarship Category is 1. |

| G989 | CS Entitlement balance for Ordinary Scholarship Category (ORD Balance) | ORD Balance = Maximum ORD minus ORD Consumption minus ASC Consumption with no ORD |

| G990 | CS Entitlement balance for Associate Scholarship Category (ASC Balance) | ASC Balance = Maximum ASC minus ASC Consumption |

<p>| G991 | CS Entitlement balance for Enabling Scholarship Category (ENB Balance) | ENB Balance = Maximum ENB minus ENB Consumption |</p>
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>G992</td>
<td>CS Entitlement balance for Indigenous Scholarship Category (IND Balance)</td>
<td>IND Balance = Maximum IND minus IND Consumption</td>
</tr>
<tr>
<td>G998</td>
<td>ENB Consumption (ENB Consumption)</td>
<td>Enabling commonwealth scholarship consumption for a student.</td>
</tr>
<tr>
<td>G999</td>
<td>IND Consumption (IND Consumption)</td>
<td>Indigenous commonwealth scholarship consumption for a student.</td>
</tr>
<tr>
<td>G1035</td>
<td>ORD Consumption (ORD Consumption)</td>
<td>Ordinary commonwealth scholarship consumption for a student.</td>
</tr>
<tr>
<td>G1036</td>
<td>ASC Consumption (ASC Consumption)</td>
<td>Associate commonwealth scholarship consumption for a student.</td>
</tr>
</tbody>
</table>