Ed-Fi 2.x School Entity Mapping

Overview

Use this page to help understand and define the mapping of Aeries data to Ed-Fi data

Status

Ready for certification

Document Version 1.5

Last Updated 25 Feb 2020

Links

Entity
https://mappingedu.ed-fi.org/#/entity/378bfca4-4ed2-e811-80d9-005056b36109/info?dataStandardId=db3697f4-4dd2-e811-80d9-005056b36109

Scenarios
https://techdocs.ed-fi.org/display/EDFICERT/Education+Organization+%3E+School+Scenarios

Schema
https://schema.ed-fi.org/datahandbook-v22/Ed-Fi-UDM-Handbook-Index.html#/School579

Root Table

LOC

SQL

Scheduled

LOC_Functions.GetEdFiDataTable_Schools

```
SELECT *
FROM [LOC]
WHERE [DEL] = 0
ORDER BY [CD]
```

Real-Time

LOC_Functions.GetEdFiDataTable_Schools

```
SELECT *
FROM [LOC]
WHERE [DEL] = 0 AND [DTS] >= @TimeSince
ORDER BY [CD]
```

Mappings

Parent	Properties	Type	Required	Table	Field	Other	Code / Notes
\	stateOrganizationId	String	**Required**	LOC	DC		GetString(s("DC"))
\	schoolId	Integer	**Identity**	LOC	CD		GetNullableInteger(s("CD"))

\	nameOfInstitution	String	**Required**	LOC	NM		GetString(s("NM"))
\	shortNameOf Institution	String	**Required**	LOC	NM		GetString(LOC_Functions.Shrink SchoolName(s("NM"))) see function below
\	type	SchoolType	**Required**	LOC	TY		GetEdFiCodeValue(SchoolType, "LOC", "TY", drLOC!TY)
\	addresses	List (Of EducationOrganization Address)	Required				Only one created
addresses	addresses[1]	EducationOrganization Address	Required				
addresses[1]	addressType	AddressType	**Required**			hard code to 'Physical'	EdFiAddressTypes(EdFiAddressTypeEnum. Physical)
addresses[1]	streetNumberName	String	**Required**	LOC	AD		GetString(s("AD"))
addresses[1]	city	String	**Required**	LOC	CY		GetString(s("CY"))
addresses[1]	stateAbbreviationType	StateAbbreviationType	**Required**	LOC	ST		GetUpperCaseString(s ("ST")) Not Mapped
addresses[1]	postalCode	String	**Required**	LOC	ZC		GetString(s("ZC"))
\	educationOrganization Categories	List (Of EducationOrganization Category)	Required				Only one created
educationOrganization Categories[1]	educationOrganization Categories[1]	EducationOrganization Category	Required				
educationOrganization Categories[1]	type	EducationOrganization CategoryType	**Required**			Hard code to 'School'	EdFiEducationOrganization CategoryTypeEnum.School
\	identificationCodes	List (Of EducationOrganization IdentificationCode)	Required				
identificationCodes	identificationCodes[1]	EducationOrganization IdentificationCode	Required				
identificationCodes[1]	identificationCode	String	**Required**	LOC	CC,SC, DC		(GetString(drLOC!CC) & GetString(drLOC!DC) & GetString(drLOC!SC)) For SEA concatenate these fields together CC+SC+DC with no spaces or punctuation. NOTE we may support other types in the future, right now we only support SEA
identificationCodes[1]	educationOrganization IdentificationSystem Descriptor	EducationOrganization IdentificationSystem MapType	**Required**			Hard code to "SEA"	EdFiEducationOrganization Identification SystemTypeEnum.SEA
\	gradeLevels	List (Of SchoolGradeLevel)	Required		Multiple created. There are 2 field values in the LOC table that represent the Low and High end of a range of grades. LO and HI. They are integers. What we do is create a list of grades from the low to high values, using the mapped values and add each one to the list		
gradeLevels	gradeLevels[1]	SchoolGradeLevel	Required				
gradeLevels[1]	gradeLevelDescriptor	GradeLevelMapType	**Required**	LOC	LO, HI	Mapped	GetSchoolGradeLevels (drLOC) see function below
\	institutionTelephones	List (Of EducationOrganization InstitutionTelephone)	Required				Only one created
institutionTelephones	institutionTelephones [1]	EducationOrganization InstitutionTelephone	Required				
institutionTelephones [1]	institutionTelephone NumberType	InstitutionTelephone NumberType	**Required**			Hard code to 'Main'	EdFiInstitution TelephoneNumber TypeEnum.Main

institutionTelephones [1]	telephoneNumber	String	**Required**	LOC	AC, TL		FilterNonNumerics(s("AC") & s("TL")) Values from AC and TL combined into a sequence of integers
\	localEducationAgency Reference	LocalEducationAgency Reference	Required				
localEducationAgency Reference	localEducationAgency Id	Integer	**Required**			Use District Id from EdFiConfiguration	OdsConfiguration.DistrictId
\	shortNameOfInstitution	String	**Conditional**				Not a field in Aeries.

Code

```
Public Shared Function ShrinkSchoolName(SchoolName As String) As String

        Dim strTemp As String
        strTemp = ""

        strTemp = SchoolName
        strTemp = Replace(strTemp, "Junior High School", "JHS")
        strTemp = Replace(strTemp, "Jr High School", "JHS")
        strTemp = Replace(strTemp, "High School", "HS")
        strTemp = Replace(strTemp, "Middle School", "MS")
        strTemp = Replace(strTemp, "Elementary School", "Elem")
        strTemp = Replace(strTemp, "Elementary", "Elem")
        strTemp = Replace(strTemp, "Continuation", "Cont")
        strTemp = Replace(strTemp, "Alternative", "Alt")
        strTemp = Replace(strTemp, "Intermediate", "Int")
        strTemp = Replace(strTemp, "School", "Schl")

        Return strTemp

End Function

Public Function GetSchoolGradeLevels(ByVal drLOC As DataRow) As List(Of SchoolGradeLevel)

        Dim gradeLevels As List(Of SchoolGradeLevel) = New List(Of SchoolGradeLevel)
        Dim gradeLevelStrings As List(Of String) = New List(Of String)

        Try
                Dim lowGrade As Integer = drLOC!LO
                Dim highGrade As Integer = drLOC!HI
                Dim gradelevel As String = String.Empty

                For grade As Integer = lowGrade To highGrade
                        AddGradeLevel(gradeLevels, grade)
                Next

        Catch ex As Exception
                ' TODO log exception then keep mapping
        End Try

        Return gradeLevels

End Function

Private Sub AddGradeLevel(ByRef gradeLevels As List(Of SchoolGradeLevel), ByVal grade As Integer)

        Dim gradelevelDescriptor As String = EdFiDescriptors.GetEdFiCodeValue(EdFiResourceDescriptors.
GradeLevelDescriptor, "STU", "GR", grade.ToString(), OdsConfiguration.ConfigId, DbConnectionString)

        If Not String.IsNullOrEmpty(gradelevelDescriptor) Then

                Dim gradeLevel As SchoolGradeLevel = New SchoolGradeLevel With {
                        .gradeLevelDescriptor = gradelevelDescriptor
                }

                If Not gradeLevels.Any(Function(g) g.gradeLevelDescriptor = gradeLevel.gradeLevelDescriptor)
Then
                        gradeLevels.Add(gradeLevel)
                End If
        End If

End Sub
```