## Определения:

* Класс синхронизации - класса, реализующего интерфейс ISync
* Объект синхронизации – экземпляр класса синхронизации.
* Метаинформация – информация о событии, инициирующем процесс синхронизации объекта синхронизации.

## Система синхронизации состоит из ХХХ основных компонентов:

Интерфейс ISync и его бизнес-сервер ISyncBS.

Интерфейс содержит в себе единственное свойство - Guid? ObjectPK

ObjectPK является сквозным идентификатором объектов синхронизации, равенство ключей объектов синхронизации из разных систем является признаком «идентичности» объекта.

### Сервис синхронизации ISyncService

Реализации, которого должны обеспечивать следующий функционал:

* void SyncObject(ISync value, bool forced = false);  
  Синхронизация объекта синхронизации.
* void PopEntities();  
  Сброс метаинформации в хранилище; метаинформация из хранилища считается утвержденной.
* void ProceedEntities(bool throwException = false);  
  Обработка метаинформации находящейся в хранилище.
* void SyncType<T>(bool forced = false) where T : ISync;  
  Формирование метаинформации обо всех объектах класса синхронизации.

DefaultSyncService – базовая реализация, обеспечивает накопление метаинформации об объектах синхронизации (ISyncEntity), с последующим извлечением в БД.

### Настройки синхронизации

Определяются между типами синхронизации, содержат в себе указания

AssemblyQualifiedName для реализаций ISyncObserver и ISyncMapper

Типов SyncType, реализующих интерфейс ISync, для источника и приёмника

А также набор связанных настроек SyncSubsetting:

### ISyncObserver – паттерн «наблюдатель», обеспечивает реакцию на изменение объекта синхронизации, должен обеспечивать следующий функционал:

* bool Check(ISync val);  
  Проверить, необходима ли данному объекту синхронизация.
* DateTime? GetDate(ISync val);  
  Получить метку времени события синхронизации (используется в GetSyncEntity)
* ISyncEntity GetSyncEntity(ISync val, bool forced = false);  
  Получить метаинформацию об объекте синхронизации.

ISyncObserver<out T, in TIn> - generic-интерфейс для типизированных классов.

Также имеются базовые реализации ISyncObserver:

* BaseObserver<T, TIn> – является обёрткой интерфейсов ISyncObserver<T, TIn>, ISyncObserver; в наследниках необходимо реализовывать только generic-часть.
* DataObjectObserver<TIn> – базовая реализация наблюдателей для DataObject, в наследниках достаточно переопределить метод View GetSourceView(). Он вызывается в конструкторе класса с кешированием возвращаемого значения. Реализация метода bool Check(TIn val) проверяет наличие измененных свойств объекта синхронизации среди свойств представления, определенного методом GetSourceView().

### ISyncMapper – схож с паттерном «адаптер», обеспечивает преобразование объекта системы А в объект системы Б, должен обеспечивать следующий функционал:

* ISync[] GetSourceObjects();  
  Получить все объекты синхронизации из системы-источника. Используется в ISyncService. SyncType<T>(bool forced = false) для инициализации класса синхронизации.
* void Map(ISyncEntity val);  
  Преобразовать объект из системы-источника в систему-приемник на основании метаинформации. Данный метод является ключевым, весь функционал реализуется в нём.
* object Map(ISync source);  
  Получить объект системы-приемника на основании объекта из системы-источника. Используется при преобразовании сложных свойств исходного объекта, таких как мастера.

ISyncMapper<in T, in TIn, out TOut> - generic-интерфейс для типизированных классов.

Также имеются базовые реализации ISyncMapper:

* BaseMapper<T, TIn, TOut> - является обёрткой интерфейсов ISyncMapper<T, TIn, TOut>, ISyncMapper; в наследниках необходимо реализовывать только generic-часть, можно переопределить метод GetSourceObjects.
* DataObjectMapper<TIn, TOut> - базовая реализация маппера для DataObject, в большинстве случаев достаточно переопределить методы:  
  void Map(TIn source, TOut dest);   
  protected override View GetSourceView();  
  protected override View GetDestinationView();  
  В более редких случаях:  
  Function GetSourceFunction(SyncDOEntity val) – для усиленной фильтрации объектов из системы-источника, например, фильтрация по контексту системы-получателя (территория) или исключение объектов, которые не могут быть преобразованы в объекты системы-получателя (notnull свойства)

## Направления развития:

1. Возможность ратификации метаинформации: допустимы сценарии, при которых метаинформация может попадать в хранилище, но процесс изменения инициатора был прерван (например, ошибка БД).
2. Возможность настройки связанных обсерверов: для случая, когда в сам объект непосредственно маппится свойство мастера (например, денормализация модели), необходимо реагировать на изменение самого мастера, т.к. объект синхронизации при этом не обновляется.
3. Возможность работы сервиса с шиной (имплементация сервиса синхронизации ISyncService)
4. Возможность работы сервиса систем с идентичными схемами: на текущий момент маппинг идентифицируется по имени класса и имени сборки класса.
5. Рефактор базовых реализаций ISyncMapper для возможности использования DataObject -> Non-DataObject: возникла необходимость в такой фиче при маппинге Файл в WebFile (ember), на текущий момент очень много кода приходится переписывать.