Blackboard DEVELOPERS CONFERENCE 2011

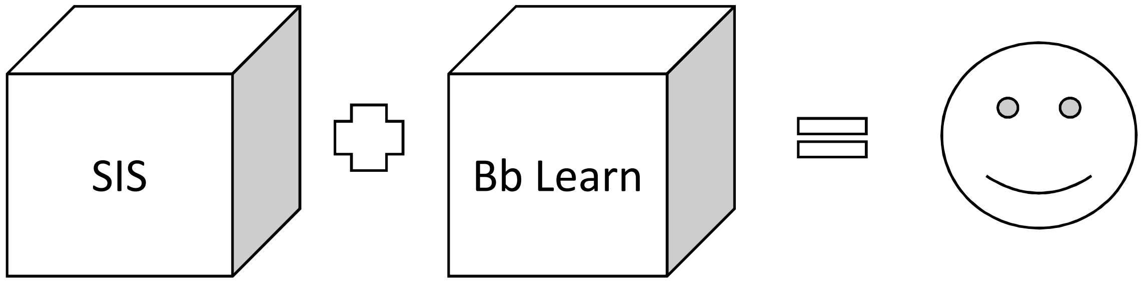
Custom SIS Integration Type Development

Jim Riecken (jim.riecken@blackboard.com)
Blackboard Learn Product Development

Blackboard

Outline

- What is the Student Information System (SIS) Integration framework?
- How does it work?
- How to implement your own custom integration type.



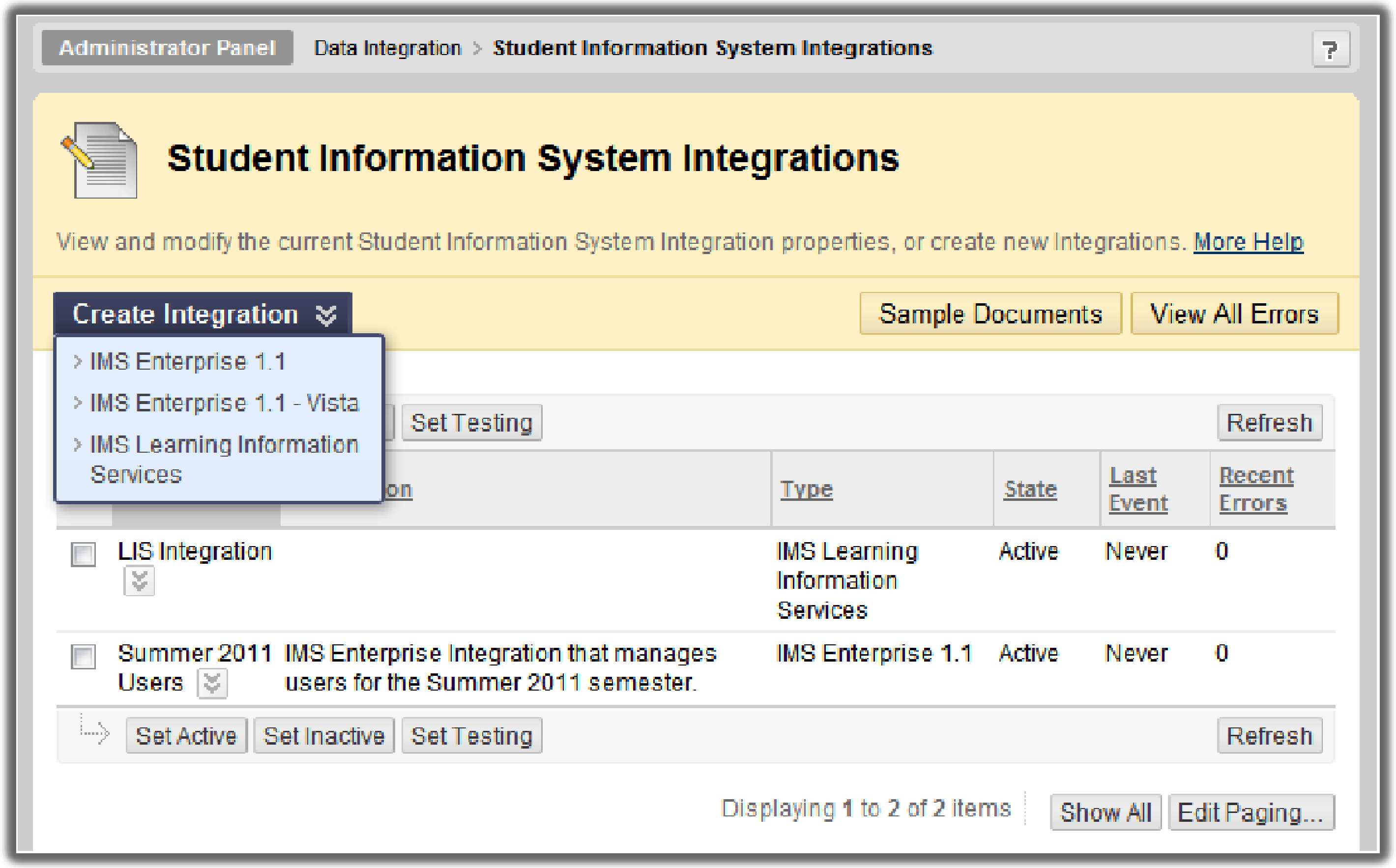
What is the SIS Integration Framework?

- Like Snapshot
  - Takes in data feeds from SIS
  - Processes and manipulates the data
  - Creates, updates, or deletes records in Learn
- Unlike Snapshot
  - Completely UI-based for admin. No command line!
  - Allows custom integration types to be created
  - Not forced to generate data in a proprietary format
  - Allows incoming data to be transformed using scripts

What is the SIS Integration Framework?

- Integration types are implemented as Blackboard Building Blocks™
  - – 9.1 SP6 ships with 2 such Blackboard Building Blocks
    - IMS Enterprise 1.1
      - – Handles IMS compliant XML (as well as Vista-specific extensions) via HTTP POSTs
    - IMS Learning Information Services
      - – Exposes LIS Web Services
        - » Person, CourseSection, Membership, Bulk Data
        - » Released source code! Check EduGarage

Managing SIS Integrations



Administrator Panel
Data Integration > Student Information System Integrations
?

Student Information System Integrations

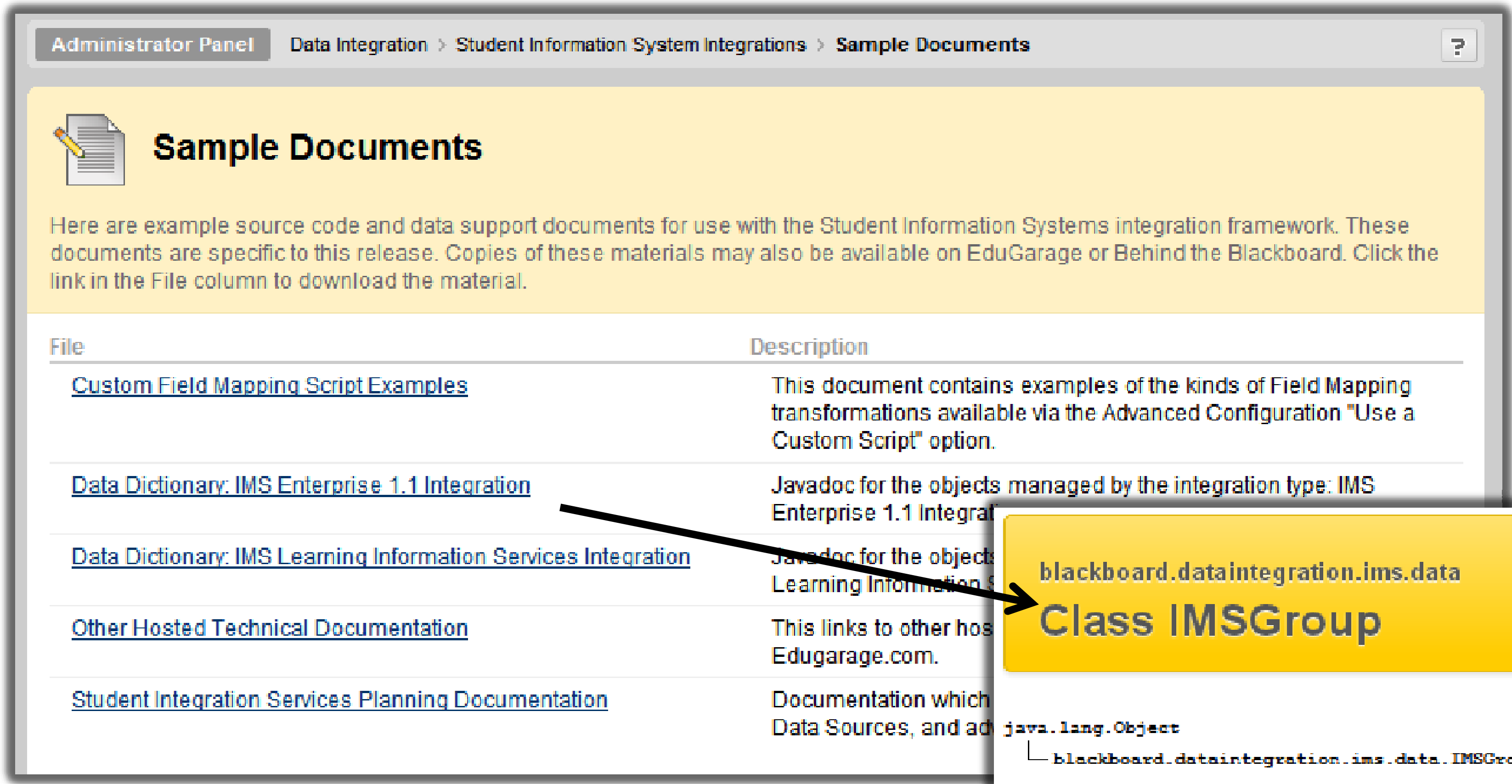
View and modify the current Student Information System Integration properties, or create new Integrations. More Help

Create Integration
IMS Enterprise 1.1
IMS Enterprise 1.1 - Vista
IMS Learning Information Services
Sample Documents
View All Errors
Set Testing
Refresh

	…on	Type	State	Last Event	Recent Errors
☐	LIS Integration	IMS Learning Information Services	Active	Never	0
☐	Summer 2011 Users — IMS Enterprise Integration that manages users for the Summer 2011 semester.	IMS Enterprise 1.1	Active	Never	0

Displaying 1 to 2 of 2 items

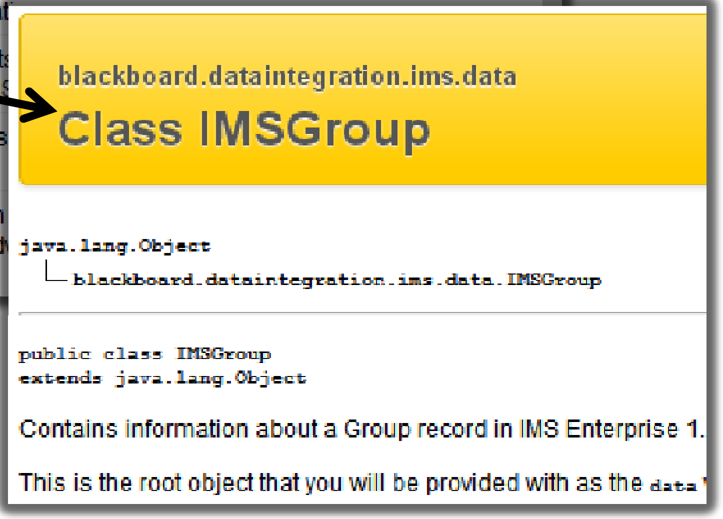
Inline Documentation


Administrator Panel
Data Integration > Student Information System Integrations > Sample Documents

Sample Documents

Here are example source code and data support documents for use with the Student Information Systems integration framework. These documents are specific to this release. Copies of these materials may also be available on EduGarage or Behind the Blackboard. Click the link in the File column to download the material.

File	Description
Custom Field Mapping Script Examples	This document contains examples of the kinds of Field Mapping transformations available via the Advanced Configuration "Use a Custom Script" option.
Data Dictionary: IMS Enterprise 1.1 Integration	Javadoc for the objects managed by the integration type: IMS Enterprise 1.1 Integrat
Data Dictionary: IMS Learning Information Services Integration	Javadoc for the objects Learning Information S
Other Hosted Technical Documentation	This links to other hos Edugarage.com.
Student Integration Services Planning Documentation	Documentation which Data Sources, and ad



blackboard.dataintegration.ims.data

Class IMSGroup

```
java.lang.Object
  └ blackboard.dataintegration.ims.data.IMSGroup
```

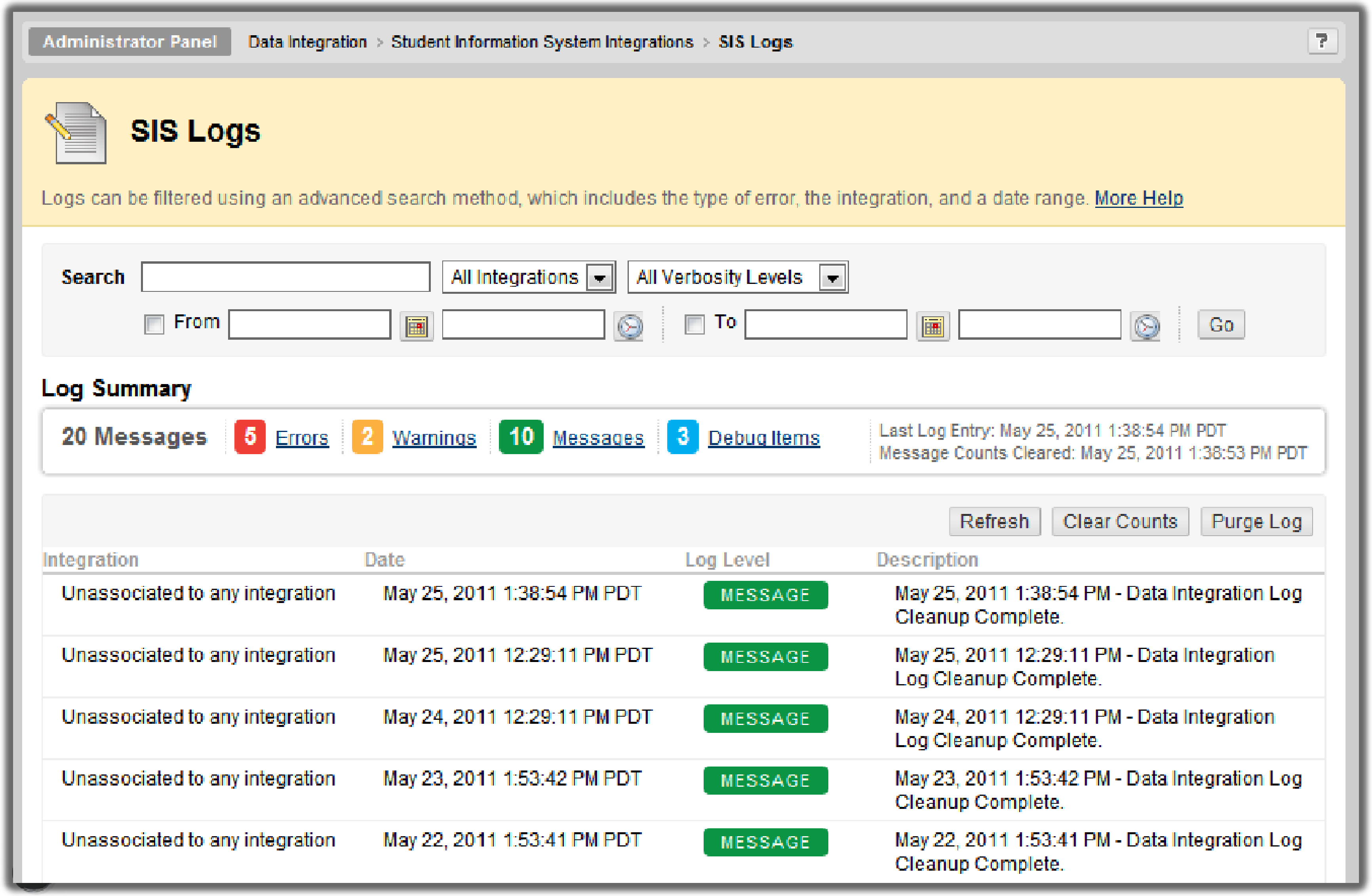
```
public class IMSGroup
extends java.lang.Object
```

Contains information about a Group record in IMS Enterprise 1.

This is the root object that you will be provided with as the data

Blackboard
DEVELOPERS CONFERENCE 2011

Logging


Administrator Panel
Data Integration > Student Information System Integrations > SIS Logs
?

SIS Logs

Logs can be filtered using an advanced search method, which includes the type of error, the integration, and a date range. More Help

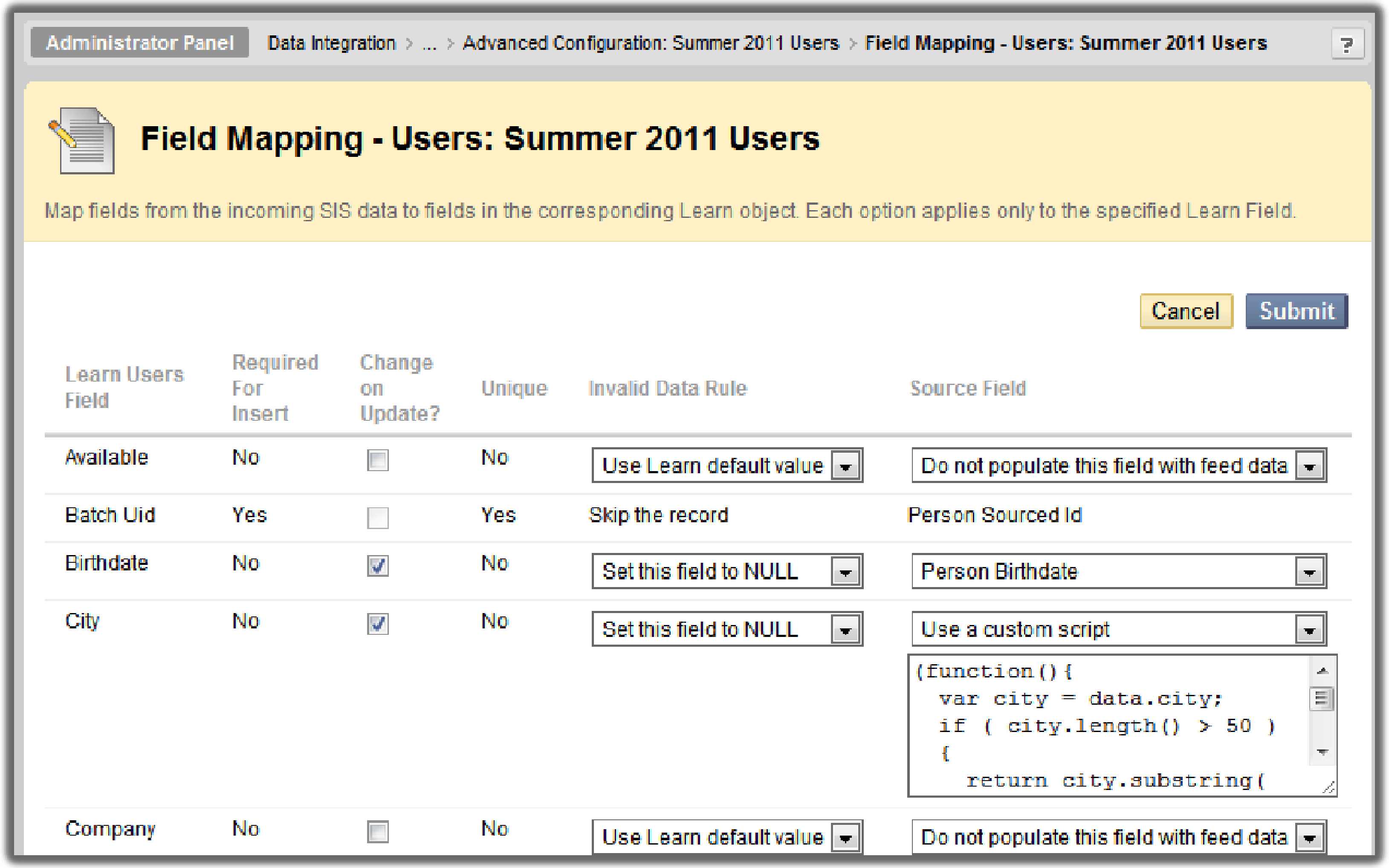
Log Summary

20 Messages | 5 Errors | 2 Warnings | 10 Messages | 3 Debug Items

Last Log Entry: May 25, 2011 1:38:54 PM PDT
Message Counts Cleared: May 25, 2011 1:38:53 PM PDT

Integration	Date	Log Level	Description
Unassociated to any integration	May 25, 2011 1:38:54 PM PDT	MESSAGE	May 25, 2011 1:38:54 PM - Data Integration Log Cleanup Complete.
Unassociated to any integration	May 25, 2011 12:29:11 PM PDT	MESSAGE	May 25, 2011 12:29:11 PM - Data Integration Log Cleanup Complete.
Unassociated to any integration	May 24, 2011 12:29:11 PM PDT	MESSAGE	May 24, 2011 12:29:11 PM - Data Integration Log Cleanup Complete.
Unassociated to any integration	May 23, 2011 1:53:42 PM PDT	MESSAGE	May 23, 2011 1:53:42 PM - Data Integration Log Cleanup Complete.
Unassociated to any integration	May 22, 2011 1:53:41 PM PDT	MESSAGE	May 22, 2011 1:53:41 PM - Data Integration Log Cleanup Complete.

Mapping of SIS data fields to Learn data fields



Administrator Panel   Data Integration > ... > Advanced Configuration: Summer 2011 Users > Field Mapping - Users: Summer 2011 Users

Field Mapping - Users: Summer 2011 Users

Map fields from the incoming SIS data to fields in the corresponding Learn object. Each option applies only to the specified Learn Field.

Cancel   Submit

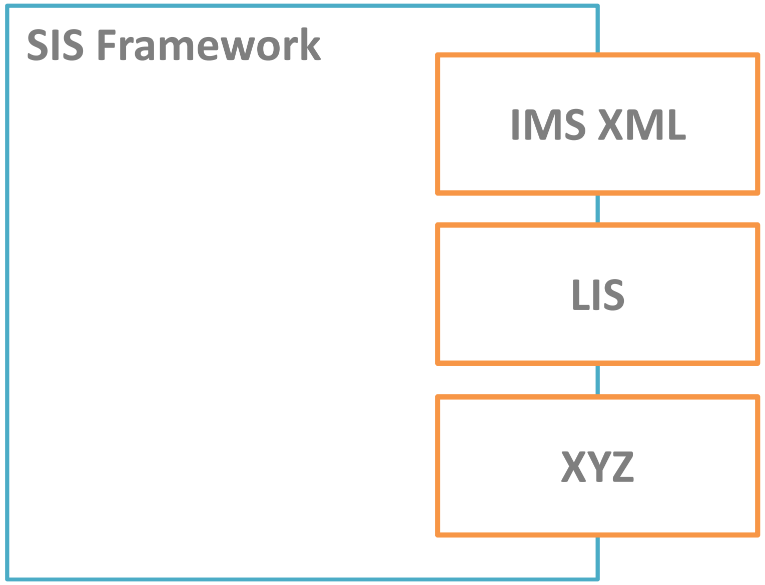
Learn Users Field	Required For Insert	Change on Update?	Unique	Invalid Data Rule	Source Field
Available	No	☐	No	Use Learn default value	Do not populate this field with feed data
Batch Uid	Yes	☐	Yes	Skip the record	Person Sourced Id
Birthdate	No	☑	No	Set this field to NULL	Person Birthdate
City	No	☑	No	Set this field to NULL	Use a custom script
Company	No	☐	No	Use Learn default value	Do not populate this field with feed data

```
(function(){
  var city = data.city;
  if ( city.length() > 50 )
  {
    return city.substring(
```

Blackboard
DEVELOPERS CONFERENCE 2011

How does it work?

- Two main pieces
  - SIS framework
  - Blackboard Building Blocks



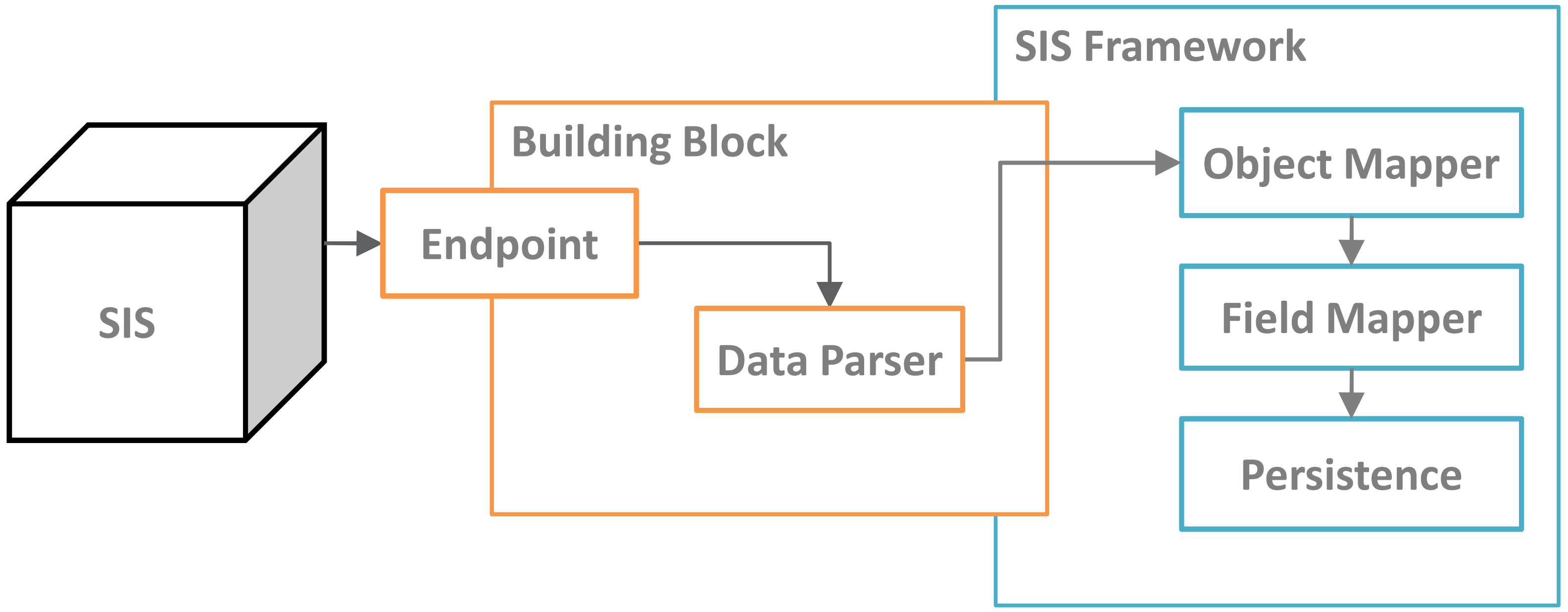
SIS Framework

- UI for managing integrations + field mappings
- UI for logs
- Provides APIs for Blackboard Building Blocks to implement custom integration types.
  - Defines Blackboard Learn™ data types
    - User, Course, Membership, etc.
  - Maps SIS data fields to Blackboard Learn data fields
    - Using scripts
  - Persists/deletes data

Building Blocks

- Define SIS data types
  - Group, Person, etc.
- Define mapping between SIS data types and Learn data types
  - Objects
    - Person ➔ User
    - Group ➔ Course
  - Fields
    - Default scripts – e.g. Person name ➔ User name
- Parse SIS data into Java objects
- Send parsed data into SIS framework

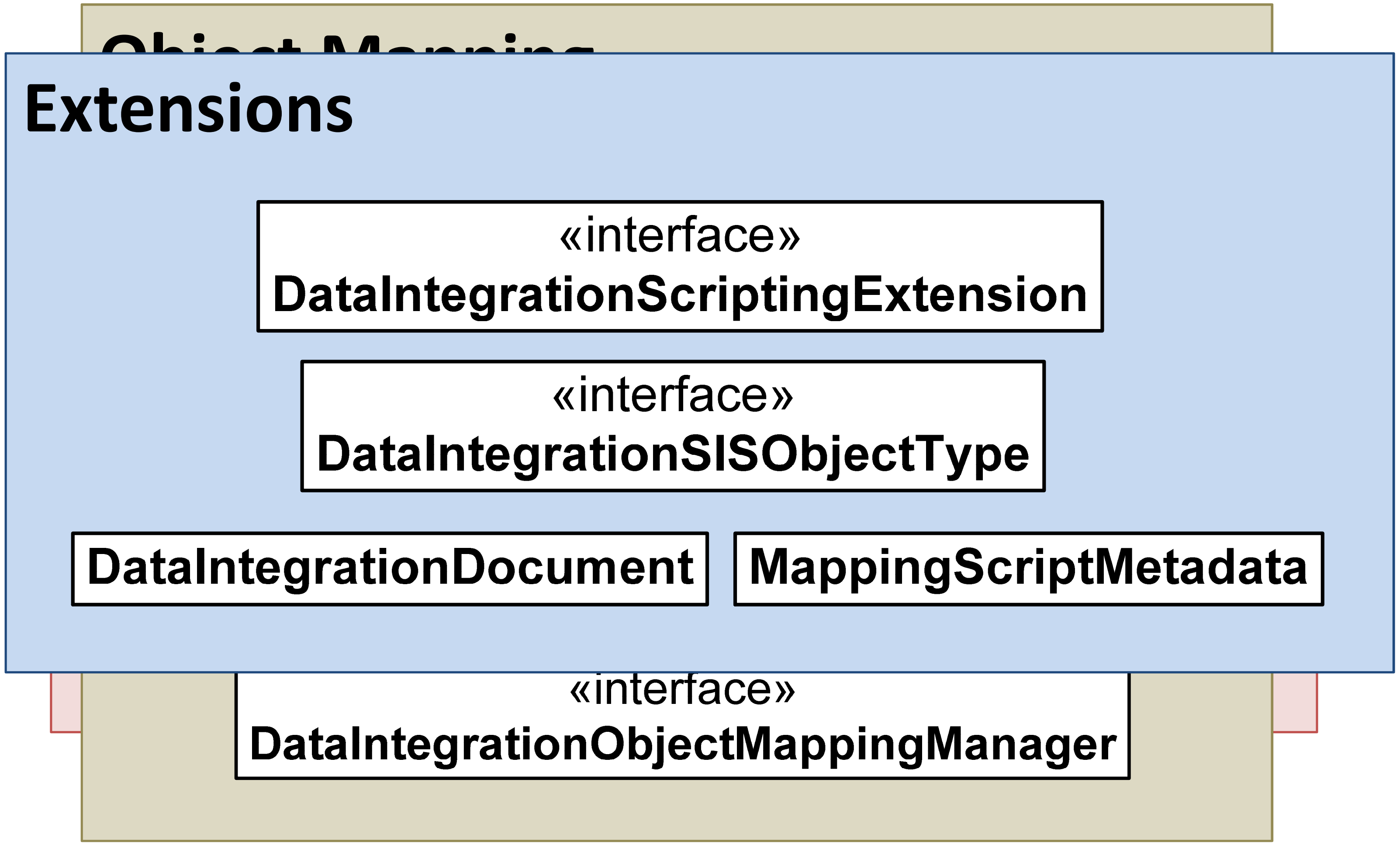
What happens in a typical request



How to implement your own custom integration type

- Use a Blackboard Building Block
- Define integration handler in `bb-manifest.xml`
  - Create/Edit pages
  - Optional custom pages
- Implement SIS Object Type extensions
  - Define default field mapping
  - Create default scripts
- Implement one or more endpoints and invoke SIS APIs
  - Push - HTTP POST, Web Service, Upload JSP, etc.
  - Pull – Database, File System, External URL, etc.

APIs


Object Mapping
Extensions
«interface»
DataIntegrationScriptingExtension
«interface»
DataIntegrationSISObjectType
DataIntegrationDocument
MappingScriptMetadata
«interface»
DataIntegrationObjectMappingManager

APIs

Integration Management

- DataIntegration
- DataIntegrationManagerFactory
- «interface» DataIntegrationManager
- DataIntegrationUtil

Logging

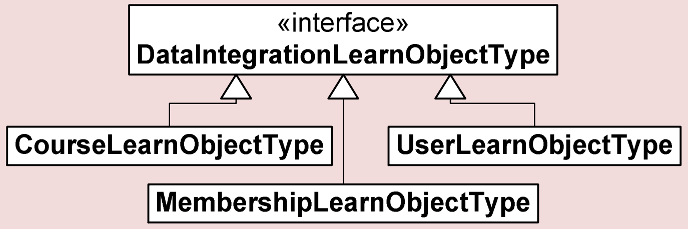
- DataIntegrationLogFactory
- LogLevel
- AggregatingLogger

Authentication

- «interface» DataIntegrationAuthenticator
- DataIntegrationAuthenticatorFactory

Learn Object Types

- AttributeMetadata


«interface» DataIntegrationLearnObjectType
CourseLearnObjectType
UserLearnObjectType
MembershipLearnObjectType

Object Mapping

- DataIntegrationMappingException
- DataIntegrationAttributeMapping
- DataIntegrationObjectMapping
- DataIntegrationObjectMappingView
- DataIntegrationObjectMappingManagerFactory
- «interface» DataIntegrationObjectMappingManager

Extensions

- «interface» DataIntegrationScriptingExtension
- «interface» DataIntegrationSISObjectType
- DataIntegrationDocument
- MappingScriptMetadata

bb-manifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest>
  <plugin>
    <!-- Edited out normal plugin stuff -->
    <webapp-type value="javaext" />

    <data-integration-handlers>
      <data-integration-handler>
        <name value="My Integration Type"/>
        <handle value="vid-my-integration-type"/>
        <create-url value="execute/modifyIntegration?cmd=create"/>
        <edit-url value="execute/modifyIntegration?cmd=edit ><!-- &diId=XXXXX -->
        <links><!-- Custom links are optional -->
          <link>
            <name value="Upload Data"/>
            <action-url value="execute/uploadData"/>
          </link>
        </links>
      </data-integration-handler>
    </data-integration-handlers>

    <extension-defs>
      <definition namespace="blackboard.platform">
        <extension id="vidMyIntegrationSISType"
                   point="blackboard.platform.dataIntegrationSISObjectType"
                   class="my.package.MyIntegrationSISType"
                   singleton="true"/>
      </definition>
    </extension-defs>
  </plugin>
</manifest>
```

Create/Edit Pages

- Responsible for:
  - Creating `DataIntegration` object
  - Setting up `DataIntegrationObjectMapping` mappings between SIS objects and Learn objects
    - `DataIntegrationObjectMappingView` can be useful for showing information in the page

Create/Edit Pages

DataIntegration

- `guid`
- `name`
- `description`
- `typeHandle`
- `integrationState`
- `authPassword`
- `dataSourceBatchUid`
- `batchUidPrefix`
- `logLevel`

DataIntegrationObjectMapping

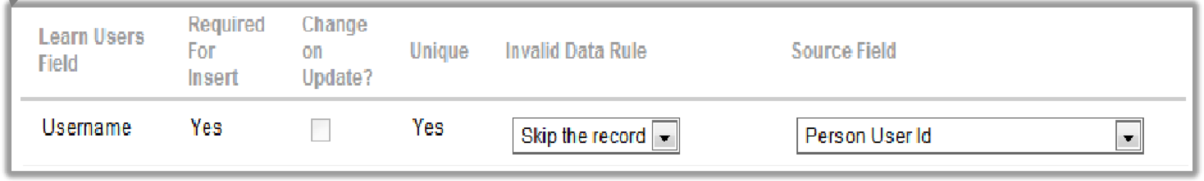
- `dataIntegrationId`
- `sisObjectType`
- `learnObjectType`
- `insertSupport`
- `deleteSupport`

Create/Edit Pages

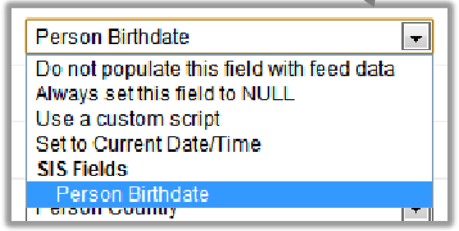
```
DataIntegrationManager diMgr =
  DataIntegrationManagerFactory.getInstance();
DataIntegrationObjectMappingManager omMgr =
  DataIntegrationObjectMappingManagerFactory.getInstance();

DataIntegration di = new DataIntegration();
di.setTypeHandle( "vid-my-integration-type" );
di.setName( "My Integration" );
di.setDescription( "This is my integration" );
di.setDataSourceBatchUid( "MY_DATA_SOURCE" );
di.setIntegrationState( IntegrationState.ACTIVE );
di.setLogLevel( LogLevel.DEBUG );
diMgr.saveDataIntegration( di );

DataIntegrationObjectMapping mapping = new DataIntegrationObjectMapping();
mapping.setDataIntegrationId( di.getId() );
mapping.setSisObjectType( MyIntegrationSISType.TYPE );
mapping.setLearnObjectType( UserLearnObjectType.TYPE );
mapping.setInsertSupport( InsertSupport.SmartUpdate );
mapping.setDeleteSupport( DeleteSupport.DisableOnly );
omMgr.saveMapping( mapping );
```

DataIntegrationSISObjectType

- Extension point interface implemented by B2
- Implement for each "type" of SIS data
  - String getType()
  - String getDisplayName()
  - List<DataIntegrationDocument> getDocumentation()
  - Map<String,DataIntegrationAttributeMapping> getDefaultAttributeMapping(String learnObjectType, DataIntegration dataIntegration)
  - List<MappingScriptMetadata> getMappingScriptMetadata()
  - String getMappingScript(String scriptName)



Learn Users Field	Required For Insert	Change on Update?	Unique	Invalid Data Rule	Source Field
Username	Yes	☐	Yes	Skip the record	Person User Id



Person Birthdate
Do not populate this field with feed data
Always set this field to NULL
Use a custom script
Set to Current Date/Time
SIS Fields
Person Birthdate
Person Country

Blackboard
DEVELOPERS CONFERENCE 2011

DataIntegrationSISObjectType

```java
public class MyIntegrationSISObjectType implements DataIntegrationSISObjectType
{
  public static final String TYPE = "blackboard.platform.vidMyIntegrationSISType";

  @Override
  public String getType()
  {
    return TYPE;
  }

  @Override
  public String getDisplayName()
  {
    return "My Type";
  }

  @Override
  public List<DataIntegrationDocument> getDocumentation()
  {
    DataIntegrationDocument dd =
      new DataIntegrationDocument( PlugInUtil.getUri( "vid", "handle", "path/to/file" ),
                                   "My Doc", "My Custom Documentation" );
    return Arrays.asList( dd );
  }
  @Override
  public Map<String, DataIntegrationAttributeMapping>
   getDefaultAttributeMappings( String learnObjectType, DataIntegration dataIntegration )
  {
    Map<String,DataIntegrationAttributeMapping> result =
      new HashMap<String, DataIntegrationAttributeMapping>();
    DataIntegrationAttributeMapping m = new DataIntegrationAttributeMapping();
    m.setAttributeName( "username" );
    m.setUpdatedOnChange( false );
    m.setInvalidDataRule( InvalidDataRule.SkipRecord );
    m.setScriptName( "usernameScript" );
    result.put("username", m);
    return result;
  }

  @Override
  public List<MappingScriptMetadata> getMappingScriptMetadata()
  {
    MappingScriptMetadata msm = new MappingScriptMetadata();
    msm.setScriptName( "usernameScript" );
    msm.setScriptDisplayName( "My SIS User Name" );
    msm.setReturnType( String.class );
    return Arrays.asList(msm);
  }

  @Override
  public String getMappingScript( String scriptName )
  {
    if ( scriptName.equals("usernameScript") ) return "data.username";
    else return null;
  }
}
```

What attributes are supported on a given Learn Object?

- Look at documentation
- Or, use `getAttributeMetadataForPersistOperation()` method on `DataIntegrationLearnObjectType`
  - Returns metadata about supported fields
  - Can get instance by calling
    - `DataIntegrationObjectMappingManager`'s `getLearnObjectType(String type)` method
      - Where type is the fully qualified extension id
        - » E.g. `blackboard.platform.courseLearnObjectType`
      - All types have a `TYPE` static field that contains this value

Mapping Scripts

- Scripts that map a SIS data object to a specific Learn field.
- JavaScript (Rhino)
  - Object is exposed to the script as “`data`” variable
  - A helper containing some useful utilities is exposed as “`helper`”
    - You can extend this helper by implementing the `DataIntegrationScriptingExtension` extension point

Mapping Scripts: Script Helper

- Methods:
  - `getBatchUid(String id)`
    - Prefix the specified id with the `batchUidPrefix` from the current integration
  - `getXPathString(String xmlString, String xpath)`
    - Given a string of XML, run an XPath query on it
  - `getHelper(String name)`
    - Retrieve a helper defined by a `DataIntegrationScriptingExtension`

Mapping Scripts: Examples

- Suppose we have an instance of the following Person object as our “`data`”

`Person`

- `String getUniqueId()`
- `Name getName()`
- `Map<String,PhoneNumber> getPhoneNumbers()`
- `String getDescription()`

`Name`

- `String getGiven()`
- `String getFamily()`
- `String getMiddle()`

`PhoneNumber`

- `String getType()`
- `String getNumber()`

Mapping Scripts: Examples

```
// Get Person's first name
data.name.given; // Could also be data.getName().getGiven();
```

```
// Get Person's home phone number
data.phoneNumbers.get( 'home' ).number;
```

```
// Get Person's last name followed by their first initial.
data.name.family + ', ' + data.name.given.substring( 0, 1 ) + '.';
```

```
// Truncate the Person's description if it is more than 50 chars
(function(){
  var desc = data.description;
  if ( desc.length() > 50 )
  {
    return desc.substring( 0, 50 );
  }
  else
  {
    return desc;
  }
}());
```

Invoking SIS APIs in endpoint

- Authentication
  - `DataIntegrationAuthenticator`
    - `DataIntegration authenticate(HttpServletRequest request, HttpServletResponse response)`
      - Current implementation supports Basic Auth
      - Username is integration `guid`, password is the `authPassword`
      - Returns the matching `DataIntegration`, or `null`

```
DataIntegration di = DataIntegrationAuthenticatorFactory.
        getAuthenticator().authenticate( req, res );
if ( di != null )
{
  // Request is valid.
}
```

Invoking SIS APIs in endpoint

- Logging
  - SIS framework logs things it does
  - B2 can log its own messages
  - `DataIntegrationLogFactory`
    - `Log getInstanceByDataIntegration(DataIntegration di)`
    - `Log getSystemInstance()`
    - `void startAggregating(DataIntegration di)`
    - `void finishAggregating(DataIntegration di)`
  - Log is instance of `blackboard.platform.log.Log`
    - Standard `logError`, `logWarning`, `logInfo`, `logDebug` methods.

Invoking SIS APIs in endpoint

- Persistence
  - After parsing SIS data into objects
  - DataIntegrationObjectMappingManager
    - `void persistSISObject(String sisObjectType, Object sisObject, DataIntegration integration)`
    - `void deleteSISObject(String sisObjectType, String batchUid, DataIntegration integration)`

Invoking SIS APIs in endpoint: Example - Persist

```
DataIntegrationObjectMappingManager omMgr =
  DataIntegrationObjectMappingManagerFactory.getInstance();
DataIntegration di =
  DataIntegrationAuthenticatorFactory.getAuthenticator().authenticate( req, res );
if ( di != null )
{
  Log log = DataIntegrationLogFactory.getInstanceByDataIntegration( di );
  List<MySISObject> toPersist = parse( req );
  for ( MySISObject sisObj : toPersist )
  {
    try
    {
      DataIntegrationLogFactory.startAggregating( di );
      omMgr.persistSISObject( MySISObjectType.TYPE, sis, di );
    }
    catch ( Exception e ) {
      log.logError( "Error persisting object.", e );
    }
    finally {
      DataIntegrationLogFactory.finishAggregating( di );
    }
  }
}
```

Invoking SIS APIs in endpoint: Example - Delete

```
DataIntegrationObjectMappingManager omMgr =
  DataIntegrationObjectMappingManagerFactory.getInstance();
DataIntegration di = DataIntegrationAuthenticatorFactory.
  getAuthenticator().authenticate( req, res );
if ( di != null )
{
  Log log = DataIntegrationLogFactory.getInstanceByDataIntegration( di );
  List<String> toDelete = parse( req );
  for ( String batchUid : toDelete )
  {
    try
    {
      DataIntegrationLogFactory.startAggregating( di );
      omMgr.deleteSISObject( MySISObjectType.TYPE,
        DataIntegrationUtil.constructBatchUid( batchUid, di ), di );
    }
    catch ( Exception e ) {
      log.logError( "Error deleting object.", e );
    }
    finally {
      DataIntegrationLogFactory.finishAggregating( di );
    }
  }
}
```

Please provide feedback for this session by emailing DevConFeedback@blackboard.com.

The title of this session is:

Custom SIS Integration Type Development