Module 20 Data Extension

IBM InfoSphere Master Data Management Fundamentals

Unit objectives

After completing this unit, you should be able to:

- Create an MDM Physical Entity Extension
- Understand how an entity extension impacts the existing services
- Understand how an entity extension fits into the InfoSphere MDM Architecture

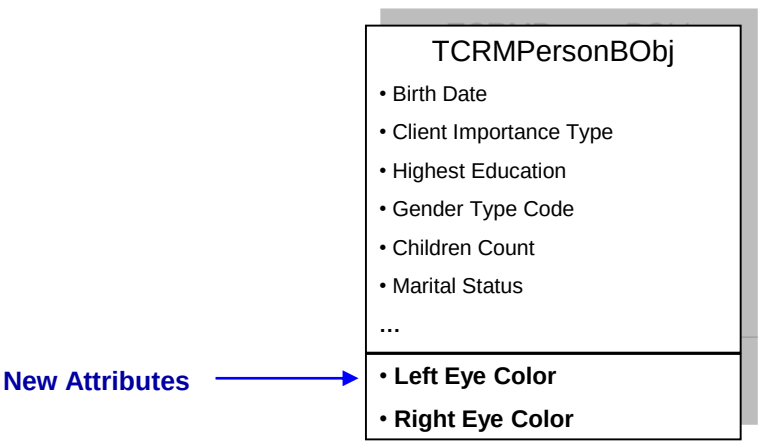
Extending an existing Data Entity

What if we want to store additional attributes on an existing Physical MDM Entity?

We do not have the Person BObj code to modify, so how would I add the new attribute to the Person Entity?

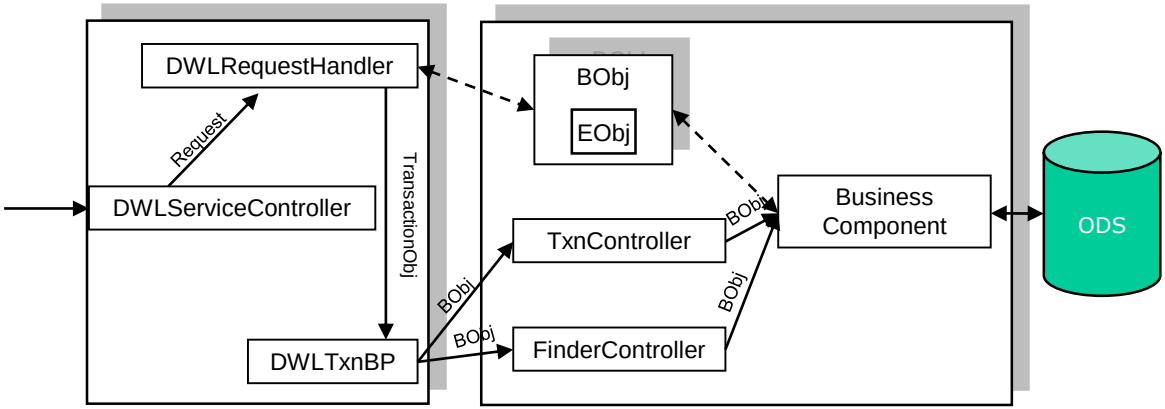
Data Extensions allow you to add additional attributes onto existing Data Entities

Extending Existing MDM Entities

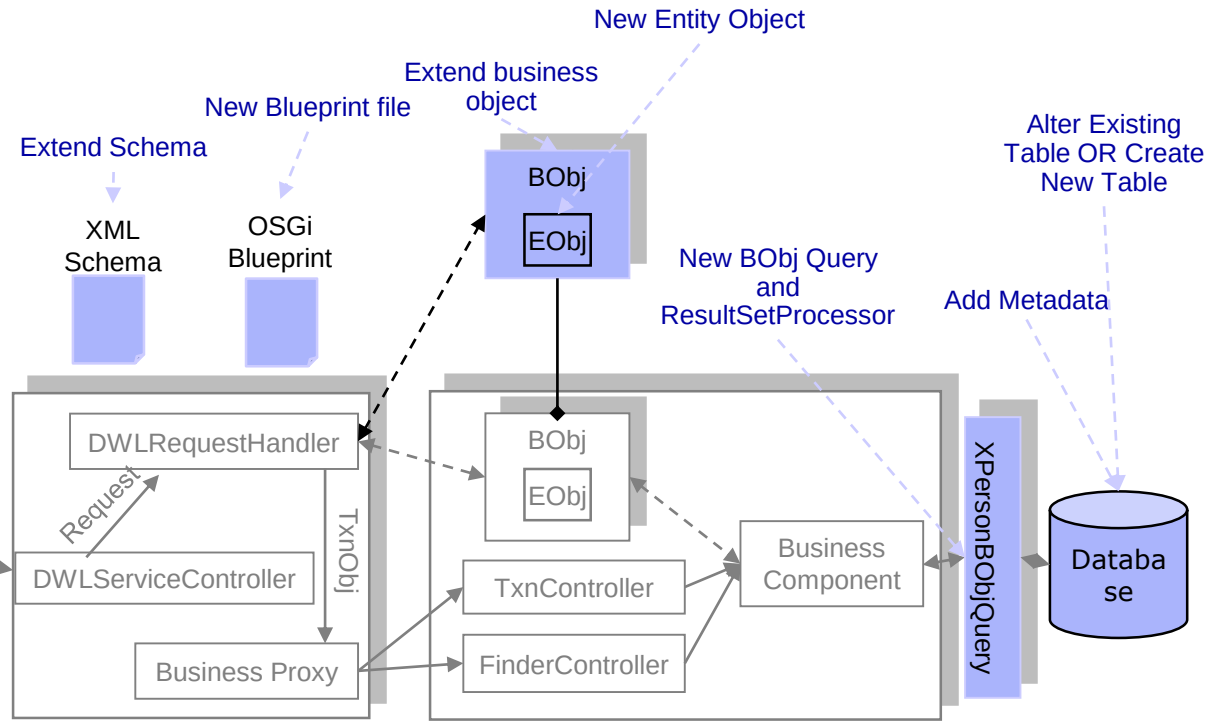


Extensions Architecture: Fill In

What would we need to add to the architecture to add a new attributes to the "Person" Entity along with the services: addPerson, updatePerson, getPerson

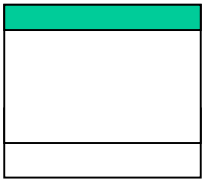


Extension architecture

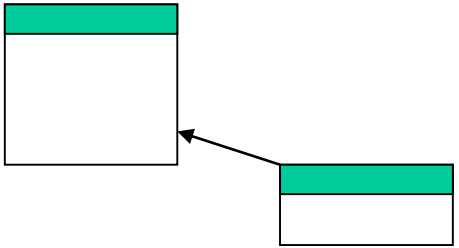


Two Approaches to Data Extension Persistency

- Persist Attributes to Existing Table (Recommended Approach)



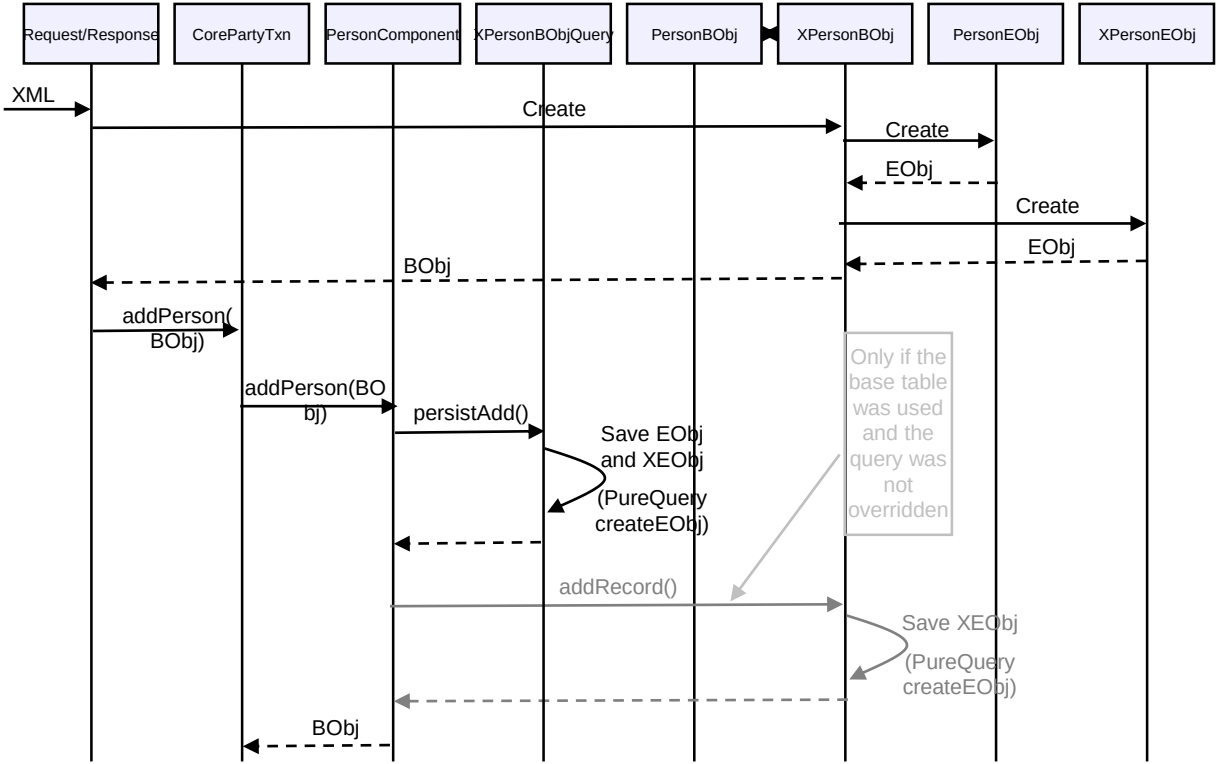
- Persist Attributes to a New Table



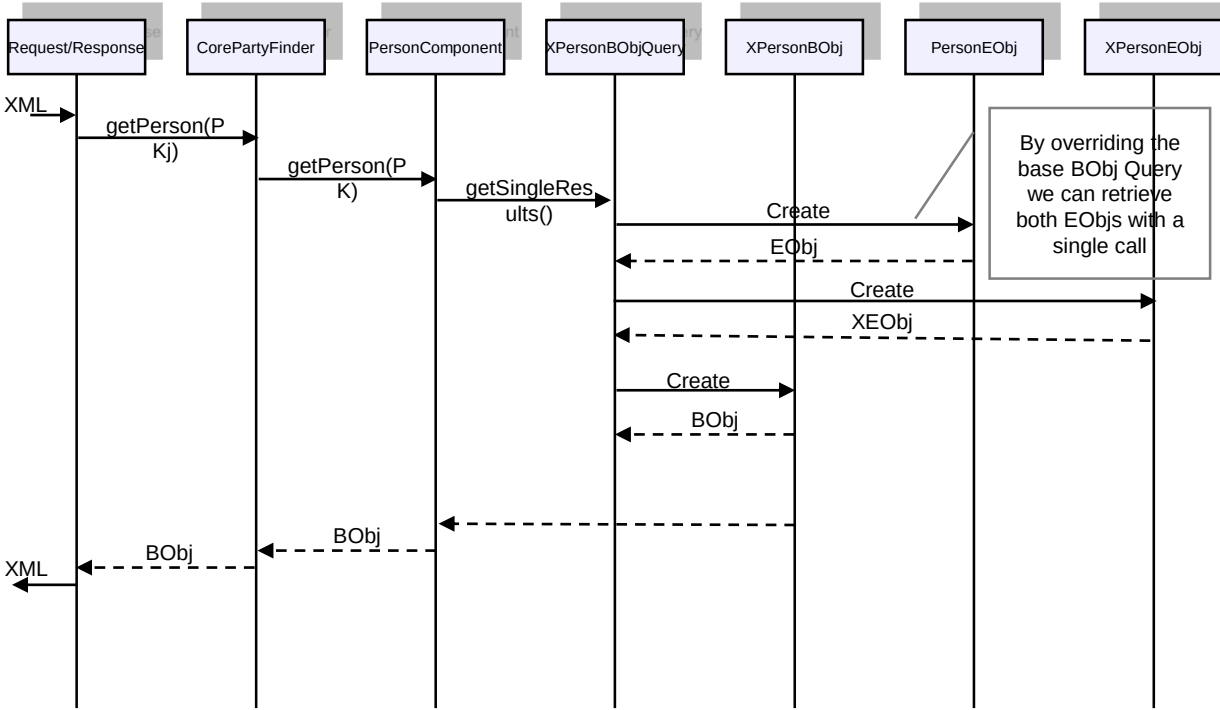
Persistency Guidelines

Consideration	New Table	Alter Existing Table
Compatibility Issues	None	None assuming naming standards followed (prefix column with an “x”).
DB I/O Performance	Inquiry transactions require two DB calls: one for the core table, and another one for the extension table	When using the Inquiry / Persistence framework only one DB call is required selecting, inserting and updating data.
Mapping Extension Attributes	Extension entity object is mapped to the extension table using pureQuery Java annotations.	The extension entity object is mapped to the same table as the original entity object was mapped to, using pureQuery Java annotations.
Ability to Tune	Low	High
Data History	History for the extension columns is kept in its own history table	History for the extension columns is kept in the base history table.
Development Effort	Low	Medium. An extra step is required to “Override base query” to ensure single INSERT/UPDATE SQL statements are used.

The AddPerson sequence



The GetPerson sequence



Exercise introduction

- In this exercise, you will:
  - Create a new Extension to the Person Entity
  - Update the Inquiry Level
  - Run the services addPerson, getPerson, updatePerson with the extension

Exercise overview

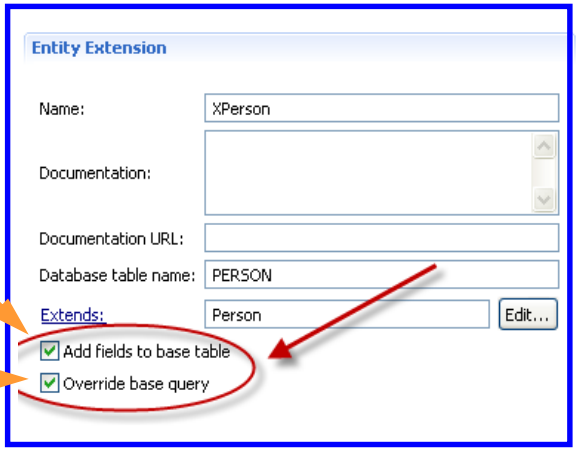
Over the next couple of slides, we will take a high-level look at building a data extension and then give you hands-on practice with the Extensions exercise.

Choosing where the database field will exist…

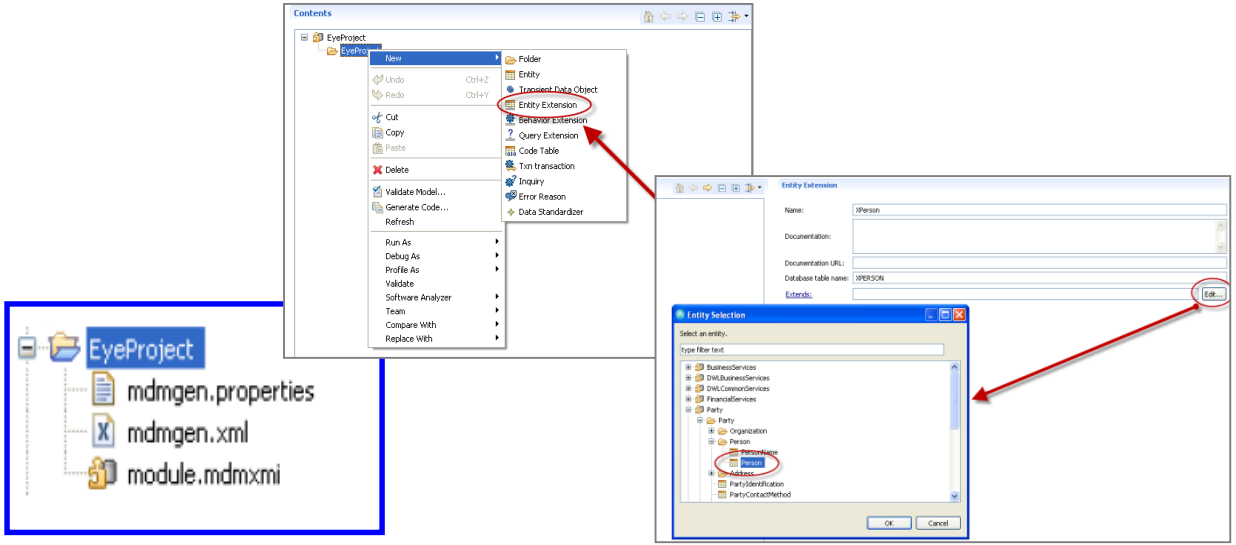
- Supports building extension in brand new table or base table
- Supports customizing BObjQuery for accessing both extend and base attributes in one call

Adding fields to the base table allows us to optimize the queries against the database

Overriding the base query optimizes the extension so only one DB call is required selecting, inserting and updating data.



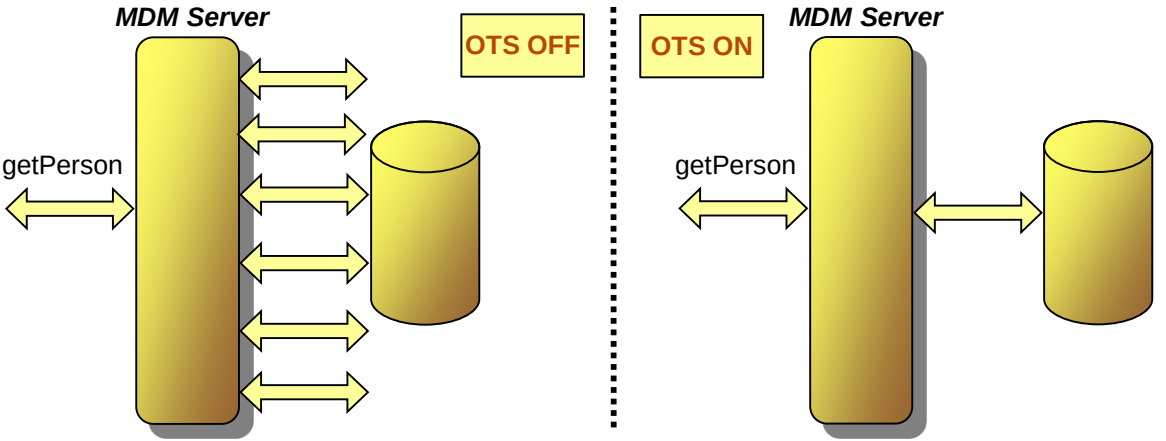
Defining the Extension



The Model Editor

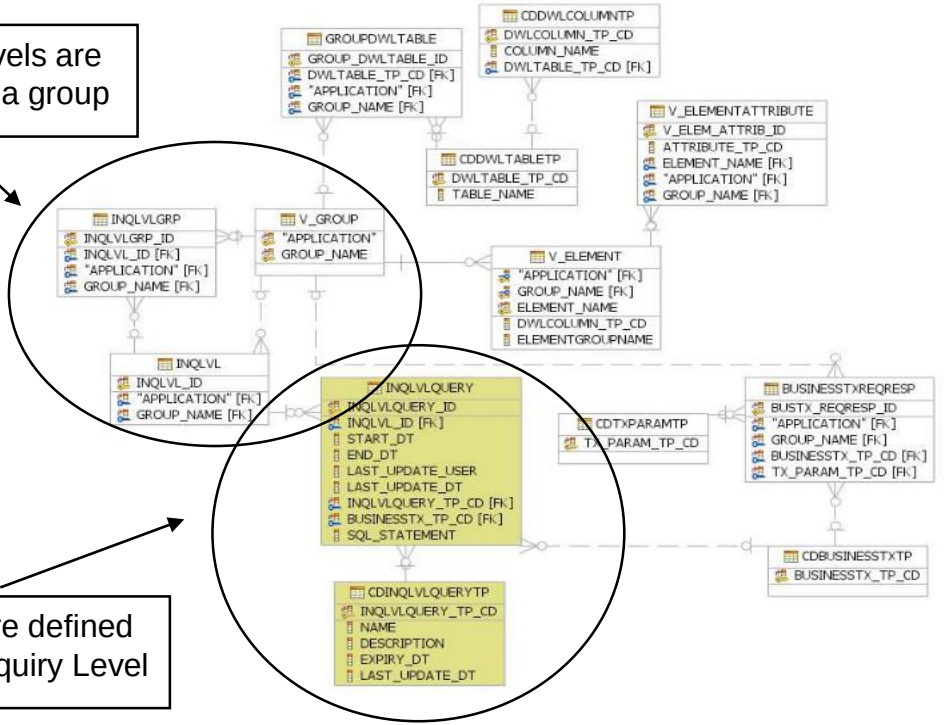
Add a MDM Entity Extension to the project.
Add attributes to the extension.

Inquiry Levels - PluggableSQL Generator



PluggableSQL Generator

Inquiry Levels are defined for a group



SQLs are defined for the Inquiry Level

Unit summary

Having completed this unit, you should be able to:

- Create an MDM Physical Entity Extension
- Understand how an entity extension impacts the existing services
- Understand how an entity extension fits into the InfoSphere MDM Architecture