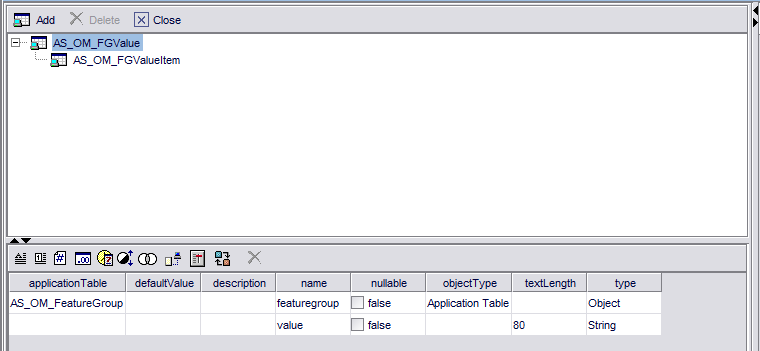
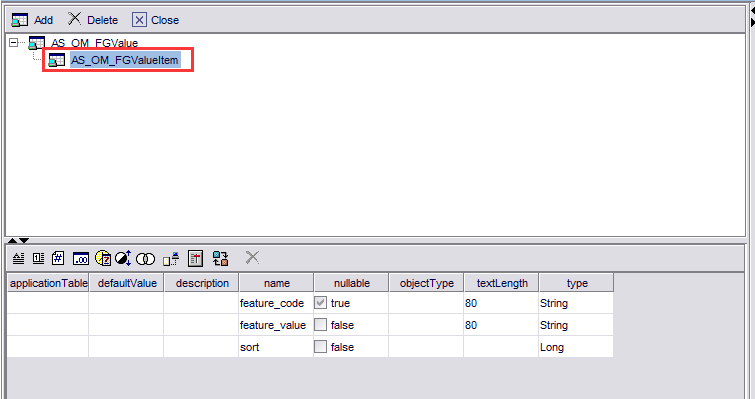
1. 创建AT表对象

1.主表

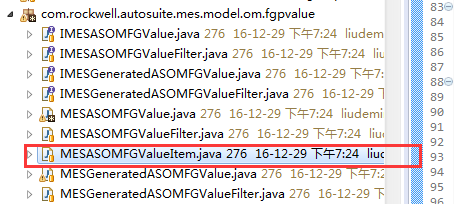


2.子表



1. 创建对象类
2. 打开Form（Tool\_ATDefinitionAccessClassGenerator），生成主表对象类；（参考创建AT表对象）；
3. 打开Form（Tool\_ATDefinitionAccessClassGenerator），生成主表对象类；（参考创建AT表对象）；

备注：拷贝子表的主体类（MESASOMFGValueItem）到主表类所在包内

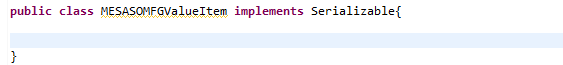


1. 修改子表对象类
2. 删除子表MESASOMFGValueItem的继承类（MESGeneratedASOMFGValueItem）和接口（IMESASOMFGValueItem）；
3. 增加子表MESASOMFGValueItem的继承接口（Serializable）

|  |
| --- |
| **public** **class** MESASOMFGValueItem **implements** Serializable |

备注：需要生成***serialVersionUID***

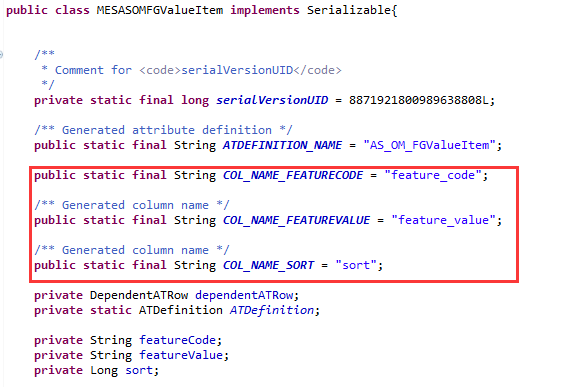
1. 删除子表主体类的所有构造方法，结果如下：



1. 从MESGeneratedASOMFGValueItemFilter类中复制AT表定义名字；

|  |
| --- |
| public static final String ATDEFINITION\_NAME = "AS\_XX\_XXXXXX\_XXX\_Item"; |

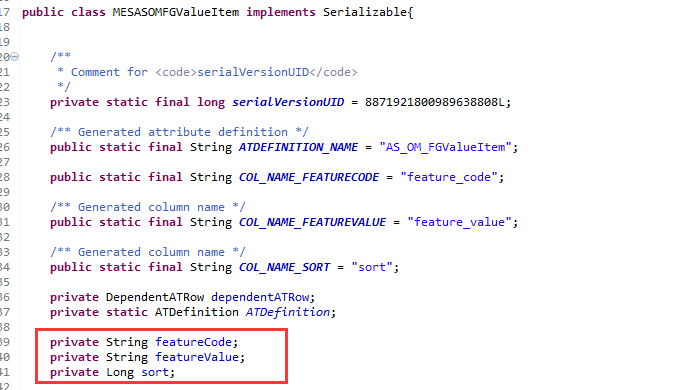
1. 从MESGeneratedASOMFGValueItem类中复制所有属性的字段名



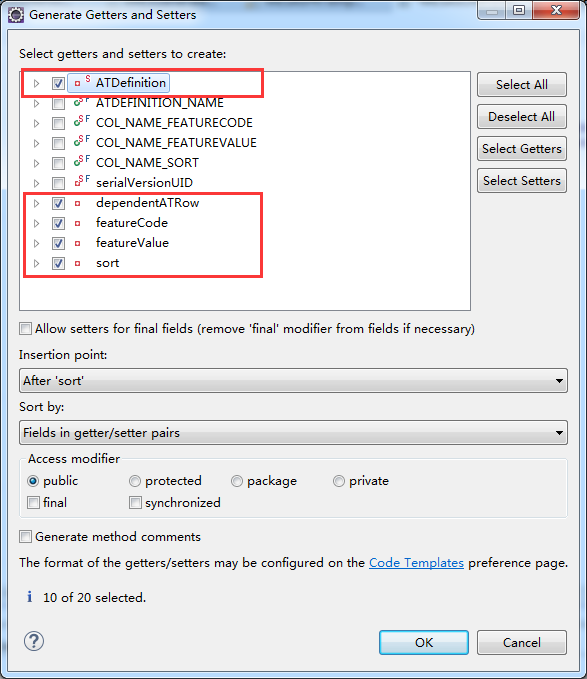
1. 定义DependentATRow，ATDefinition变量

|  |
| --- |
| **private** DependentATRow dependentATRow;  **private** **static** ATDefinition *ATDefinition*; |

1. 定义子表类的所有属性的变量

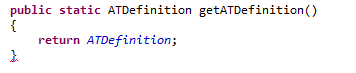


1. 针对6和7声明的变量生成set和get方法；



1. 修改public static ATDefinition getATDefinition()

修改前：



修改后：

|  |
| --- |
| public static void getATDefinition()  {  if (ATDefinition == null)  {  ATDefinition = AsContext.getFunctions().getATDefinition(  ATDEFINITION\_NAME);  }  Return ATDefinition;  } |

1. 修改public static void setATDefinition(ATDefinition aTDefinition)方法

修改前：



修改后：

|  |
| --- |
| public static void setATDefinition()  {  if (getATDefinition() == null)  {  ATDefinition = AsContext.getFunctions().getATDefinition(  ATDEFINITION\_NAME);  }  } |

1. 增加fromAtRow方法，方法如下

|  |
| --- |
| public void fromAtRow()  {  if(dependentATRow != null)  {  this.setFeatureCode((String) dependentATRow.getValue(COL\_NAME\_FEATURECODE));  this.setFeatureValue((String) dependentATRow.getValue(COL\_NAME\_FEATUREVALUE));  this.setSort((Long) dependentATRow.getValue(COL\_NAME\_SORT));  }  } |

备注：如果是对象，先从dependentATRow.getValue获取主键，再转成对象；

this.setProfile(new MESASSMProfile ((Long)dependentATRow.getValue(COL\_NAME\_PROFILE)));

1. 增加toAtRow方法，方法如下

|  |
| --- |
| public void toAtRow(ParentATRow parentatrow) throws DatasweepException  {  if (dependentATRow == null)  {  dependentATRow = parentatrow.createDependentATRow\_(  getATDefinition(), UUID.randomUUID().toString());  }  dependentATRow.setValue(COL\_NAME\_FEATURECODE, getFeatureCode());  dependentATRow.setValue(COL\_NAME\_FEATUREVALUE, getFeatureValue());  dependentATRow.setValue(COL\_NAME\_SORT, getSort());  } |

备注：如果是对象，需要获取对象的主键再进行赋值；

dependentATRow.setValue(COL\_NAME\_PROFILE, getProfile().getKey());

1. 增加initialize方法，方法如下

|  |
| --- |
| **public** **void** initialize()  {  *setATDefinition*();  dependentATRow = **null**;  } |

1. 增加构造MESASOMFGValueItem()方法，方法如下

|  |
| --- |
| **public** MESASOMFGValueItem()  {  initialize();  } |

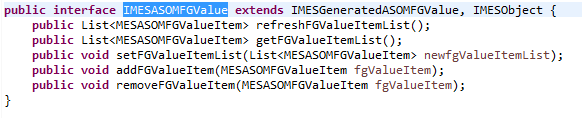
1. 增加MESASOMFGValueItem(DependentATRow dependentATRow)方法，方法如下

|  |
| --- |
| **public** MESASOMFGValueItem(DependentATRow dependentATRow)  {  initialize();  **this**.dependentATRow = dependentATRow;  fromAtRow();  } |

1. 增加getKey()方法，方法如下

|  |
| --- |
| public Long getKey()  {  Long key = -1l;  if(dependentATRow != null)  {  key = dependentATRow.getKey();  }  return key;  } |

1. 修改主表对象类
2. 在IMESASOMFGValue类中增加以下5个接口方法



|  |
| --- |
| **public interface IMESASOMFGValue extends IMESGeneratedASOMFGValue, IMESObject {**  **public List<MESASOMFGValueItem> refreshFGValueItemList();**  **public List<MESASOMFGValueItem> getFGValueItemList();**  **public void setFGValueItemList(List<MESASOMFGValueItem> newfgValueItemList);**  **public void addFGValueItem(MESASOMFGValueItem fgValueItem);**  **public void removeFGValueItem(MESASOMFGValueItem fgValueItem);**  **}** |

1. 在主表类（MESASOMFeatureValue）增加变量用于表示子表对象是否未加载

|  |
| --- |
| private Boolean isNotLoadItemList = true; |

1. 在主表类（MESASOMFeatureValue）增加成员变量用于表示子表对象集合

|  |
| --- |
| private List<MESASOMFGValueItem> fgValueItemList = new ArrayList<MESASOMFGValueItem>(); |

备注：类型为子表对象的类名；

1. 在MESASOMFGValue类中重写IMESASOMFGValue类中的**refreshFGValueItemList**方法，方法实现如下

|  |
| --- |
| **public List<MESASOMFGValueItem> refreshFGValueItemList()**  **{**  **ParentATRow parentATRow = (ParentATRow) this.getATRow();**  **if(fgValueItemList != null)**  **{**  **fgValueItemList.clear();**  **}**  **Vector dependATRows = parentATRow.getDependentATRows(MESASOMFGValueItem.getATDefinition());**  **for(Object atrowObj : dependATRows)**  **{**  **DependentATRow atRow = (DependentATRow)atrowObj;**  **MESASOMFGValueItem fgValueItem = new MESASOMFGValueItem(atRow);**  **fgValueItemList.add(fgValueItem);**  **}**  **isNotLoadItemList = false;**  **return fgValueItemList;**  **}** |

1. 在MESASOMFGValue类中重写IMESASOMFGValue类中的getFGValueItemList方法，方法实现如下

|  |
| --- |
| public List<MESASOMFGValueItem> getFGValueItemList()  {  if(isNotLoadItemList)  {  refreshFGValueItemList();  }  return fgValueItemList;  } |

1. 在MESASOMFGValue类中重写IMESASOMFGValue类中的setFGValueItemList方法，方法实现如下

|  |
| --- |
| public void setFGValueItemList(List<MESASOMFGValueItem> newfgValueItemList) throws DatasweepException  {  **if**(isNotLoadItemList)  {  refreshFGValueItemList();  }  if(fgValueItemList != newfgValueItemList)  fgValueItemList.clear();    **if**(newfgValueItemList != **null**)  {  **for**(MESASOMFGValueItem fgValueItem:newfgValueItemList)  {  addFGValueItem(fgValueItem);  }  }  }  } |

1. 在MESASOMFGValue类中重写IMESASOMFGValue类中的addFGValueItem方法，方法实现如下

|  |
| --- |
| public void addFGValueItem(MESASOMFGValueItem fgValueItem) throws DatasweepException  {  if(isNotLoadItemList)  {  refreshFGValueItemList();  }  fgValueItemList.add(fgValueItem);  } |

1. 在MESASOMFGValue类中重写IMESASOMFGValue类中的removeFGValueItem方法，方法实现如下

|  |
| --- |
| public void removeFGValueItem(MESASOMFGValueItem fgValueItem)  {  if(isNotLoadItemList)  {  refreshFGValueItemList();  }  fgValueItemList.remove(fgValueItem);  } |

1. 在MESASOMFGValue类中重写MESGeneratedASOMFGValue类中方法

void Save(Time, String, AccessPrivilege)

|  |
| --- |
| **public** **void** Save(Time time, String comment, AccessPrivilege accessPrivilege)  **throws** DatasweepException  {  **if**(isNotLoadItemList)  {  refreshFGValueItemList();  }  Response resp = **new** Response();  **if**(**this**.getFeaturegroup() == **null**)  {  resp.addError(**new** Error(***MESSAGE\_PACK\_NAME***, ***MESSAGE\_ID\_FIELD\_NOT\_NULL***,  **new** String[] { ***COL\_NAME\_FEATUREGROUP*** }, PCContext.*getServerImpl*()));  **throw** **new** DatasweepException(resp);  }    **if** (StringUtilsEx.*isBlank*(**this**.getValue()))  {  resp.addError(**new** Error(***MESSAGE\_PACK\_NAME***, ***MESSAGE\_ID\_FIELD\_NOT\_NULL***,  **new** String[] { ***COL\_NAME\_VALUE*** }, PCContext.*getServerImpl*()));  **throw** **new** DatasweepException(resp);  }    **for** (MESASOMFGValueItem fgValueItem : fgValueItemList)  {  **if** (StringUtilsEx.*isBlank*(fgValueItem.getFeatureCode()))  {  resp.addError(**new** Error(***MESSAGE\_PACK\_NAME***, ***MESSAGE\_ID\_FIELD\_NOT\_NULL***,  **new** String[] { MESASOMFGValueItem.***COL\_NAME\_FEATURECODE*** }, PCContext.*getServerImpl*()));  **throw** **new** DatasweepException(resp);  }    **if** (StringUtilsEx.*isBlank*(fgValueItem.getFeatureValue()))  {  resp.addError(**new** Error(***MESSAGE\_PACK\_NAME***, ***MESSAGE\_ID\_FIELD\_NOT\_NULL***,  **new** String[] { MESASOMFGValueItem.***COL\_NAME\_FEATUREVALUE*** }, PCContext.*getServerImpl*()));  **throw** **new** DatasweepException(resp);  }    **if** (fgValueItem.getSort() == **null**)  {  resp.addError(**new** Error(***MESSAGE\_PACK\_NAME***, ***MESSAGE\_ID\_FIELD\_NOT\_NULL***,  **new** String[] { MESASOMFGValueItem.***COL\_NAME\_SORT*** }, PCContext.*getServerImpl*()));  **throw** **new** DatasweepException(resp);  }  }    ParentATRow parentATRow = (ParentATRow) **this**.getATRow();  Vector<DependentATRow> dependATRows = parentATRow.getDependentATRows(**MESASOMFGValueItem.getATDefinition()**);  **for**(DependentATRow dependentatrow : dependATRows)  {  Boolean findFlag = **false**;  Long rowKey = dependentatrow.getKey();  **for** (MESASOMFGValueItem fgValueItem : fgValueItemList)  {  DependentATRow atRow = fgValueItem.getDependentATRow();  **if**(atRow != **null**)  {  Long objKey =atRow.getKey();  **if**(rowKey.equals(objKey))  {  findFlag = **true**;  }  }  }  **if**(findFlag==**false**)  {  parentATRow.removeDependentATRow(dependentatrow);  }  }    **for** (MESASOMFGValueItem fgValueItem : fgValueItemList)  {  fgValueItem.toAtRow( (ParentATRow) **this**.getATRow());  }  **super**.Save(  time, comment, accessPrivilege);  } |

1. 在MESASOMFGValue类中重写MESGeneratedASOMFGValue类中方法

public boolean refresh()

|  |
| --- |
| @Override  **public** **boolean** refresh()  {  refreshFGValueItemList();  **return** **super**.refresh();  } |