

Основы программирования в Windchill

PTC Global Services



Григорий Чернобыль
Декабрь 2012

- > Программирование поисковых запросов
- > Обработка ошибок (лучшие практики)
- > Связи и работа с ними
- > Получение последних итераций последних ревизий
- > Создание объектов и связей
- > Работа с атрибутами (чтение и запись значений)
- > Создание сервисов



Программирование поисковых запросов

```
String FieldName = wt.part.WTPart.NUMBER; //, wt.doc.WTDocument.NAME,...  
String FieldValue = „%“;  
Class WTCClass = wt.part.WTPart.class; //, wt.doc.WTDocument.class,...
```

```
wt.query.QuerySpec cQuerySpec = null;  
wt.fc.QueryResult cQueryResult = null;  
wt.query.CompositeWhereExpression cCompositeWhereExpression = null;
```

```
try{  
    cCompositeWhereExpression = new  
        wt.query.CompositeWhereExpression(wt.query.LogicalOperator.AND);  
  
    cQuerySpec = new wt.query.QuerySpec(wt.part.WTPart.class);  
    cCompositeWhereExpression.append(new wt.query.SearchCondition(WTCClass,  
        wt.part.WTPart.NUMBER, wt.query.SearchCondition.EQUAL, FieldValue), new int[] { 0 });  
    cQuerySpec.appendWhere(cCompositeWhereExpression, new int[] {0});  
  
    cQueryResult = new wt.fc.QueryResult();  
    cQueryResult = wt.fc.PersistenceHelper.manager.find( (wt.pds.StatementSpec) cQuerySpec);  
}catch(Exception cException){  
    //  
}
```

```
while ( cQueryResult.hasMoreElements() ) {  
  
    wt.fc.Persistable cPersistable = cQueryResult.nextElement();  
  
    out.println(„<br>” + cPersistable.toString());  
  
}
```

```
cQuerySpec = new wt.query.QuerySpec(WTClass);  
  
cCompositeWhereExpression.append(new wt.query.SearchCondition(WTClass, FieldName1,  
    wt.query.SearchCondition.EQUAL, FieldValue1), new int[] { 0 });  
  
cCompositeWhereExpression.append(new wt.query.SearchCondition(WTClass, FieldName2,  
    wt.query.SearchCondition.EQUAL, FieldValue2), new int[] { 0 });  
  
cQuerySpec.appendWhere(cCompositeWhereExpression, new int[] {0,0});
```

//1. Поиск идентификатора типа объекта

```
String WCTypelIdentifier = <?>; // CustomType
```

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

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

```
long BRANCHIDITERATIONINFO = 0;
```

```
Object cObject = null;
```

```
cQuerySpec = new
```

```
wt.query.QuerySpec(com.ptc.core.meta.type.mgmt.server.impl.WTTypeDefinition.class)
```

```
;
```

```
cQuerySpec.appendWhere(new
```

```
wt.query.SearchCondition(com.ptc.core.meta.type.mgmt.server.impl.WTTypeDefinition.
```

```
class, com.ptc.core.meta.type.mgmt.server.impl.WTTypeDefinition.NAME,
```

```
wt.query.SearchCondition.EQUAL, WCTypelIdentifier ,false), new int[] { 0 }));
```

```
cQueryResult = new wt.fc.QueryResult();
```

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

```
while ( cQueryResult.hasMoreElements() )
```

```
BRANCHIDITERATIONINFO =
```

```
((WTTypeDefinition)cQueryResult.nextElement()).getBranchIdentifier();
```

//2. Поиск объекта заданного типа

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

```
cQuerySpec = new wt.query.QuerySpec(wt.part.WTPart.class);  
CompositeWhereExpression where = new  
    CompositeWhereExpression(wt.query.LogicalOperator.AND);
```

```
where.append(new wt.query.SearchCondition(wt.part.WTPart.class, wt.part.WTPart.NAME,  
    wt.query.SearchCondition.LIKE, „%”), new int[] { 0 });  
where.append(new wt.query.SearchCondition(wt.part.WTPart.class,  
    "typeDefinitionReference.key.branchId", wt.query.SearchCondition.EQUAL,  
    BRANCHIDITERATIONINFO), new int[] { 0 });
```

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

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

```
while ( cQueryResult.hasMoreElements() ) {  
    wt.part.WTPart cWTPart = (wt.part.WTPart) cQueryResult.nextElement();  
}
```


//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});
```

//2. Поиск идентификатора атрибута

```
long IBAStrngDefinitionId = 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();
```

```
}
```

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

//3. Включение идентификатора атрибута в поиск

```
IBASStringValue = <?>; //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, IBASStringDefinitionId);
```

```
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", "=",  
    IBASStringValue);
```

```
cQuerySpec.appendWhere(awhereexpression[0], i);
```

```
cQuerySpec.appendCloseParen();
```

```
cQuerySpec.appendCloseParen();
```

//4. Поиск

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

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

```
while(cQueryResult.hasMoreElements()){
```

```
    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();
```

```
}
```

Обработка ошибок (лучшие практики)

```

= (wt.part.WTPart) mohawk.acquisition.BasicHelper.GetPersistentByObjectIdentifier(oid);
= mohawk.mpml.loader.RawMaterials.createNewRawMaterial(
wt.type.TypedUtilityServiceHelper.service.getTypeDefinitionReference(mohawk.config.ConfigHelper.GetConfigValue("mpml" + mohawk.config.Con
(wt.inf.library.WTLibrary) cWTPart.getCont
""),
cWTPart.getName(),
cWTPart.getDefaultUnit().toString(),
cWTPart.getState().getState(),
Parameters,
2 //skip
);

```

```
try{  
    //код  
}catch(wt.util.WTException cWTException){  
    String error_desc = cWTException.getMessage(); //описание ошибки  
    //код если ошибка  
}  
//-----  
try{  
    //код  
}catch(Exception cException){  
    String error_desc = cException.getMessage(); //описание ошибки  
    //код если ошибка  
}
```

```
String error_desc = "";
```

```
try{
```

```
//код
```

```
}catch(Exception cException){
```

```
    error_desc = "";
```

```
    try{
```

```
        error_desc = cException.getMessage(); //описание ошибки
```

```
    }catch(Exception ExceptionOnException){
```

```
        //
```

```
    }
```

```
    //код если ошибка
```

```
}
```


Связи и работа с ними

```
QueryResult cQueryResult = null;
WTPartUsageLink cWTPartUsageLink = null;
WTPartMaster cWTPartChildMaster = null;

try {

    cQueryResult = wt.part.WTPartHelper.service.getUsesWTPartMasters(cWTPart);

} catch (WTEException cWTEException) {
    //код ошибки
}

while( cQueryResult.hasMoreElements()){

    cWTPartUsageLink = null;
    cWTPartUsageLink = (wt.part.WTPartUsageLink) cQueryResult.nextElement();

    cWTPartChildMaster = null;
    cWTPartChildMaster = cWTPartUsageLink.getUses();

}
```

```
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();  
}
```

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
}
```

Создание объектов и связей

```
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 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 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);
```

Работа с атрибутами

//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++)
    if ( cAbstractValueViewArray[i].getDefinition().getName().equalsIgnoreCase(AttributeID) ){
        cAbstractValueView = cAbstractValueViewArray[i];
        break;
    }
```

//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).getLiteIBAResourceable().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);
```

```
    }
```

```
}
```

Создание сервисов

//1. Объявление интерфейса для сервиса

```
package mohawk.mpml.service;
```

```
public interface ProcessPlanInterface {
```

```
//-----
```

```
/*
```

```
 * MPMPProcessPlan creation service listener
```

```
 *
```

```
 */
```

```
//-----
```

```
}
```


//2. Объявление события для сервиса

```
package mohawk.mpml.service;
import java.io.Serializable;
import wt.events.KeyedEvent;
import wt.fc.PersistenceManagerEvent;
import wt.services.*;
import wt.util.WTException;
import mohawk.mpml.service.ProcessPlanInterface;
public class ProcessPlanListener extends StandardManager implements ProcessPlanInterface, Serializable {
    private static final long serialVersionUID = 1L;
    private static final String CLASSNAME = mohawk.mpml.service.ProcessPlanListener.class.getName();
    public ProcessPlanListener(){
    }
    public String getConceptualClassname(){ return CLASSNAME; }
    public static ProcessPlanListener newProcessPlanListener() throws WTException{
        ProcessPlanListener instance = new ProcessPlanListener(); instance.initialize(); return instance;
    }
    protected void performStartupProcess() throws ManagerException {
        getManagerService().addEventListener(new ServiceEventListenerAdapter(getConceptualClassname()) {
            public void notifyVetoableEvent( Object event ) throws WTException {
                if(!(event instanceof KeyedEvent))
                    return;
                KeyedEvent cKeyedEvent = (KeyedEvent) event;
                Object cObject = cKeyedEvent.getEventTarget();
                if ( cKeyedEvent.getEventType().equals(PersistenceManagerEvent.POST_STORE) )
                    if ( cObject instanceof MPMPartToProcessPlanLink )
                        performNewVersionEventProcess( cObject );
            }
        }, PersistenceManagerEvent.generateEventKey(PersistenceManagerEvent.POST_STORE));
        super.performStartupProcess();
    }
}
```

//3. Метод сервиса

```
protected void performNewVersionEventProcess(Object cObject) throws WtException {
```

```
    //код сервиса
```

```
}
```

//4. Установка конфигурационных параметров сервиса

Строка запуска из командной строки Windchill:

```
@xconfmanager -t codebase/wt.properties -s  
"wt.services.service.11001=mohawk.mpml.service.ProcessPlanInterface/mohawk.mpml.s  
ervice.ProcessPlanListener" -p
```


Григорий Чернобыль

Руководитель PTC Global Services

gchernobyl@ptc.com

[\(+7\)985-922-84-07](tel:+79859228407)

Спасибо