PUBLIC SCHOOLS OF NORTH CAROLINA
State Board of Education | Department of Public Instruction

HRMS Professional Development
Web Service
API Documentation

Table of Contents

Version History

Version	Release Date	Description
0.5	12/10/2008	Initial Proof of Concept
0.9	06/15/2009	First Beta Release
0.91	06/16/2009	Addition of GetEmployee* & Sandbox
0.92	07/1/2017	Added Digital Learning Credits to the API

Purpose

Provide a technical description of the professional development interface provide by HRMS at the North Carolina Department of Public Instruction (NCDPI).

Audience

1. Any third party wishing to integrate with HRMS.
2. DPI technical personnel who need to support the interface.

Terminology Used In This Document

Abbreviation	Explanation
CEU	Continuing Education Unit (1 CEU = 1.5 Semester Hours)
EmpHRMSID	An employee ID assigned and unique to the HRMS (Human Resource Management System)
HRMS	Human Resources Management System
LEA	Local Education Agency (School District)
NCDPI	North Carolina Department of Public Instruction
PDMS	Professional Development Management System
UID	A unique identifier assigned to Staff through the Staff UID system.
XML	eXtensible Mark-up Language

Vendor On Boarding Checklist

1. Register - Contact HRMS for setup
2. Review the API documentation
3. Test using the sandbox environment with sample tools provided by NCDPI
4. Receive Login information from HRMS
5. Receive a list of Employees for your LEA, only needed when using the EmpHRMSID web services.
6. Coordinate schedule for Piloting the interface with an LEA representative
7. Go Live with the LEA

Overview of Web Services

This XML based data-exchange functionality will allow authorized vendors to exchange data with NCDPI HRMS. Authorized third parties can interact with HRMS by posting SOAP/XML formatted requests (over HTTPS) to the appropriate web services. For each "Create" request, DPI will provide a response in the form of a system assigned Transaction ID. Receipt of this ID confirms the upload was accepted, but does not guarantee the validity of the CEU data.

Each Employee has an identifier provided by NCDPI, separate from their SSN, which NCDPI will require the third party to provide for each CEU processed by the web service. This identified can be either the UID or the EmpHRMSId.

The web service methods currently available are listed below by the type of Staff ID that is being used.

Method Name	Description
CreateCEUTransaction	Authorized third parties will be able to send a package of completed CEU records into HRMS that will update the respective LEA's CEU data. This web service varies based on the EmpHRMSID.
GetCEUTransaction	Authorized third parties can look up a CEU transaction to check the status of each CEU. This web service varies based on the EmpHRMSID or the UID is used.
GetEmployeesByLEA	Authorized third parties can retrieve a list of employees, which includes the EmpHRMSIds, from a specific LEA.
GetEmployeesByLEAandDate	Authorized third parties can retrieve a list of employee, which includes the EmpHRMSIds, from a specific LEA and Date range.
CreatsCEUTransactionUID	Authorized third parties will be able to send a package of completed CEU records into HRMS that will update the respective LEA's CEU data. This web service varies based on the UID.
GetCEUTransactionUID	Authorized third parties can look up a CEU transaction to check the status of each CEU. This web service varies based on the UID.

EmpHRMSID Web Services

Web Service: CreateCEUTransaction (EmpHRMSId)

The CreateCEUTransaction web service method allows an authorized third party to send completed CEU records to HRMS. HRMS will then replicate that data to the specific LEA. Even though the third party may be working with multiple LEAs, each transaction is limited to a single LEA. To send a CEU record to HRMS, the following data elements must be gathered and compiled in an XML package and sent to HRMS via a secure HTTP post.

Web service URL: https://licsalweb.dpi.state.nc.us/hrms/api.asmx

CreateCEUTransaction XML Field/Element Details

Field Name	Description	Req	Type	Example
LEA	3 digit LEA identifier	Yes	String()	123
VendorID	Vendor ID as provided by NCDPI	Yes	String()	PDMSystem
Password	Vendor ID password	Yes	String()	password
EmpHRMSId	HRMS EmployeeID of the person that will receive the CEU credit.	Yes	String()	360512
ClassCode	Class Code ID	Yes	String()	CL123
CompleteDate	Complete Date of Class	Yes	String()	06/01/2007
Credits	Credit Amount	Yes	String()	3
SemesterHours	Hour Amount	Yes	String()	3
CreditDate	Credit Date of Class (Typically same as CompleteDate)	Yes	String()	06/01/2007
Description	Description of Class	Yes	String()	Math
SourceName	Class provider	Yes	String()	NC State
HighlyQualified	Boolean / True False	Yes	String()	0
NoCredit	Boolean / True False	Yes	String()	1
TechCredits	Credit Amount	Yes	String()	0
NclbCredits	Credit Amount	Yes	String()	0
ReadingCredits	Credit Amount	Yes	String()	0
PrincipalCredits	Credit Amount	Yes	String()	0
AcademicCredits	Credit Amount	Yes	String()	0

Example: Send employee CEU records to HRMS

The above data must be compiled into an XML format exactly as noted below. Note this example contains two records. The following is a sample SOAP 1.1 request. The **placeholders** shown need to be replaced with actual values.

```
POST /hrms/api.asmx HTTP/1.1
Host: licsalweb.dpi.state.nc.us
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://hrms.dpi.state.nc.us/CreateCEUTransaction"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateCEUTransaction xmlns="http://hrms.dpi.state.nc.us/">
      <txFacade>
        <Lea>string</Lea>
        <VendorId>string</VendorId>
        <Password>string</Password>
        <CeuList>
```

```
          <CeuFacade>
            <EmpHRMSId>string</EmpHRMSId>
            <ClassCode>string</ClassCode>
            <CompleteDate>string</CompleteDate>
            <Credits>string</Credits>
            <SemesterHours>string</SemesterHours>
            <CreditDate>string</CreditDate>
            <Description>string</Description>
            <SourceName>string</SourceName>
            <HighlyQualified>string</HighlyQualified>
            <NoCredit>string</NoCredit>
            <TechCredits>string</TechCredits>
            <NclbCredits>string</NclbCredits>
            <ReadingCredits>string</ReadingCredits>
            <PrincipalCredits>string</PrincipalCredits>
            <AcademicCredits>string</AcademicCredits>
            <DigLitCredits>string</DigLitCredits>
          </CeuFacade>
          <CeuFacade>
            <EmpHRMSId>string</EmpHRMSId>
            <ClassCode>string</ClassCode>
            <CompleteDate>string</CompleteDate>
            <Credits>string</Credits>
            <SemesterHours>string</SemesterHours>
            <CreditDate>string</CreditDate>
            <Description>string</Description>
            <SourceName>string</SourceName>
            <HighlyQualified>string</HighlyQualified>
            <NoCredit>string</NoCredit>
            <TechCredits>string</TechCredits>
            <NclbCredits>string</NclbCredits>
            <ReadingCredits>string</ReadingCredits>
            <PrincipalCredits>string</PrincipalCredits>
            <AcademicCredits>string</AcademicCredits>
            <DigLitCredits>string</DigLitCredits>
          </CeuFacade>
        </CeuList>
       </txFacade>
    </CreateCEUTransaction>
  </soap:Body>
</soap:Envelope>
```

The web service will receive the request and return a response that appears below. The following is a sample SOAP 1.1 response. The **placeholders** shown need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateCEUTransactionResponse xmlns="http://hrms.dpi.state.nc.us/">
      <CreateCEUTransactionResult>string</CreateCEUTransactionResult>
    </CreateCEUTransactionResponse>
  </soap:Body>
```

```
</soap:Envelope>
```

Example: Update existing CEU records in HRMS

To update a record in HRMS the PDMS must know the unique identifier of the CEU record. The unique identifier of a CEU record consists of the empHRMSId, ClassCode, and CompleteDate. To update the CEU record, submit a new CEU entry that contains the same unique identifiers and it will update the existing entry with the new data. An entry can be updated multiple times, with the last entry being the latest. If the unique identifier fields do not match to an existing entry, a new CEU record will be created based on the three fields.

Only CEU credits created using the API can be modified. Any pre-existing CEUs the LEA may have in their system prior to their implementation of the API will not be updateable by this interface.

The following is a sample SOAP 1.1 request. The **placeholders** shown need to be replaced with actual values.

```
POST /hrms/api.asmx HTTP/1.1
Host: licsalweb.dpi.state.nc.us
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://hrms.dpi.state.nc.us/CreateCEUTransaction"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateCEUTransaction xmlns="http://hrms.dpi.state.nc.us/">
      <txFacade>
        <Lea>string</Lea>
        <VendorId>string</VendorId>
        <Password>string</Password>
        <CeuList>
          <CeuFacade>
            <EmpHRMSId>UNIQUEKEY</EmpHRMSId>
            <ClassCode>UNIQUEKEY</ClassCode>
            <CompleteDate>UNIQUEKEY</CompleteDate>
            <Credits>string</Credits>
            <SemesterHours>string</SemesterHours>
            <CreditDate>string</CreditDate>
            <Description>string</Description>
            <SourceName>string</SourceName>
            <HighlyQualified>string</HighlyQualified>
            <NoCredit>string</NoCredit>
            <TechCredits>string</TechCredits>
            <NclbCredits>string</NclbCredits>
            <ReadingCredits>string</ReadingCredits>
            <PrincipalCredits>string</PrincipalCredits>
            <AcademicCredits>string</AcademicCredits>
            <DigLitCredits>string</DigLitCredits>
          </CeuFacade>
      </txFacade>
    </CreateCEUTransaction>
  </soap:Body>
</soap:Envelope>
```

Example: Deleting existing CEU records in HRMS

To delete a record in HRMS, the PDMS must know the unique identifier of the CEU record. The unique identifier of a CEU record consists of the empHRMSId, ClassCode, and CompleteDate. It is not possible to delete an entire entry from the system; instead a vend or can remove the credits by zeroing out all the data fields. To remove CEU credit, submit a new CEU that contains the same unique identifiers and zero out all the other fields. The existing entry will update and all the credits associated will be cleared.

The following is a sample SOAP 1.1 request. The **placeholders** shown need to be replaced with actual values.

```
POST /hrms/api.asmx HTTP/1.1
Host: licsalweb.dpi.state.nc.us
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://hrms.dpi.state.nc.us/CreateCEUTransaction"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateCEUTransaction xmlns="http://hrms.dpi.state.nc.us/">
      <txFacade>
        <Lea>string</Lea>
        <VendorId>string</VendorId>
        <Password>string</Password>
        <CeuList>
          <CeuFacade>
            <EmpHRMSId>UNIQUEKEY</EmpHRMSId>
            <ClassCode>UNIQUEKEY</ClassCode>
            <CompleteDate>UNIQUEKEY</CompleteDate>
            <Credits>0</Credits>
            <SemesterHours>0</SemesterHours>
            <CreditDate>string</CreditDate>
            <Description>0</Description>
            <SourceName>0</SourceName>
            <HighlyQualified>0</HighlyQualified>
            <NoCredit>0</NoCredit>
            <TechCredits>0</TechCredits>
            <NclbCredits>0</NclbCredits>
            <ReadingCredits>0</ReadingCredits>
            <PrincipalCredits>0</PrincipalCredits>
            <AcademicCredits>0</AcademicCredits>
            <DigLitCredits>0</DigLitCredits>
          </CeuFacade>
      </txFacade>
    </CreateCEUTransaction>
  </soap:Body>
</soap:Envelope>
```

Web Service: GetCeuTransaction (EmpHRMSId)

The GetCeuTransaction web service method allows a third party to check the status of a previously sent CEU Transaction package. To use this method, the following data must be gathered, compiled in an XML file and sent to HRMS via a secure HTTP post. The vendor can only use this method for the LEA(s) with which they are registered.

Web service URL: https://licsalweb.dpi.state.nc.us/hrms/api.asmx

GetCeuTransaction XML Field/Element Details

Field Name	Description	Required	Type	Example
TransactionID	HRMS provided transaction ID. This was returned from the CreateCEUTransaction	Yes	String ()	123
VendorID	Vendor ID as provided by the NCDPI	Yes	String ()	PDMSystem
password	Vendor ID password	Yes	String()	password

Example: Get Transaction Status from HRMS

The above data must be compiled into an XML format exactly as noted below. The following is a sample SOAP 1.1 request. The **placeholders** shown need to be replaced with actual values.

```
POST /hrms/api.asmx HTTP/1.1
Host: licsalweb.dpi.state.nc.us
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://hrms.dpi.state.nc.us/GetCEUTransaction"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCEUTransaction xmlns="http://hrms.dpi.state.nc.us/">
      <transactionId>int</transactionId>
      <vendorId>string</vendorId>
      <password>string</password>
    </GetCEUTransaction>
  </soap:Body>
</soap:Envelope>
```

The web service will receive the request and return a response. The response includes two additional fields that return validation information, IsValid and ValidationMessage. The following is a sample SOAP 1.1 response. The **placeholders** shown need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCEUTransactionResponse xmlns="http://hrms.dpi.state.nc.us/">
      <GetCEUTransactionResult>
        <Lea>string</Lea>
        <VendorId>string</VendorId>
        <Password>string</Password>
        <CeuList>
          <CeuFacade>
            <EmpHRMSId>string</EmpHRMSId>
            <ClassCode>string</ClassCode>
            <CompleteDate>string</CompleteDate>
            <Credits>string</Credits>
            <SemesterHours>string</SemesterHours>
            <CreditDate>string</CreditDate>
            <Description>string</Description>
            <SourceName>string</SourceName>
            <HighlyQualified>string</HighlyQualified>
            <NoCredit>string</NoCredit>
            <TechCredits>string</TechCredits>
            <NclbCredits>string</NclbCredits>
            <ReadingCredits>string</ReadingCredits>
            <PrincipalCredits>string</PrincipalCredits>
            <AcademicCredits>string</AcademicCredits>
            <DigLitCredits>string</DigLitCredits>
            <IsValid>boolean</IsValid>
            <ValidationMessage>string</ValidationMessage>
          </CeuFacade>
        </CeuList>
      </GetCEUTransactionResult>
    </GetCEUTransactionResponse>
  </soap:Body>
</soap:Envelope>
```

Web Service: GetEmployeesByLEA and GetEmployeesByLEAandDate (EmpHRMSId)

The GetEmployees web service method allows a third party to get a list of employee HRMSIDs by LEA or LEA and Date. This method will return enough employee data that can be used to identify employee matches within your system. To use this method, the following data must be gathered, compiled in an XML file and sent to HRMS via a secure HTTP post. The vendor can only use this method for the LEA(s) with which they are registered.

Web service URL: https://licsalweb.dpi.state.nc.us/hrms/api.asmx

GetEmployees* XML Field/Element Details

Field Name	Description	Required	Type	Example
VendorID	Vendor ID as provided by the NC DPI	Yes	String ()	PDMSystem
Password	Vendor ID password	Yes	String()	password
LEA	3 Digit LEA Number	Yes	String()	911
LastModified	Last Modified Time Stamp	Yes	String()	2009-05-07T09:08:52.86

Example: Get Employees by LEA (GetEmployeesByLEA)

The above data must be compiled into an XML format exactly as noted below. The following is a sample SOAP 1.1 request. The **placeholders** shown need to be replaced with actual values.

```
POST /hrmssandbox/api.asmx HTTP/1.1
Host: licsalweb.dpi.state.nc.us
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://hrms.dpi.state.nc.us/GetEmployeesByLEA"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetEmployeesByLEA xmlns="http://hrms.dpi.state.nc.us/">
      <vendorId>string</vendorId>
      <password>string</password>
      <lea>string</lea>
    </GetEmployeesByLEA>
  </soap:Body>
</soap:Envelope>
```

Example: Get Employees by LEA and Date (GetEmployeesByLEAandDate)

The above data must be compiled into an XML format exactly as noted below. The following is a sample SOAP 1.1 request. The **placeholders** shown need to be replaced with actual values.

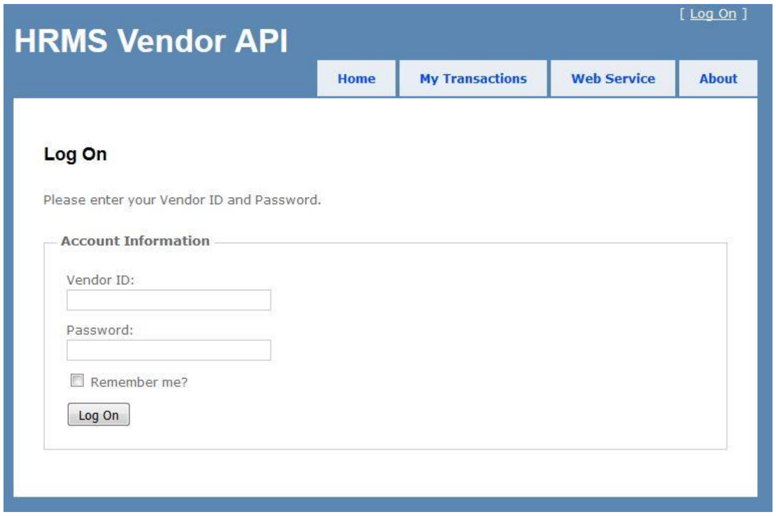
```
POST /hrmssandbox/api.asmx HTTP/1.1
Host: licsalweb.dpi.state.nc.us
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://hrms.dpi.state.nc.us/GetEmployeesByLEAandDate"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetEmployeesByLEAandDate xmlns="http://hrms.dpi.state.nc.us/">
      <vendorId>string</vendorId>
      <password>string</password>
      <lea>string</lea>
      <lastModified>string</lastModified>
    </GetEmployeesByLEAandDate>
  </soap:Body>
</soap:Envelope>
```

Web Reports

https://licsalweb.dpi.state.nc.us/hrms/api

HRMS includes a web report where a PDMS can check the status of transactions. Vendors login with their username and password to review a list of transactions.



Log On

Please enter your Vendor ID and Password.

Account Information

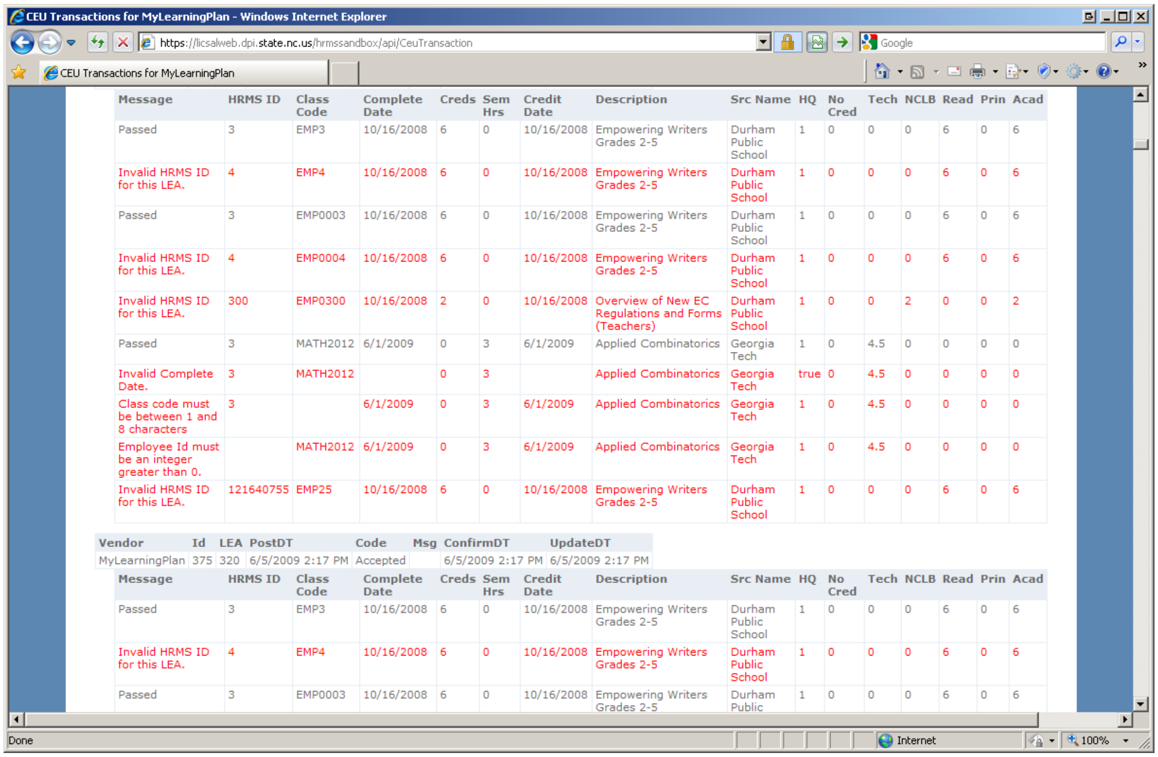
Vendor ID:

Password:

□ Remember me?

Once logged in, the user can see the web report. The report displays failed transactions with the correlated error.

(We will be adding Digital Learning Credits to this page.)



CEU Transactions for MyLearningPlan - Windows Internet Explorer
https://licsalweb.dpi.state.nc.us/hrmssandbox/api/CeuTransaction
Google
CEU Transactions for MyLearningPlan

Message	HRMS ID	Class Code	Complete Date	Creds	Sem Hrs	Credit Date	Description	Src Name	HQ	No Cred	Tech	NCLB	Read	Prin	Acad
Passed	3	EMP3	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6
Invalid HRMS ID for this LEA.	4	EMP4	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6
Passed	3	EMP0003	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6
Invalid HRMS ID for this LEA.	4	EMP0004	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6
Invalid HRMS ID for this LEA.	300	EMP0300	10/16/2008	2	0	10/16/2008	Overview of New EC Regulations and Forms (Teachers)	Durham Public School	1	0	0	2	0	0	2
Passed	3	MATH2012	6/1/2009	0	3	6/1/2009	Applied Combinatorics	Georgia Tech	1	0	4.5	0	0	0	0
Invalid Complete Date.	3	MATH2012		0	3		Applied Combinatorics	Georgia Tech	true	0	4.5	0	0	0	0
Class code must be between 1 and 8 characters	3		6/1/2009	0	3	6/1/2009	Applied Combinatorics	Georgia Tech	1	0	4.5	0	0	0	0
Employee Id must be an integer greater than 0.		MATH2012	6/1/2009	0	3	6/1/2009	Applied Combinatorics	Georgia Tech	1	0	4.5	0	0	0	0
Invalid HRMS ID for this LEA.	121640755	EMP25	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6

Vendor	Id	LEA	PostDT	Code	Msg	ConfirmDT	UpdateDT
MyLearningPlan	375	320	6/5/2009 2:17 PM	Accepted		6/5/2009 2:17 PM	6/5/2009 2:17 PM

Message	HRMS ID	Class Code	Complete Date	Creds	Sem Hrs	Credit Date	Description	Src Name	HQ	No Cred	Tech	NCLB	Read	Prin	Acad
Passed	3	EMP3	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6
Invalid HRMS ID for this LEA.	4	EMP4	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public School	1	0	0	0	6	0	6
Passed	3	EMP0003	10/16/2008	6	0	10/16/2008	Empowering Writers Grades 2-5	Durham Public	1	0	0	0	6	0	6

Sandbox

HRMS provides a sandbox environment to help vendors test the interface connection without risking production data.

The sandbox environment can be found here:

Web Site: https://licsalweb.dpi.state.nc.us/hrmssandbox/api/
Web Service: https://licsalweb.dpi.state.nc.us/hrmssandbox/api.asmx

Appendix A: CEU Validation Rules

- CompleteDate is required.
- CompleteDate can't be more than 5 years past
- CompleteDate can't be in the future
- CreditDate is required, but should be usually same as CompleteDate
- CreditDate can't be more than 5 years past
- CreditDate can't be in the future
- (Tech, Nclb, Reading, Principal, Academic, Digital Literacy) credits cannot exceed 150% of the Semester Hours
- (Tech, Nclb, Reading, Principal, Academic, Digital Literacy) credits cannot exceed Renewal Credits
- Source is required (30 chars max)
- Class Description is required (75 chars max)
- EmpHRMSId is required
- EmpHRMSId must be an integer > 0
- EmpHRMSId must strip any leading zeroes
- LEA is required, must be exactly 3 characters
- Must be an existing employee
- ClassCode is required (8 chars max)
- You can only enter Renewal Credits OR Semester Hours but not both
- Renewal Credits cannot be < 0
- Semester Hours cannot be < 0
- Maximum value for Semester Hours is 66
- Maximum value for Renewal Credits is 99

Business Rules

- Each XML package can only contain the CEU data for one LEA.
- An XML package can contain a maximum of 1000 entries.
- Each XML package can contain the CEU data for one too many users, up to 1000 entries.
- CEU records are identified uniquely based on the combination of: empHRMSId, ClassCode, and CompleteDate.
- CEU records can be overwritten if matched by an existing: empHRMSId, ClassCode, and CompleteDate, with the most recent information displaying last.
- CEU records can be cleared by updating the record and zeroing out the credit hours.

Appendix B: Frequently Asked Questions

Question: Are social security numbers or other sensitive data being sent through the data exchange?
Answer: No sensitive data is being transmitted through the data exchange. All employees will be assigned an EmpHRMSID which will replace SSN as the unique identifier. If you are using the EmpHRMSID, Before you begin using the system, HRMS will provide you with a onetime EmpHRMSID sync to establish the EmpHRMSID to SSN correlation.

Question: When I send CEU data through, does it get updated instantaneous to the LEA database?
Answer: Although the post is processed immediately at HRMS, there is a delay before the data is updated at the LEA side. This is due to the fact that the web service resides at DPI centrally, and all CEU data is housed locally at the LEA location. Typically data will be replicated down to the LEA level twice a day, but the results of your transaction can be viewed almost immediately after it is sent.

Question: I don't have experience writing Internet applications, what is a Secure HTTP Post?
Answer: HTTP (HyperText Transfer Protocol) is the protocol that is used to interact with a Web Server. Each time you type a URL in a web browser (i.e. http://www.yahoo.com), the browser uses HTTP to connect with the server and request the document. HTTP defines how requests are made and how results are returned to the browser. When you begin to write your Data Exchange program you will have to use HTTP. Fortunately, there are many well written libraries, objects, and components to do the dirty work for you. If you use a MS-Windows environment, a free system object is included with newer version of Internet Explorer (it's installed automatically) that handles XML (MSXML) and HTTP (MSXMLHTTP). See the accompanying sample code for more details. Secure HTTP (also knows as HTTPS) is an encrypted version of HTTP which makes requests sent over the internet virtually impossible to be read by unintended recipients. Post is a method (one of two) of sending the data to the server. You are probably familiar with the GET method, whereby parameters are passed on the URL like http://calendar.yahoo.com?mode=month&day=15 (everything after yahoo.com is a parameter). The Get method is appropriate for small numbers of parameters. The Post Method places the parameters in the header of the HTTP request, so they don't appear in the URL. The Post method can support an almost unlimited numbers of parameters.

Question: I am using a system other than MS-Windows. What should I do?
Answer: Check to see if you have a collection of HTTP libraries, objects, classes, etc on your system. You want a solution that allows you to easily take a text file (XML) and send it via Secure HTTP Post to a web page and receive a text file (XML) back. Normally, HTTP libraries are part of the TCP/IP protocol suite. If you don't have the libraries available, you may be able to obtain them from a third party. If all else fails you could always use a Windows based PC as a "gateway" between your system and HRMS.

Question: What is XML?
Answer: XML (eXtensible Markup Language) is a set of document structure guidelines. There are documents (files) and elements (data items). XML makes it possible for dissimilar systems to exchange data as long as both systems use the same structure guidelines. For example, if we wanted to exchange first and last name data, our document may have the following elements <FirstName>John</FirstName> and <LastName>Doe</LastName>. Element names begin in <> and end with the exact name preceded by a / (like HTML). The difference between XML and HTML is that XML is much more rigid in its specification, and is case sensitive. For example, <FirstName>John</firstname> is not valid XML because the closing element </firstname> does not have the same case as opening element <FirstName>. In addition – each opening element <FirstName> MUST have a closing element </FirstName>. An XML Document is a collection of Elements inside a <root></root> pair. XML is plain text.

Question: When building the Data Exchange program, do I just create a big XML string with

elements?
Answer: Yes, that is one way to do it, but you are then responsible for making sure the document is VALID. Another way to do it is to use an XML DOM (Document Object Model) that allows you to create the document, add and remove elements, etc. To use a DOM you need to find a DOM Library, component, class, etc. A DOM object (like the freely available MSXML Component) gives you programmatic access to the contents of a document. Thus, there are commands to create the document, add elements, delete elements, and even transform them (sort, filter, etc) using other tools like XLS. The benefit of using a DOM is that your code will always produce clean, valid XML. The downside to using DOM is that it may take many lines of code to generate a small amount of XML.

Question: Can you recommend some resources we can use to better understand XML?
Answer: Microsoft MSDN XML Developer Center: http://msdn.microsoft.com/xml/default.asp
The XML Standard: http://www.w3.org/XML/
Annotated XML Standard from xml.com: http://www.xml.com/axml/axml.html
Some Good Tutorials: http://www.geocities.com/SiliconValley/Peaks/5957/xml.html