

This document serves as a supplement to the high-level delta published alongside it. Its purpose is to fill in software developers with additional detailed changes that they may have to address as they move from the 2.8 or 3.5 Data Model to Unity. These changes are necessary to having a quality data model release, if unfortunate. Please keep in mind that the overall goal was to combine the 2.8 Data Model with the best of 3.5 to form one united North American Data Model everyone can migrate to and be supported going forward.

### **Address Type & References:**

One of the components brought over with the 3.5 Data Model was the separate Address object, which also automatically generated a new AddressType. Unfortunately, the 2.8 Data Model already had an AddressType for embedded addresses found in various objects. So, after much thought, the existing address type was renamed to OldAddressType. Additionally in places it is commonly used, a parallel structure was added to reference the new Address objects as desired.

### **More “Type” Suffixes:**

The system we have moved to globally to produce our specifications adds “Type” as a suffix to the name of every Type. So, you may encounter more of these in not only the type definitions but also in their usages/references. The 4.0 Data Model Schema and included examples have been validated and all broken references corrected.

### **Fewer Complex Extensions**

A common design pattern in the 3 Data Model was to use Complex Extensions as a base to define a type, even when they were not needed. These have been dropped as accurately as possible. Additionally, it is good to note that the tool used to generate the 3 Data Model in the past breaks on Complex Extensions of Simple Types and falls back on xs:string. For instance, this can be seen in the Excel file accompanying this release wherever the AddressRefIdListType is used. However, the correct base type is used in the published schemas and must be implemented.

### **Wrong Types Corrected**

Sometimes we just have bad inputs from the work done in the past. Wherever these have been found they have been corrected. The only one we are aware of is the “classEndingTime” in the xRoster object which now matches the type of the “classBeginningTime” (xs:time). If you ever run into what you believe is an error, please feel free to share it with the community, so it can be addressed.

### **Changes in Characteristics**

Unity is published using the traditional 2 Data Model tools. As such, some things are more old school than they were in the 3 Data Models. One of these is the characteristics used. They have been converted to the closest match one of: Mandatory (M), Optional (O), or Conditional (C).

## **Better Processing of Schemas**

As we endeavored to convert 3 Data Model components over as completely as possible. It was discovered that some things were processed incorrectly. Most concerning, one condition caused fields to shift. Almost all of this was successfully addressed (see Fewer Complex Extensions above for the exception) and you should encounter a solid SIF Data Model Implementation Specification (NA) 4.0 release.