

## Экспорт диаграмм классов из Enterprise Architect в Cache

В Enterprise Architect(EA) существует возможность создавать диаграммы классов. Настоящий инструмент позволяет импортировать диаграммы классов EA для генерации классов Cache.

### Пакет поставки

EA.xml - проект на Caché для импорта/экспорта диаграмм классов;  
Example.eap - пример диаграммы классов в EA;  
Example.xml - экспорт диаграммы Example.eap в XMI 2.1.

### Установка

Импортируйте в студию проект EA.xml в область %SYS. Для этого разрешите на запись область %CACHELIB. После компиляции проекта доступ можно сменить снова на ReadOnly.

В меню Инструменты/Add-Ins/Расширения появится пункт "Импорт UML-диаграмм"

### Порядок работы

Создайте UML-диаграмму классов в Enterprise Architect (EA). Пример UML диаграммы можно посмотреть в файле example.eap.

Экспортируйте UML-диаграмму в формат XMI 2.1. Пример экспорта смотрите в файле example.xml.

Откройте Студию и зайдите в меню Инструменты->Add Ins -> Расширения...

Выберите пункт Import UML Diagram

В появившемся окне введите путь к xml файлу, содержащему экспорт UML диаграммы классов из EA. Файл находится на сервере.

### Отображение UML диаграммы в классы Caché

Возможности EA по проектированию диаграмм классов не полностью соответствуют объектной модели Caché. Далее приведены решения и способы представления различных элементов объектной модели Caché с помощью EA.

Преобразование элементов диаграммы классов EA в Cache представлено в таблице ниже:

Элемент EA	Преобразование в Cache
Пакет	Пакет
Класс	Класс
Ассоциация(Association)	Объектная связь по ссылке
Агрегация (Aggregation)	Relationship one-to-many
Композиция (Composition)	Relationship parent-child

Атрибут класса	Свойство класса (property)
Операция (Operation)	Метод

## Особенности описания класса

Атрибуты описания класса в EA и их соответствия в Cache

Атрибут	Значение
Name	Имя класса
Persistence	Укажите для хранимых классов значение persistent
Abstract	Признак абстрактного класса
Notes	Документация класса
IsLeaf	Указать, если класс-Final

## Особенности описания свойств

Атрибуты описания свойств классов в EA и их соответствия в Cache

Атрибут	Значение	Примечание
Name	Имя свойства	
Type	Тип свойства, можно указывать типы Cache - впишите желаемый тип вручную	
Scope	Можно задать атрибуту значения private, public и т.д. Public указываем явно для свойств Cache, Здесь есть некоторое неудобство, т.к. приходится всегда для свойства явно указывать, что оно public	
Initial	Начальное значение	
Notes	документация свойства	
Property	Если использовать это свойство, автоматически добавляются методы set и get	

Attribute Is collection	Укажите, если свойство является коллекцией	Вкладка Detail
Transient	Указать, если свойство не хранится в БД	Вкладка Detail

## Соответствие типов EA и типов Caché

Возможно также указание типов EA, при этом они будут преобразованы в типы Cache. Таблица преобразования:

EA	CACHE
boolean	%Boolean
char	%String
string	%String
short	%Binary
long	%Binary
byte	%String
double	%Float
float	%Float
int or integer	%Integer
void	пусто ( только в методах класса)

## Описание методов

В EA определяется только объявление метода - код метода предполагается реализовать в студии Cache.

Поле	Значение
Name	Название метода
Parameters	Множество параметров метода
Static	Укажите "Да" для метода класса

Return Type	Возвращаемый тип метода. Можно указать тип Cache
Scope	Область видимости метода (private, public)
Notes	Документация
Abstract	Указать, если абстрактный

## Особенности описания параметров

Параметры описания свойств классов в EA и их соответствия в Cache

Атрибут	Значение
Name	Имя свойства
Type	Тип свойства, можно указывать типы Cache - впишите желаемый тип вручную
Default Value	Значение по умолчанию

## Особенности описания связей

Соотношение типов связи EA с связями в Caché.

Enterprise Achitect	CACHE
Association	Объектная ссылка
Generalization	Наследование
Aggregation	One-to-many
Composition	Parent-Child

### 1)Association

Объектная ссылка

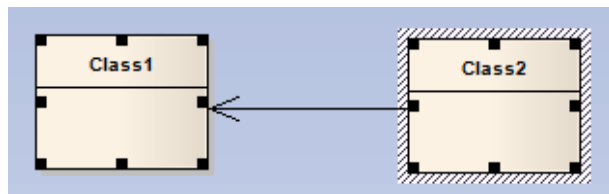
Используется для вставки в класс свойства - объектной ссылки. Целостность не поддерживается.

Выберите тип связи **association** и соедините от класса, у которого должна быть ссылка, к классу, на который будет ссылка.

По этой связи конвертер добавит объектное свойство в класс - источник ссылки (откуда стрелка). Имя свойства возьмется от класса-адресата.

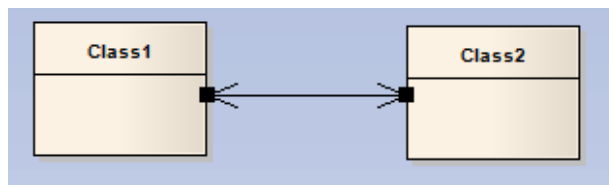
Направление стрелки можно изменить в свойствах связи.

Пример:



В коде класса:

```
Property linkClass1 as Class1;
```



Ссылка друг на друга

При экспорте в Cache:

a) В Class1

```
Property linkClass2 as Class2;
```

b) В Class2

```
Property linkClass1 as Class1;
```

Если необходимо создать ссылки классов друг на друга (реализация связи один-к-одному), укажите направление на обоих концах.

## 2) Aggregation

Агрегация (Aggregation) используется для реализации в классе Cache связи Relationship one-to-many. Выберите тип связи **aggregate** и проведите от класса "MANY" к классу "ONE". Если Вы ошиблись с направлением, то изменить можно 2 способами:

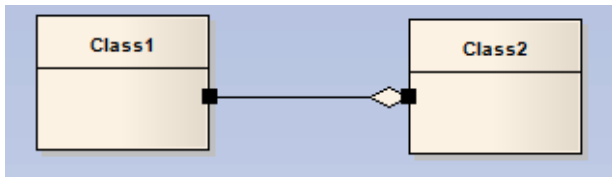
Двойной щелчок мыши по связи

a) Во вкладке General в пункте Direction выбрать необходимое направление

b) Во вкладках Source role и Target role в пункте Aggregation выбрать необходимые значения

При экспорте в Cache в классе в этих классах появится Relationship one - to - many.

Пример:



a) В коде Class1

Relationship myClass2 As Class2 [ Cardinality = one, Inverse = myClass1 ];

b) В коде Class2

Relationship myClass1 As Class1 [ Cardinality = many, Inverse = myClass2 ];

### 3) Composition

Композиция (Composition) используется для реализации в классе Cache связи Relationship Parent-child.

Выберите тип связи **compose** и проведите от класса "Child" к классу "Parent".

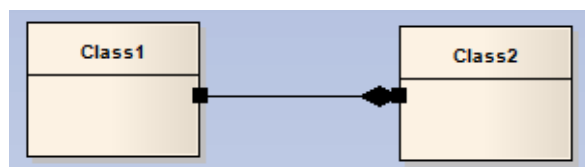
Если ошиблись с направлением, изменить можно 2 способами:

Двойной щелчок мыши по связи

a) Во вкладке General в пункте Direction выбрать необходимое направление

б) Во вкладках Source role и Target role в пункте Aggregation выбрать необходимые значения.

Пример:



При экспорте в Cache в коде классов появится Relationship parent-child:

a) Class1

Relationship parentClass2 As Class2 [ Cardinality = parent, Inverse = childClass1 ];

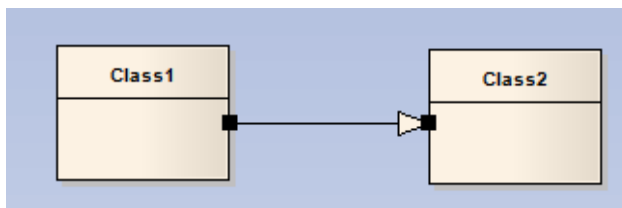
b) Class2

Relationship childClass1 As Class1 [ Cardinality = children, Inverse = parentClass2 ];

### 4) Generalization

Наследование (Generalization). Вид связи, который иллюстрирует, что класс наследуется от другого класса. Выберите тип связи **generalize** и проведите ее от потомка к предку

Пример:



При экспорте в Cache в коде класса Class1 будет:

Class Class1 Extends Class2

## Теги

В Enterprise Architect есть возможность расширять элементы UML с помощью тегов. Теги используются для записи в диаграмму некоторых особенностей модели классов Caché.

Выберите необходимый элемент(Класс, атрибут, метод/операция) -> меню ELEMENT -> add target value -> введите название тега и значение

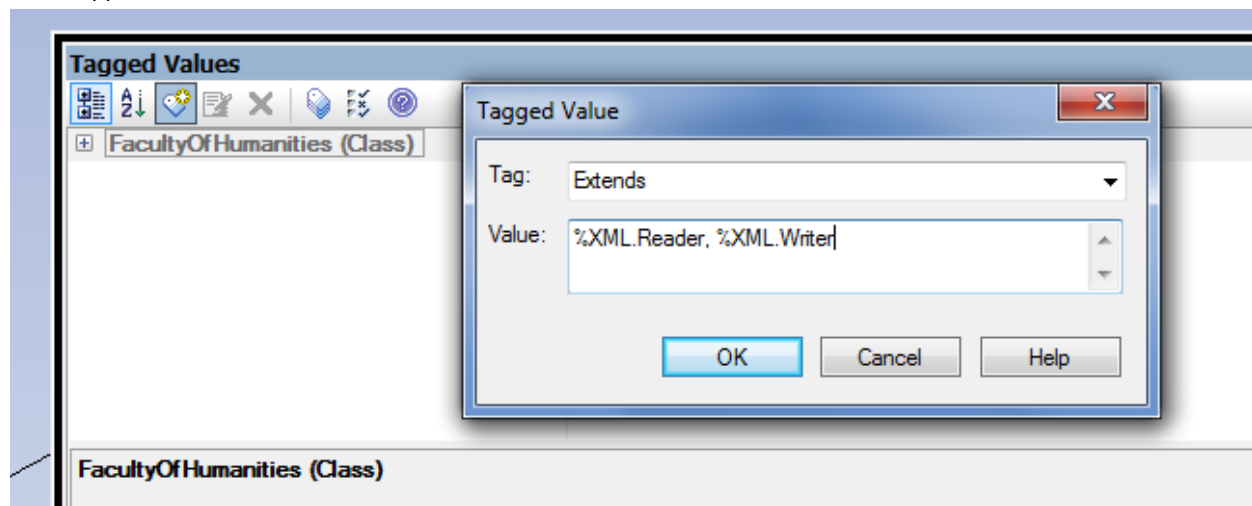
На данный момент поддерживаются теги:

### Теги классов

TAG	VALUE	Значение в CACHE
EXTENDS	Имя предка	Множественное наследование

### Теги свойств (attribute)

Как задавать



TAG	VALUE	Значение в CACHE
Final	Пусто	Final

required	Пусто	Значение свойству должно быть присвоено, прежде чем оно будет сохранено на диске.
readonly	Пусто	Только чтение
PARAMETER	Название параметра = Значение параметра	Параметры атрибута

#### Теги методов(Operation)

TAG	VALUE	Значение в CACHE
Final	Пусто	Финальный метод (нельзя переопределить)
NotInheritable	Пусто	Метод не наследуется в подклассах
WebMethod	Пусто	Указывает, что метод может быть вызван в качестве веб-метода с использованием SOAP протокола.
ZenMethod	Пусто	Должен быть спроектирован на веб-клиента для использования в пределах ZEN