



# Основы программирования в Windchill

PTC Global Services



Григорий Чернобыль  
Декабрь 2012

# Содержание

- > Поиск по атрибутам
- > Получение последних итераций последних ревизий
- > Работа с атрибутами (чтение и запись значений)
- > Связи и работа с ними
- > Создание связей



# Поиск по атрибутам

## //1. Включение в поиск идентификатора типа объекта и параметра поиска

```
wt.query.CompositeWhereExpression cCompositeWhereExpression = null;
```

```
wt.query.QuerySpec cQuerySpec = null;
```

```
String cName = <?>;
```

```
cCompositeWhereExpression = new
```

```
    wt.query.CompositeWhereExpression(wt.query.LogicalOperator.AND);
```

```
cCompositeWhereExpression.append(new wt.query.SearchCondition(wt.part.WTPart.class,  
    wt.part.WTPart.NAME, wt.query.SearchCondition.EQUAL, cName), new int[] { 0 });
```

```
cCompositeWhereExpression.append(new wt.query.SearchCondition(wt.part.WTPart.class,  
    "typeDefinitionReference.key.branchId", wt.query.SearchCondition.EQUAL,  
    BRANCHIDITERATIONINFO), new int[] { 0 });
```

```
cQuerySpec.appendWhere(cCompositeWhereExpression, new int[] {0,0});
```

# Поиск по атрибутам. 2 из 4

## //2. Поиск идентификатора атрибута

```
long IBAStringDefinitionId = 0;
```

```
String AttributId = <?>;
```

```
wt.query.QuerySpec cQuerySpecEx = null;
```

```
wt.query.CompositeWhereExpression cCompositeWhereExpressionEx = null;
```

```
wt.fc.QueryResult cQueryResultEx = null;
```

```
wt.iba.definition.StringDefinition cStringDefinition = null;
```

```
cQuerySpecEx = new wt.query.QuerySpec(wt.iba.definition.StringDefinition.class);
```

```
cCompositeWhereExpressionEx = new
```

```
    wt.query.CompositeWhereExpression(wt.query.LogicalOperator.AND);
```

```
cCompositeWhereExpressionEx.append(new
```

```
    wt.query.SearchCondition(wt.iba.definition.StringDefinition.class,
```

```
    wt.iba.definition.StringDefinition.NAME, wt.query.SearchCondition.EQUAL, AttributId),
```

```
    new int[] { 0 });
```

```
cQuerySpecEx.appendWhere(cCompositeWhereExpressionEx, new int[] {0});
```

```
cQueryResultEx = wt.fc.PersistenceHelper.manager.find( (wt.pds.StatementSpec) cQuerySpecEx);
```

```
while(cQueryResultEx.hasMoreElements()) {
```

```
    cStringDefinition = (wt.iba.definition.StringDefinition) cQueryResultEx.nextElement();
```

```
}
```

```
IBAStringDefinitionId = cStringDefinition.getPersistInfo().getObjectIdentifier().getId();
```

# Поиск по атрибутам. 3 из 4

## //3. Включение идентификатора атрибута в поиск

```
//>cQuerySpec
IBAStringValue = <?>; //значение атрибута
int i = 0;
wt.query.SearchCondition cSearchCondition = null;
wt.query.SearchCondition cSearchConditionEx = null;
i = cQuerySpec.appendClassList(wt.iba.value.StringValue.class, false);
int ai[] = { 0, i };

cSearchCondition = new wt.query.SearchCondition(wt.iba.value.StringValue.class,
    "definitionReference.key.id", wt.query.SearchCondition.EQUAL, IBAStringDefinitionId);
cSearchConditionEx = new wt.query.SearchCondition(wt.part.WTPart.class,
    "thePersistInfo.theObjectIdentifier.id", wt.iba.value.StringValue.class,
    "theIBAHolderReference.key.id");

cQuerySpec.appendAnd();
cQuerySpec.appendWhere(cSearchConditionEx, ai);
cQuerySpec.appendAnd();
cQuerySpec.appendOpenParen();
cQuerySpec.appendWhere(cSearchCondition, i);
cQuerySpec.appendAnd();
cQuerySpec.appendOpenParen();

wt.query.WhereExpression awhereexpression[] = new wt.query.WhereExpression[1];
awhereexpression[0] = new wt.query.SearchCondition(wt.iba.value.StringValue.class, "value2", "=",
    IBAStringValue);
cQuerySpec.appendWhere(awhereexpression[0], i);
cQuerySpec.appendCloseParen();
cQuerySpec.appendCloseParen();
```

# Поиск по атрибутам. 4 из 4

## //4. Поиск

smp.pkb.pkgproj.AttributHelper2653

```
wt.part.WTPart cWTPart = null;
```

```
cQueryResult = wt.fc.PersistenceHelper.manager.find( (wt.pds.StatementSpec) cQuerySpec);
```

```
while(cQueryResult.hasMoreElements()){  
Object javaTempObj = cQueryResult.nextElement();
```

```
Out.println(javaTempObj.toString());
```

```
/*
```

```
    cWTPart = null;
```

**//если атрибут успешно добавился**

```
    cWTPart = (wt.part.WTPart) ((wt.fc.Persistable[])cQueryResult.nextElement())[0];
```

**//если атрибут не добавился (к примеру отсутствует)**

```
    cWTPart = (wt.part.WTPart) cQueryResult.nextElement();
```

```
*/
```

```
}
```

# Получение последних итераций последних ревизий

## //1. Получение последней итерации

```
Object cObject = null;
QueryResult cQueryResultLT = null;

try {
    cQueryResultLT = wt.vc.VersionControlHelper.service.allIterationsOf(cMastered, false);
} catch (PersistenceException cPersistenceException) {
    //код ошибки
} catch (WTException cWTException) {
    //код ошибки
}

while( cQueryResultLT.hasMoreElements()){
    cObject = null;
    cObject = cQueryResultLT.nextElement();
    break; //получение последней итерации
}

if ( cObject == null ){
    //код ошибки
}
```

## //2. Последняя итерация может быть в не последней ревизии

```
try{
    long id = 0;
    Object cObjectEx = null;

    cQueryResultLT = null;
    cQueryResultLT = wt_vc.config.ConfigHelper.service.filteredIterationsOf(cMastered,
        wt_vc.config.ConfigHelper.service.getConfigSpecFor( (wt_vc.Iterated) cObject));

    while( cQueryResultLT.hasMoreElements()){
        cObjectEx = null;
        cObjectEx = cQueryResultLT.nextElement();

        if ( PersistenceHelper.getObjectIdentifier( (Persistable) cObjectEx).getId() > id){
            id = PersistenceHelper.getObjectIdentifier( (Persistable) cObjectEx).getId();
            cObject = cObjectEx;
            break; //
        }
    }

}catch(WTException cWTException){
    //код ошибки
}
```

# Работа с атрибутами (чтение и запись значений)

# Получение значения атрибута. 1 из 2

## //1. Получение значения атрибута без привязки к его типу

```
wt.iba.value.litevalue.AbstractValueView cAbstractValueView = null;
```

```
wt.iba.value.IBAHolder cIBAHolder = null;
```

```
wt.iba.value.DefaultAttributeContainer cDefaultAttributeContainer = null;
```

```
wt.iba.value.litevalue.AbstractValueView[] cAbstractValueViewArray = null;
```

```
cIBAHolder = (wt.iba.value.IBAHolder) cPersistable;
```

```
cIBAHolder = (wt.iba.value.IBAHolder)
```

```
    wt.iba.value.service.IBAValueHelper.service.refreshAttributeContainer(cIBAHolder, null, null,  
    null);
```

```
cDefaultAttributeContainer = (wt.iba.value.DefaultAttributeContainer)
```

```
    cIBAHolder.getAttributeContainer();
```

```
cAbstractValueViewArray = null;
```

```
cAbstractValueViewArray = cDefaultAttributeContainer.getAttributeValues();
```

```
for (int i = 0; i < cAbstractValueViewArray.length; i++){
```

```
    out.println(cAbstractValueViewArray[i].getDefinition().getName());
```

```
}
```

```
//if ( cAbstractValueViewArray[i].getDefinition().getName().equalsIgnoreCase(AttributeID) ){
```

```
    //cAbstractValueView = cAbstractValueViewArray[i];
```

```
    //break;
```

# Получение значения атрибута. 1 из 2

## //2. Получение значения

```
String AttributeValue = "";
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.StringValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.StringValueDefaultView)cAbstractValueView).getValueAsString();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.FloatValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.FloatValueDefaultView)cAbstractValueView).getValueAsString();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.BooleanValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.BooleanValueDefaultView)cAbstractValueView).getValueAsString();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.IntegerValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.IntegerValueDefaultView)cAbstractValueView).getValueAsString()
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.RatioValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.RatioValueDefaultView)cAbstractValueView).getValueAsString();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.ReferenceValueDefaultView)  
    AttributeValue =  
        ((wt.iba.value.litevalue.ReferenceValueDefaultView)cAbstractValueView).getLiteIBAReferenceable().getReferenceID();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.TimestampValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.TimestampValueDefaultView)cAbstractValueView).getValueAsString();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.UnitValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.UnitValueDefaultView)cAbstractValueView).getValueAsString();
```

```
if ( cAbstractValueView instanceof wt.iba.value.litevalue.URLValueDefaultView)  
    AttributeValue = ((wt.iba.value.litevalue.URLValueDefaultView)cAbstractValueView).getValueAsString();
```

# Запись значения атрибута

```
// Формат NewAttributeValue --> new java.lang.String(value)
// Внимание: для изменения значения атрибута, объект должен быть взят на изменение

wt.iba.value.DefaultAttributeContainer cDefaultAttributeContainer = null;
wt.iba.definition.litedefinition.AttributeDefDefaultView cAttributeDefDefaultView = null;
wt.iba.value.IBAHolder cIBAHolder = null;
wt.iba.value.litevalue.AbstractValueView cAbstractValueViewArray[] = null;

cIBAHolder =
    (wt.iba.value.IBAHolder)wt.iba.value.service.IBAValueHelper.service.refreshAttributeContainer((wt.iba.value.IBAHolder)
cPersistable, null, null, null);

cDefaultAttributeContainer = (wt.iba.value.DefaultAttributeContainer) cIBAHolder.getAttributeContainer();

cAttributeDefDefaultView = wt.iba.definition.service.IBADefinitionHelper.service.getAttributeDefDefaultViewByPath(AttributeID);

cAbstractValueViewArray = cDefaultAttributeContainer.getAttributeValues(cAttributeDefDefaultView);

if ( NewAttributeValue instanceof java.lang.String ){
    if(cAbstractValueViewArray == null || cAbstractValueViewArray.length == 0){
        cDefaultAttributeContainer.addAttributeValue(new
            wt.iba.value.litevalue.StringValueDefaultView((wt.iba.definition.litedefinition.StringDefView)cAttributeDefDefaultView,
            (java.lang.String) NewAttributeValue ));
    }else{
        wt.iba.value.litevalue.StringValueDefaultView cStringValueDefaultView =
            (wt.iba.value.litevalue.StringValueDefaultView)cAbstractValueViewArray[0];
        cStringValueDefaultView.setValue( (java.lang.String) NewAttributeValue );
        cDefaultAttributeContainer.updateAttributeValue(cStringValueDefaultView);
    }
}
```

# Связи и работа с ними

# Получение входящих объектов (потомков)

```
QueryResult cQueryResult = null;  
WTPartUsageLink cWTPartUsageLink = null;  
WTPartMaster cWTPartChildMaster = null;  
  
try {  
  
    cQueryResult = wt.part.WTPartHelper.service.getUsesWTPartMasters(cWTPart);  
  
} catch (WTException cWTException) {  
    //код ошибки  
}  
  
while( cQueryResult.hasMoreElements()){  
  
    cWTPartUsageLink = null;  
    cWTPartUsageLink = (wt.part.WTPartUsageLink) cQueryResult.nextElement();  
  
    cWTPartChildMaster = null;  
    cWTPartChildMaster = cWTPartUsageLink.getUses();  
  
}
```

# Получение объектов (родителей)

```
QueryResult cQueryResult = null;  
WTPart cWTPartParent = null;  
  
try {  
    cQueryResult = wt.part.WTPartHelper.service.getUsedByWTParts(cWTPartMaster);  
} catch (WTEexception cWTEexception) {  
    //код ошибки  
}  
  
while( cQueryResult.hasMoreElements()){  
    cWTPartParent = null;  
    cWTPartParent = (wt.part.WTPart) cQueryResult.nextElement();  
}
```

# Создание связей

# Создание нового объекта

```
wt.part.WTPart newWTPart = null;  
  
newWTPart = wt.part.WTPart.newWTPart(newNumber, newName);  
newWTPart.setOrganization(cWTContainer.getOrganization());  
newWTPart.setContainer(cWTContainer);  
  
newWTPart.setView(wt_vc.views.ViewReference.newViewReference(wt_vc.views.ViewHelper.ser  
vice.getAllRootViews()[0]));  
  
newWTPart = (wt.part.WTPart)wt_fc.PersistenceHelper.manager.store(newWTPart);
```

# Создание новой связи между объектами

## //Создание связи между Частями

```
wt.part.WTPartUsageLink cWTPartUsageLink = null;  
wt.part.Quantity cQuantity = null;
```

```
cQuantity = wt.part.Quantity.newQuantity(1.0, wt.part.QuantityUnit.getQuantityUnitDefault());
```

```
cWTPartUsageLink = WTPartUsageLink.newWTPartUsageLink(WTParentPart, (WTPartMaster)  
WTChildPart.getMaster());  
cWTPartUsageLink.setQuantity(cQuantity);
```

//Внимание: родительская Часть должна быть взята на изменение, до начала выполнения  
следующей операции

```
cWTPartUsageLink = (WTPartUsageLink)  
wt.fc.PersistenceHelper.manager.store(cWTPartUsageLink);
```

# Создание новой связи между объектами

## //Создание связи между Частью и Документом

```
wt.part.WTPartDescribeLink cWTPartDescribeLink = null;
```

```
cWTPartDescribeLink = wt.part.WTPartDescribeLink.newWTPartDescribeLink(cWTPart,  
cWTDocument);
```

```
cWTPartDescribeLink = (wt.part.WTPartDescribeLink)  
wt.fc.PersistenceHelper.manager.save(cWTPartDescribeLink);
```

# Вопросы?

PTC®



Григорий Чернобыль

Руководитель PTC Global Services

[gchernobyl@ptc.com](mailto:gchernobyl@ptc.com)

(+7)985-922-84-07

Спасибо