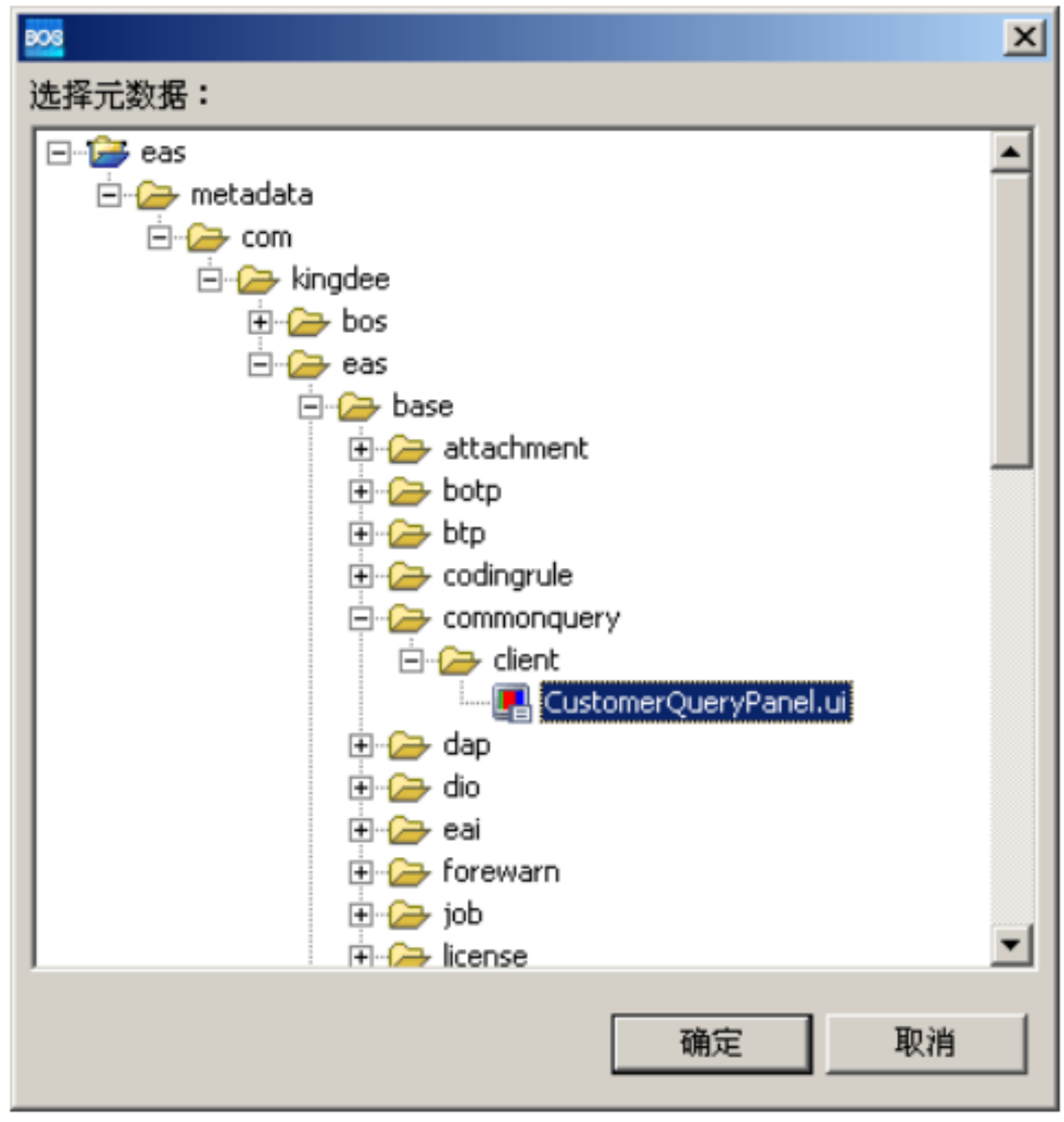
客户端常用代码

[redacted]

tblMain 作为框架为 list 列表界面中的 kdTable 绑定的变量， 通过它可以访问到列表上的每一行、每一列，可以做数据检查，控制，格式化，行合并等操作

1.1 初始化用户自己定义的查询过滤框（系统默认为每个定义的列表生成了通用过滤框，如果采用系统通用过滤框则不需要实现下面的方法）

dialog 可在 bos中继承 CustomerQueryPanel UI 新建用户自定义的过滤框



在 Bos 中发布该元数据后，可以实例化该类并引用

在生成的 dialog 类中继承方法 getFilterInfo()，并实现用户想要的过滤条件，应用框架会在取数时调用 dialog 类自动获取过滤条件

在 ListUI 类中重载方法 initCommonQueryDialog

```
protected CommonQueryDialog initCommonQueryDialog()
{
        dialog = super.initCommonQueryDialog();
        try
        {
            dialog.addUserPanel(getUsierPanel());
            dialog.setShowFilter(true);
            dialog.setShowSorter(true);
            dialog.setHeight(380);
            dialog.setWidth(500);
            dialog.setTitle( 客户自定义过滤框 ”);
      }
      catch (Exception e)
      {
            handUIException(e);
```

```
        }
        return dialog;
}
protected CustomerQueryPanel getUserPanel() throws Exception
{
                if (this.userPanel == null)
                        this.userPanel = new PurOrderQueryUI();
                userPanel.onLoad();
                return this.userPanel;
                }
```

如上，可以实现通过在通用过滤界面上增加自定义的过滤框实现对列表数据的过滤

1.2 设置列表界面的默认过滤条件，可用来过滤列表数据，也可以直接设置 mainQuery 的值实现过滤

```
protected EntityViewInfo getInitDefaultSolution()
{
        EntityViewInfo ev = new EntityViewInfo();
        FilterInfo filter = new FilterInfo();
        //给 filter 赋过滤条件项
        ev.setFilter(filter);
        return ev;
}
```

1.3 设置是否在调入列表界面之前先出过滤框

```
protected boolean initDefaultFilter()
{
        return true;
}
```

1.4 客户端对审核的操作

应用框架预定义了一个审核操作 actionAuditing

```
public void actionAuditing_actionPerformed(ActionEvent e) throws Exception
{
    //检查单据状态
    if (!checkStatus(tblMain, BillStatusEnum.SUBMITED))
    {
        MsgBox.showInfo(this,
        Day4Resource.getStrResource("isNotSubmitStatus"));
        //  状态不正确，终止处理
        SysUtil.abort();
    }
    IPurOrder purorder = (IPurOrder) getBizInterface();
    String[] billIdlist = getSelectedListId();

    //常用的弹出对话框方法
    int i = MsgBox.showConfirm2(this, "sureAudit");
    if (i == MsgBox.OK)
    {
        //审核操作
        purOrder.audit(new ObjectStringPK(bill));
        //完成其他业务逻控制
        //  刷新列表界面
        actionRefresh_actionPerformed(null);
    }
}
```

1.5 关联生成

```
public void actionCreateTo_actionPerformed(ActionEvent e) throws Exception {
    checkSelected();
    //  未审核单据不能关联生成
```

```
        if (!checkStatus(tblMain, BillStatusEnum.AUDITED)) {
            MsgBox.showInfo(this,       "billIsUnAudited");
            SysUtil.abort();
        }
        //有系统调用配置好的‘      botp’规则
        super.actionCreateTo_actionPerformed(e);
    }
```

1.6 动态调用 UI 界面，并在 UI 之间传递变量

```
    private void makePurOrderUI(PurOrderInfo srcBillInfo)
            throws EASBizException, UIException,
            BOSException, Exception {
        String    destBillEditUIClassName =    “com.kingdee…PurOrderEditUI   ”;
        Map map = new UIContext(this);
        map.put("srcBillID", srcBillInfo.getId().toString());
        map.put(UIContext.OWNER, this);
        map.put("srcBillBOSTypeString", destBillInfo.getBOSType());
        IUIWindow uiWindow = null ;
        // UIFactoryName.MODEL    为弹出模式
        uiWindow = UIFactory.createUIFactory(UIFactoryName.MODEL).
                create(destBillEditUIClassName, map, null,
                OprtState.ADDNEW);
        //可对创建的   ui 进行操作
        //((CoreBillEditUI).uiWindow.getUIObject()).
        //setMakeRelations(btpResult.getBOTRelationCollection());

        //开始展现   UI
        uiWindow.show();
    }
```

1.7 关于编码规则

```
// 是否存在编码规则
protected boolean isCodeRuleEnable(IObjectValue objValue)
        throws EASBizException, BOSException {
        String companyId = OrgInnerUtils.getCurCompany();
        ICodingRuleManager codeRuleMgr = null    ;
        codeRuleMgr     = CodingRuleManagerFactory.getRemoteInstance();
        return codeRuleMgr.isExist(objValue, companyId);
}
```

```
// 得到自动编码
protected String getAutoCode(IObjectValue objValue)
        throws EASBizException, BOSException {
        String companyId = OrgInnerUtils.getCurCompany();
        ICodingRuleManager codeRuleMgr = null    ;

        codeRuleMgr = CodingRuleManagerFactory.getRemoteInstance();
        if (codeRuleMgr.isUseIntermitNumber(objValue, companyId)) {
                return codeRuleMgr.readNumber(objValue, companyId);
        } else {
                return codeRuleMgr.getNumber(objValue, companyId);
        }

}
```

1.8 如何得到当前选中行的 id

```
// 得到当前选中行的 id
public String[] getSelectedListId() {
        checkSelected();
// SelectManager 是 kdtable 中行管理类
ArrayList blocks = tblMain.getSelectManager().getBlocks();
ArrayList idList = new ArrayList();
Iterator iter = blocks.iterator();
while (iter.hasNext()) {
        KDTSelectBlock block = (KDTSelectBlock) iter.next();
        int top = block.getTop();
        int bottom = block.getBottom();
```

```
for (int rowIndex = top; rowIndex <= bottom; rowIndex++)
{
        ICell cell = tblMain.getRow(rowIndex).
                                getCell(getKeyFieldName());

        if (!idList.contains(cell.getValue())) {
                idList.add(cell.getValue());
        }
}
        }
        String[] listId = null;
        if (idList != null && idList.size() > 0) {
                Iterator iterat = idList.iterator();
                listId = new String[idList.size()];
                int index = 0;
                while (iterat.hasNext()) {
                        listId[index] = (String) iterat.next();
                        index++;
                }
        }
        return listId;
}
```

1.9 关于数字精度的设置

```
private void changeShowData(int fistRow, int lastRow)
        throws EASBizException, BOSException
{
                for (int i = fistRow; i <= lastRow; i++)
                {
IRow row = tblMain.getRow(i);
//  根据币别设置精度
ICell cell = row.getCell(    columnName”);

//假定精度为   2
int precision = 2;
if (cell != null && cell.getValue() != null) {
        try {
                precision = Integer.parseInt(cell.getValue().toString());
```

```
                } catch (Exception e) {
                        precision = 2;
                }
        }
// 当该行的币别精度不能于原币精度时才更改精度
if (precision != basePrecision) {
                for (int j = 0; j <rowcount; j++) {

        tblMain.getColumn("billDate").getStyleAttributes().setNumberFormat(
                "%{yyyy-MM-dd}t");

        tblMain.getColumn("debitFor").getStyleAttributes().setNumberFormat(
                "%r{#,##0.00}f");

        tblMain.getColumn("debitFor").getStyleAttributes().setHorizontalAlign(
                HorizontalAlignment.RIGHT);

        tblMain.getColumn("endBalanceFor").getStyleAttributes()
                .setNumberFormat("%r{#,##0.00}f");

        tblMain.getColumn("endBalanceFor").getStyleAttributes()
                .setHorizontalAlign(HorizontalAlignment.RIGHT);
                        }
                }
        }
}
```

1.10 返回当前列表的主键

```
protected String getKeyFieldName() {
        return "id";
}
```

1.11 返回列表界面对应的编辑界面名称

```
protected String getEditUIName() {
        return    client.PurOrderEditUI.class.getName();
}
```

1.12 返回远程调用接口

```
protected com.kingdee.eas.framework.ICoreBase getBizInterface()
throws Exception {
        return    PurOrderFactory.getRemoteInstance();
}
```

2.1 获取菜单参数

getUIContext().get("UIClassParam")

2.2 客户端环境工具类

com.kingdee.eas.common.client.SysContext ； 静态存储用户当前登录信息, 登录的当前组织信息 (ContextUtil 在服务端使用)

获得当前用户的环境变量

Context ctx = SysContext.getSysContext() ；

获得当前财务组织，其他组织如：销售等也可从环境变量中取得

SysContext.getSysContext().getCurrentFIUnit() ；

SysContext.getSysContext().getCurrentUserInfo() 取得用户信息

2.3 获取公司行政组织

ICompanyOrgUnit comOrg =null;

comOrg = com.kingdee.eas.basedata.org.CompanyOrgUnitFactory.getRemoteInstance();

CompanyOrgUnitInfo comOu =ICompanyOrgUnit.getCompanyOrgUnitInfo(***) 根据不同需要调用相应方法。

2.4 获取当前公司本位币

companyOrgUnitInfo.getBaseCurrency()

2.5 期间工具

com.kingdee.eas.basedata.assistant.PeriodUtils 提供静态方法可获取期间，进行期间运算、比较等功能

2.6 获取资源文件

EASResource.getString("com.kingdee.eas.base.TestResource","periodName") 注 ：TestResource 为 bos 中对应的资源文件名

2.7 消息框

com.kingdee.eas.util.client.MsgBox.showWarning("***") ； 提供 message 对话框，只能在客户端使用， msgbox 中提供了多种方法

2.8 SysUtil.abort()

停止当前 UI 的所有操作，或终止服务端方法

2.9 com.kingdee.eas.base.uiframe.UIFactoryHelper

可指定 UI 名称动态生成 UI 由用户加到 java 的 panel 中，实现多页签

com.kingdee.eas.base.uiframe.UIFactory.initUIObject(XXXXX) 方法

可指定 UI 名称动态生成 UI 由框架自动载入

2.10　Tree 操作

com.kingdee.eas.basedata.org.client.tree.NewOrgTreeHelper　　提供了多种构造组织树的静态方法

com.kingdee.eas.basedata.org.client.f7.*

如:

```
KDTree tree.setModel(new DefaultTreeModel(nodes));

TreePath path = new TreePath(rootNode.getPath());

tree.expandPath(path);

tree.setSelectionPath(path);
```

2.11　获取编号

```
ICodingRuleManager iCodingRuleManager = null;

iCodingRuleManager　　=　　CodingRuleManagerFactory.getRemoteInstance();

//判断规则是否存在

iCodingRuleManager.isExist(billInfo, companyID)

//是否启用断号支持

iCodingRuleManager.isUseIntermitNumber(billInfo, companyID)

//读取当前最新编码

String sysNumber = iCodingRuleManager.readNumber(billInfo, companyID);

//没有启用断号支持功能，则获取编码规则产生的编码
```

String sysNumber = iCodingRuleManager.getNumber(billInfo, companyID);

2.12　获取汇率

IExchangRate iexchangRate = ExchangeRateFactory.getRemoteInstance(); 汇 率 对象

companyOrgUnitInfo.getBaseExchangeTable().getId();

//获取当前公司汇率表

ExchangeRateInfo erInfo = iexchangRate.getExchangeRate(exchangeTableID, sourceCurrencyID, destCurrencyID, Calendar.getInstance().getTime()); 取得该公司两种币别之间的汇率

2.13　数据库工具类

com.kingdee.eas.util.app.DbUtil 提供静态方法对数据库进行操作 ,不需要关注后台的数据库资源的释放 (由系统处理)

如： executeQuery、 execute

2.14　常用数据格式

tblMain.getColumn("billDate").getStyleAttributes().setNumberFormat("%{yyyy-MM-dd}t");

```
tblMain.getColumn("creditFor").getStyleAttributes().setNumberFormat("%r{#,##0.00}f");

tblMain.getColumn("debitFor"      ).getStyleAttributes().setHorizontalAlign(HorizontalAlignme
nt.RIGHT);
```

2.15　初始化单据分录中的数据

```
    protected IObjectV alue createNewDetailData(KDTable table)
{
    PurOrderEntryInfo t2 = new PurOrderEntryInfo();
        t2.set***(*);                //新增明细行时进行初始化
    return t2;
}
```

2.16　载入编辑界面时设置明细默认值

```
protected com.kingdee.bos.dao.IObjectValue createNewData()
{
    //new  一个值对象

    PurOrderInfo objectValue = new PurOrderInfo();

    objectValue.getEntrys().add(new PurOrderEntryInfo());

    objectValue.setCompany((CompanyOrgUnitInfo)( SysContext.getSysContext().getCurre
    ntFIUnit()));

    objectValue.setCreator((UserInfo)( SysContext.getSysContext().getCurrentUser()));

    return objectValue;
}
```

2.17　BigDecimal　类型的使用方式：

```
BigDecimal amounts = new BigDecimal(String.valueOf(0.00));
```

2.18　构造 ObjectUuidPK

```
ObjectUuidPK pk = new ObjectUuidPK(paymentInfo.getId());
```

2.19　组织转换

```
SysContext.getSysContext().getCurrentFIUnit().castToFullOrgUnitInfo()　　;
```

2.20　获取不同类型的组织视图

如：公司、成本中心、责任中心等

```
OrgViewF7 orgF7 = new OrgViewF7(this);

orgF7.setCurrentCUID(company.getId().toString());

orgF7.setMultiSelect(false);

orgF7.setOrgViewTypes(new OrgViewType[]{OrgViewType.COMPANY});

orgF7.setIsShowSub(true);

prmptCompany.setSelector(orgF7);
```

2.21　弹出指定的　F7 框

```
KDCommonPromptDialog dlg = (KDCommonPromptDialog) selector;

        if (dlg==null)
            {
                dlg = new KDCommonPromptDialog() ;
                if(dlg.getQueryInfo()==null)
                {
                    dlg.setQueryInfo(boxBizEle.getQueryAgent().getQueryInfo() );

                    try
                    {
                        dlg.setEntityViewInfo(new
EntityViewInfo(boxBizEle.getQueryAgent().getRuntimeEntityView().toString()));
                    }
                    catch (com.kingdee.bos.sql.ParserException e)
                    {
                        e.printStackTrace();
                    }

dlg.setSelectorCollection(boxBizEle.getQueryAgent().getSelectorCollection());

                    dlg.setQueryExecutor(boxBizEle.getQueryAgent().getQueryExecutor());

dlg.setEnabledMultiSelection(boxBizEle.getQueryAgent().isEnabledMultiSelection());

                    dlg.setReturnValueType(boxBizEle.getQueryAgent().getReturnValueType());
                }
            }
            dlg.show();

            if (!dlg.isCanceled())
            {
                boxBizEle.setValue( ((Object[])dlg.getData())[0]);

                ICell cell = pnlTables.getFoucusCell();

                if(cell!=null)
```

```
cell.setValue(((BgContractInfo)((Object[])dlg.getData())[0]).getContractNum());
        }
```

3.1 科目 F7

com.kingdee.eas.basedata.master.account.app.F7AccountViewQuery

3.2 科目表

com.kingdee.eas.basedata.master.account.app.AccountTableQuery

3.3 客户 F7

com.kingdee.eas.basedata.master.cssp.app.F7CustomerQuery

3.4 供应商 F7

com.kingdee.eas.basedata.master.cssp.app.F7SupplierQuery

3.5 客商统一码 F7

com.kingdee.eas.basedata.master.cssp.app.F7BizAnalysisCodeQuery

3.6 物料 F7

com.kingdee.eas.basedata.master.material.app.F7MaterialQuery

3.7 辅助核算 F7

com.kingdee.eas.basedata.master.auxacct.app.F7AsstAccountQuery

3.8 币别 F7

com.kingdee.eas.basedata.assistant.app.F7CurrencyQuery

3.9 辅助核算类型 F7

com.kingdee.eas.basedata.master.auxacct.app.F7AsstActTypeQuery

3.10 汇率 F7

com.kingdee.eas.basedata.assistant.app.F7ExchangeRateQuery

3.11 银行账户 F7

com.kingdee.eas.basedata.assistant.app.F7AccountBankQuery

3.12 银行 F7

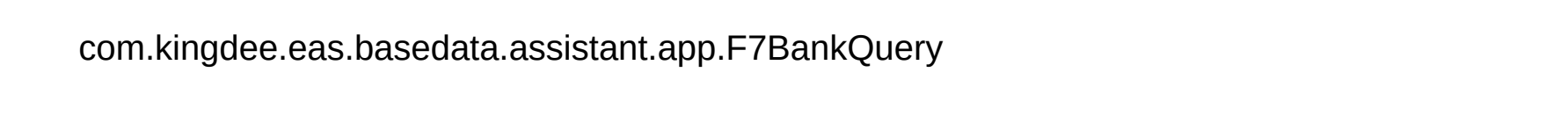
com.kingdee.eas.basedata.assistant.app.F7BankQuery

3.13 用户 F7

com.kingdee.eas.base.permission.app.F7UserQuery

3.14 银行 F7

com.kingdee.eas.basedata.assistant.app.F7BankQuery



4.1 单据新增代码

如：新增一行付款单

```
//构造一条付款单信息（构造值对象）

PaymentBillInfo payInfo = new PaymentBillInfo();

payInfo.setNumber(　　“1001”);

//关联用户
UserInfo userInfo = new UserInfo();

userInfo.setId(BOSUuid.read("867d5df6-00f8-1000-e000-0009c0a81089sysu"));　　// 通 过 BOSUuid 建立实体对象

 userInfo.setName(　　“John”);

 userInfo.setNumber(　　“2000”);
BigDecimal ex = new BigDecimal("343434.445");//　建立属性

BigDecimal ex2 = new BigDecimal(12345678901234567.8);//　错误

payInfo.setExchangeRate(ex);

payInfo.setCreator(userInfo);

payInfo.setAuditDate(new Date(System.currentTimeMillis()));//　时间属性

Timestamp createdate =　new Timestamp(System.currentTimeMillis());

payInfo.setCreateDate(createdate);

payInfo.setBizState(BillBizState.create);//　枚据属性

payInfo.setSave(true);//Boolean　属性

PaymentBillItemInfo itemInfo　　=　new PaymentBillItemInfo();//　建立分录

itemInfo.setLineNo(34);

itemInfo.setAmounts(ex2);

payInfo.getEntries().add(itemInfo);//　添加分录

payInfo.getEntries().add(new ErrandTaskItemInfo());//　添加分录
```

调用新增方法进行保存

IPaymentBill iPayBill = PaymentBillFactory.getRemoteInstance();//　　获取实体

IObjectPK pk = iPayBill.addnew(payInfo);//　保存值对象，返回逻辑键，可以通过逻辑键获取值对象（如下）

4.2 单据修改代码

修改一条付款单信息

```
IPaymentBill iPayBill = PaymentBillFactory.getRemoteInstance();

String id="8b35b903-00f8-1000-e000-0008c0a81089errt";

ObjectUuidPK pk =  new ObjectUuidPK(BOSUuid.read(id));//  建立逻辑键

PaymentBillInfo payInfo = iPayBill.getValue(pk);//  获取值对象；

System.out.println("this:"+ payInfo.getExchangeRate());//  获取属性

payInfo.setExchangeRate(ex);// 修改属性属性

iPayBill.update(pk, payInfo);//  更新数据
```

4.3 单据删除代码

iPayBill.remove(pk);// 通过逻辑键删除单据内容获取集合

4.4 获取集合

```
CurrencyInfo cur = new CurrencyInfo();

ICurrency c = CurrencyFactory.getRemoteInstance();//  建立实体对象
EntityViewInfo evi = new EntityViewInfo();//　　建立视图信息

FilterInfo i = new FilterInfo();//　建立过滤条件

i.getFilterItems().add(new FilterItemInfo("number", "ddff", CompareType.EQUELS));
```

```
i.getFilterItems().add(new FilterItemInfo("id", "ddff", CompareType.EQUELS));

i. setMaskString(" (#0 and #1)");

evi.setFilter(i);//  添加过滤条件

evi.getSelector().add(new SelectorItemInfo("id"));//    添加获取属性

evi.getSelector().add(new SelectorItemInfo("*"));

evi.getSelector().add(new SelectorItemInfo("company.id"));

evi.getSelector().add(new SelectorItemInfo("company.name"));

//三种获取集合方法

1.      获取所有数据：
CurrencyCollection co = c. getCurrencyCollection ();//    获取集合


2.      获取满足上述条件的所有数据
CurrencyCollection co = c.getCurrencyCollection(evi);//    获取集合

3.      获取满足  oql 所写的条件的所有数据
CurrencyCollection co = c. getCurrencyCollection (evi);//    获取集合
```

4.5 获取值对象

```
CurrencyInfo cur = new CurrencyInfo();
ICurrency c = CurrencyFactory.getRemoteInstance();//  建立实体对象
String id="8b35b903-00f8-1000-e000-0008c0a81089errt";
ObjectUuidPK pk =   new ObjectUuidPK(BOSUuid.read(id));//    建立逻辑键
c.getValue(pk);     或  : c.getCurrencyInfo(pk);
```

4.6 界面之间传递参数

```
1.    收集父界面要传递给子界面的参数集
HashMap map = new HashMap();
            map.put("Owner", this); //  必须。被启动    UI  的父  UI  对象
```

```
                map.put("EASMode", new Integer(this.EASMode));
                map.put("table", this.accountTablePrompBox.getData());
                map.put("cu", currentCtrlUnit);
                map.put("auxAccount", this.asstAccountPromptBox.getData());

                IUIFactory uiFactory = null;
                uiFactory = UIFactory
            .createUIFactory("com.kingdee.eas.base.uiframe.client.UIModelDialogFactory"); //
以模态对话框方式启动

IUIWindow                                    uiWindow                                    =
uiFactorycreate("com.kingdee.eas.basedata.master.auxacct.client.AccountSelectUI",    /*   被启动对
象的类名称    */
                                                map);
                    uiWindow.show();
```

2. 在子界面获取传递下来的数据

```
private void loadContext()
{
    int mode =    ((Integer)this.getUIContext().get("EASMode")).intValue();
    AccountTableInfo accountTableInfo = (AccountTableInfo) this.getUIContext().get("table");
    CtrlUnitInfo cuInfo = (CtrlUnitInfo) this.getUIContext().get("cu");
AsstAccountInfo asstAccountInfo = (AsstAccountInfo) this.getUIContext().get("auxAccount");
    AuxAccountEditUI ui = (AuxAccountEditUI) this.getUIContext().get("Owner");
}
```

4.7 给 Query 传过滤条件

```
    EntityViewInfo evi = new EntityViewInfo();
        FilterInfo filterInfo = new FilterInfo(); //     建立过滤条件
        filterInfo.getFilterItems().add(
                new FilterItemInfo("typelink.id", info.getId(),
                        CompareType.EQUALS));
        filterInfo.getFilterItems().add(
                new FilterItemInfo("currencyCompany.id", companyID,
                        CompareType.EQUALS));
    filterInfo.setMaskString("#0 and #1 ");
    evi.setFilter(filterInfo);

    if (mainQuery == null) {
        mainQuery = new EntityViewInfo();
    }

    mainQuery.setFilter(filterInfo); //   添加过滤条件
```

this.execQuery();

4.8 接口方法的访问方式

3. 客户端访问

```
CurrencyInfo cur = new CurrencyInfo();
ICurrency c = CurrencyFactory.getRemoteInstance();//  建立实体对象
c. getCurrencyCollection();
```

4. 服务端访问

```
CurrencyInfo cur = new CurrencyInfo();
ICurrency c = CurrencyFactory. getLocalInstance (ctx);//  建立实体对象
c. getCurrencyCollection();
```

4.9 传递上下文参数的接口访问方式

```
PurOrderInfo cur = new PurOrderInfo();
IPurOrder c = PurOrderFactory. getRemoteInstanceWithObjectContext(ctx);//  建立实体对象
c. getPurOrderCollection();
```

另 query 也支持上下文参数：

```
IQueryExecutor exec = QueryExecutorFactory.getRemoteInstance(queryPK,ctx);
…
```

4.10 控件的初始化

1. 使用枚举给 **ComboBox** 控件赋值

```
public void setReceiveType()
    {
        this.kDComboBox2.removeAllItems();
        List list = new List();
        Iterator it = ReceiveType.iterator();
        while (it.hasNext())
        {
            list.add(ReceiveType.getEnum(it.toString()).toString());
        }
    }
```

2. 使用值对象集合给 **ComboBox** 控件赋值

```
//  获取结算方式列表
```

```
IAsstActTypeDefault iAsstActType = AsstActTypeDefaultFactory.getRemoteInstance();
AsstActTypeDefaultCollectioni asstActTypeColl = iAsstActType
                .getAsstActTypeDefaultCollection(" where isAccountCussent = 1");
AsstActTypeDefaultInfo defaultVal = null;
int count = asstActTypeColl.size();
Object[] typeValue = new Object[count];
for (int i = 0; i < count; i++)
{
        AsstActTypeDefaultInfo value = (AsstActTypeDefaultInfo) asstActTypeColl.get(i);
                if (value.isIsDefaultAccountPayable())
                {
                        defaultVal = value;
                }
                ObjectUuidPK                pk                =                new
            ObjectUuidPK(BOSUuid.read(value.getAsstActType().getString("id")));
                AsstActTypeInfo                asstInfo                =
AsstActTypeFactory.getRemoteInstance().getAsstActTypeInfo(pk);
                //cbi[i] = new ComBoxInfo(asstInfo, asstInfo.getName(), true);
                String typeName = asstInfo.getName();
                String typeId = asstInfo.getId().toString();
                String typeQueryName = asstInfo.getDefaultQueryName();
                String typeTableName = asstInfo.getRealtionDataObject();
                int csType = 0; //asstActTypeInfo.getCsType().getValue();
                asstActType    =    new    AsstActTypeUtils(typeName,    typeId,    typeTableName,
typeQueryName, csType);

                typeValue[i] = (Object) asstActType;

            }

            ///String strQueryName = "F7SupplierQuery";
            this.kDComAccountCussentType.removeAllItems();
            this.kDComAccountCussentType.addItems(typeValue);
            this.kDComAccountCussentType.setSelectedIndex(0);
```

4.11　F7 赋值

```
        ObjectUuidPK pk = new ObjectUuidPK(id);
                IObjectValue                objVal                =
        DynamicObjectFactory.getRemoteInstance().getValue(pk.getObjectTypei(), pk);
                bizPrompt.setData(objVal);
private void setCompanyF7()
    {
```

```
//公司的  F7 的设置
OrgType[] CompanyType = { OrgType.Company };

//首先定义需要显示那些树，这里只显示财务树
//为 F7 控件指定  PromptBox
this.bizPromptCompany.setEditFormat("$number$");
this.bizPromptCompany.setDisplayFormat("$name$");
this.bizPromptCompany.setEditable(true);
this.bizPromptCompany.setCommitFormat("$number$");

        OrgUnitTreePromptBox boxCompany = new OrgUnitTreePromptBox(this,
        CompanyType, null,
                OrgSelect.OnlySelectCompanyEntity, false, false, false);
        bizPromptCompany.setSelector(boxCompany);
        bizPromptCompany.setData(currentCompany);
        bizPromptCompany.setEnabled(false);
}

//设置币别
        private void setCurrency() throws EASBizException, BOSException
        {
                ICurrency iCurrency = null;
                CurrencyCollection con = null;
                try
                {
                        iCurrency = CurrencyFactory.getRemoteInstance();
                        con = iCurrency.getCurrencyCollection(true);
                }
                catch (Exception e)
                {
                        // TODO  自动生成  catch  块
                MsgBox.showError(this, EASResource.getString(resClassName, "currencyFail"));
                SysUtil.abort();
                }
                if (con != null)
                {
                        Object[] typeValue = new Object[con.size()];
                        int j = 0;
                        for (int i = 0; i < con.size(); i++)
                        {
                                CurrencyInfo currencyInfo = (CurrencyInfo) con.get(i);
                                String typeName = currencyInfo.getName();
                                String typeId = currencyInfo.getId().toString();
                                String typeQueryName = currencyInfo.getNumber();
```

```
                asstActType = new AsstActTypeUtils(typeName, typeId, typeQueryName);
                    if (asstActType != null)
                    {
                            typeValue[i] = (Object) asstActType;
if(typeId.equalsIgnoreCase(((CurrencyInfo)currentCompany.getBaseCurrency()).getId().toString()
))
                            {
                                j = i;
                            }
                        }

                        this.bizPromptcurrency.removeAllItems();
                        this.bizPromptcurrency.addItems(typeValue);
                        //默认值为当前公司的本位币
                        this.bizPromptcurrency.setSelectedIndex(j);
                }
                else
                {
                MsgBox.showError(this, EASResource.getString(resClassName, "currencyFail"));
                        SysUtil.abort();
                }
```

4.12　设置单据分录单元格格式

```
// 设置单据分录格式
                getDetailTable().getColumn(TB_RELABILL).setWidth(100);
                getDetailTable().getColumn(TB_ORGUNIT).setWidth(180);
                getDetailTable().getColumn(TB_BIZMAN).setWidth(100);
                getDetailTable().getColumn(TB_PAY  AMOUNT).setEditor(number_CellEditor);
getDetailTable().getColumn(TB_PAYAMOUNT).getStyleAttributes().setNumberFormat("%r-
[=]{#.00}f");
        getDetailTable().getColumn(TB_PAYAMOUNT).getStyleAttributes().setHorizontalAlign(Hor
        izontalAlignment.RIGHT);
```

4.13　设置单元格可编辑

```
for (int i = 0; i < kdtEntrys.getRowCount(); i++)
```

```
{
        kdtEntrys.getRow(i).getCell("orgUnit").getStyleAttributes().setLocked(false);
        kdtEntrys.getRow(i).getCell("bizMan").getStyleAttributes().setLocked(false);
}

//如果折扣金额为    null 则初始化为   0
 if (kdtEntrys.getRow(i).getCell("discountAmount").getValue() == null)
 {
          kdtEntrys.getRow(i).getCell("discountAmount").setValue(new BigDecimal("0.0"));
  }

//汇总分录行金额
BigDecimal amount = new BigDecimal("0");
for (int i = 0, n = kdtEntrys.getRowCount(); i < n; i++)
{
amount = amount.add(UIRuleUtil.getBigDecimal(UIRuleUtil.getBigDecimalValue
(kdtEntrys.getCell(i,TB_PAYAMOUNT).getValue())));
}
this.txtTotalAmounts.setText(null);
this.txtTotalAmounts.setEnable(false);
this.txtTotalAmounts.setText(amount);

//和零比较
if (amount.compareTo(new BigDecimal("0.00")) == 0)
```

4.14　删除行

```
public void actionDeleteLine_actionPerformed(ActionEvent e)throws Exception {
        IRow row = getSelectedRow();
        if (row != null) {
            kDTable1.removeRow(row.getRowIndex());
        }
    }
```

4.15　F7 专用选择界面的设置

```
public void setF7Selector() throws Exception
    {
```

```
KDBizPromptBox bizPromptBox = new KDBizPromptBox();//        要绑定的   F7 控件
CoreUIObject ui = null;//    父界面对象
CompanyOrgUnitInfo companyInfo = null; //      当前财务组织

// 客户
bizPromptBox.setSelector(new   GeneralKDPromptSelectorAdaptor(bizPromptBox,    new
F7CustomerTreeDetailListUI(), ui));

// 供应商
bizPromptBox.setSelector(new   GeneralKDPromptSelectorAdaptor(bizPromptBox,    new
F7SupplierTreeDetailListUI(), ui));

// 物料
bizPromptBox.setSelector(new   GeneralKDPromptSelectorAdaptor(bizPromptBox,    new
F7MaterialTreeListUI(), ui));

// 职员
HashMap map = new HashMap();
map.put(PersonF7UI.ALL_ADMIN, "YES");
bizPromptBox.setSelector(new PersonPromptBox(ui, map));

// 公司
CompanyF7 org = new CompanyF7(ui);
org.setRootUnitID("");
bizPromptBox.setSelector(new CompanyF7(ui));

// 成本中心
bizPromptBox.setSelector(new CostCenterF7(ui));

// 行政组织
bizPromptBox.setSelector(new AdminF7(ui));

// 采购组织
bizPromptBox.setSelector(new PurchaseF7(ui));

// 库存组织
bizPromptBox.setSelector(new StorageF7(ui));

// 销售组织
bizPromptBox.setSelector(new SaleF7(ui));

// 利润中心组织
bizPromptBox.setSelector(new ProfitCenterF7(ui));
```

```
        // 利润中心组织
        bizPromptBox.setSelector(new ProfitCenterF7(ui));

        // 科目
        bizPromptBox.setSelector(new AccountPromptBox(ui, companyInfo, new FilterInfo()));
}
```

4.16 获取各模块系统状态信息

```
public void getSystemStatue() throws EASBizException, BOSException
    {
        CompanyOrgUnitInfo companyInfo = null; //      当前财务组织
        // SystemStatusCtrolUtils  工具类可获取各模块系统状态信息，如当前会计期间，系统是否启用或关闭等信息
        SystemStatusCtrolUtils.getCurrentPeriod(null/*     上下文信息，如在客户端使用可为null */, SystemEnum.ACCOUNTSPAY  ABLE, companyInfo/**/);
    }
```

4.17 获取当前登陆信息

```
public void getSystemInfo()
    {
        // SysContext 工具类可获取当前登陆用户的信息，可根据需要进行调用。
        //  举两例如下：
        SysContext.getSysContext().getCurrentUserInfo();//    获取当前登陆用户信息
        SysContext.getSysContext().getCurrentFIUnit();//    获取当前财务组织
    }
```

4.18 获取参数平台参数设置的示例代码

```
public void getParam() throws EASBizException, BOSException
    {
        CompanyOrgUnitInfo companyInfo = null; //      当前财务组织
        ObjectUuidPK orgPk = new ObjectUuidPK(companyInfo.getId());

        //  ParamManager 工具类提供了不同的方法获取参数俱，可根据自己的需要进行调用，
        //  示例如下：
```

ParamManager.getParamValue(null/*　上下文信息，客户端调用可以为　　null */, orgPk, "AR_INIT_CHECK_TYPE");
}

4.19　网络互斥功能示手工控制

```
public void doMutexService()
{
	IMutexServiceControl mutex = MutexServiceControlFactory.getRemoteInstance();

	UserInfo user = null; //　请求锁有用户
	String billId = null; //　　要锁定 /解锁的单据　Id

	//  请求锁定
	mutex.requestObjIDForUpdate(billId, user.getString("id"));

	//  解除锁
	mutex.releaseObjIDForUpdate(billId);
}
```

Tree:TreeBase（增加 parentid 级次）
Data:DataBase、BillBase （增加关联关系，对　Tree 的引用）

4.20　Tree- List　　实现方法 1

```
protected ITreeBase getTreeInterface() throws Exception
{
	return com.kingdee.eas.custom.TreeGroupTestTreeFactory.getRemoteInstance();
}
```

4.21　Tree-List　　点击树上结点时形成过滤条件时的字段

如： protected String getQueryFieldName()
{
return "treeid.id";
}

4.22　Tree- 树形控件的初始化级次　　[optional]

```
protected int getTreeInitialLevel()
{
	return TreeBuilderFactory.DEFAULT_INITIAL_LEVEL;
}
```

4.23　Tree- 树形空间的默认展开级次　　[optional]

```
protected int getTreeExpandLevel()
{
	return TreeBuilderFactory.DEFAULT_EXPAND_LEVEL;
}
```

4.24　Tree- 返回树形控件的根名称

```
protected String getRootName()
{
return "TreeGroupTest";
}
```

4.25　Tree- 数据过滤（重载实现对树的过滤　）

getDefaultFilterForTree() 方法，是默认的　cu 过滤条件

4.26　Tree- 控件基本使用

4.26.1 初始化树形控件

//treeMain 控件名

```
protected void initTree() throws Exception
{

TreeSelectionListener[] listeners = treeMain.getTreeSelectionListeners();

TreeSelectionListener     treeSelectionListener = listeners[0];

treeMain.removeTreeSelectionListener(treeSelectionListener);

ITreeBuilder treeBuilder
 =          TreeBuilderFactory.createTreeBuilder(getLNTreeNodeCtrl(),getTreeInitialLevel()          ,
getTreeExpandLevel(),getDefaultFilterForTree());

if (getRootName() != null)
{
        KDTreeNode rootNode = new KDTreeNode(getRootObject());

        ( (DefaultTreeModel) treeMain.getModel()).setRoot(rootNode);

}
else
{
        ( (DefaultTreeModel) treeMain.getModel()).setRoot(null);
}

//将数据填入控件中
treeBuilder.buildTree(treeMain);

//增加选择事件
treeMain.addTreeSelectionListener(treeSelectionListener);

}
```

4.26.2 返回选中的树结点

```
public KDTreeNode getSelectedTreeNode()
{
	return (KDTreeNode) treeMain.getLastSelectedPathComponent();
}
```

4.26.3 返回树结点的值

```
return (TreeBaseInfo) ((KDTreeNode)
		treeMain.getLastSelectedPathComponent()).getUserObject();
```

4.26.4 删除类别时刷新当前结点的父结点，并定位到当前结点的父结点。

```
KDTreeNode treeNode = (KDTreeNode) treeMain
	.getLastSelectedPathComponent();

if (treeNode != null && treeNode.getParent() != null)
{
	TreePath parentPath = treeMain.getSelectionPath().getParentPath();

	KDTreeNode parentNode = (KDTreeNode) treeNode.getParent();

	parentNode.remove(treeNode);

	treeMain.updateUI();

	treeMain.setSelectionPath(parentPath);

	treeBuilder.refreshTreeNode(treeMain , parentNode ,this.getDefaultFilterForTree());

	treeMain.expandPath(parentPath);

}
```

```
//刷新  Query 数据
CacheServiceFactory.getInstance().discardQuery(this.mainQueryPK);
```

4.26.5 设置选中根结点

```
treeMain.setSelectionRow(0);
```

4.26.6 类别新增与修改时，刷新当前选中结点

```
KDTreeNode treeNode = (KDTreeNode) treeMain
        .getLastSelectedPathComponent();

if (treeNode != null && treeNode.getParent() == null)
{
        TreePath oldPath = treeMain.getSelectionPath();

        treeBuilder.refreshTreeNode(treeMain , treeNode,this.getDefaultFilterForTree());

        treeMain.setSelectionPath(oldPath);

        treeMain.expandPath(oldPath);

}
```

4.26.7 修改类别时刷新当前结点的父结点，并定位到当前结点

```
KDTreeNode treeNode = (KDTreeNode) treeMain.getLastSelectedPathComponent();
if (treeNode != null && treeNode.getParent() != null)
{
        TreePath oldPath = treeMain.getSelectionPath();
```

```
TreePath parentPath = treeMain.getSelectionPath().getParentPath();

KDTreeNode parentNode = (KDTreeNode) treeNode.getParent();

treeBuilder.refreshTreeNode(treeMain , parentNode ,this.getDefaultFilterForTree());

treeMain.setSelectionPath(TreePathUtil.getNewTreePath(treeMain ,
                                    treeMain.getModel() , oldPath));
    }
```

4.27 手工发送消息

```
//以下是发送一个消息的核心片断，如果针对的是某组织批量发送，自己的需要取得组织的用户 ID 集合，然后构造 receiver
SenderAgent senderAgent = SenderAgent.getSenderAgent();
Message message;
Locale[] lcla = getContextLocales(ctx);// 获取 ctx 的语言信息列表
Locale locale = null;
message = MessageFactory.newMessage("kingdee.workflow");// 生成一个消息对象
for (int j = 0, m = lcla.length; j < m; j++) {
    //此处循环进行多语言消息的设置
    locale = lcla[j];
    message.setLocaleStringHeader("title"," 标题 ", locale);// 设置消息标题
    message.setLocaleStringHeader("sender","发送人 " , locale);// 设置发送人， 属于文本， 不是ID
    message.setLocaleStringHeader("body", "消息体内容 ",Locale);// 设置消息体内容，根据具体业务自己设定
}

message.setIntHeader("type", MsgType.NOTICE_VALUE);// 设置消息类型为通知
message.setIntHeader("bizType", MsgBizType.WORKFLOW_V ALUE);// 业务类型设置为工作流
message.setIntHeader("sourceStatus", MsgSourceStatus.EMPTY_V ALUE);// 设置任务状态， 此处是通知消息，所以设置空
message.setIntHeader("priority", MsgPriority.MIDDLE_V ALUE);// 设置消息优先级，自己根据需要设定相应的级别

message.setStringHeader("databaseCenter", ctx.getAIS());// 得到数据中心
message.setStringHeader("solution", ctx.getSolution());// 设置解决方案

message.setStringHeader("receiver",
```

'4ff9eebb-0108-1000-e000-15acc0a813c813B7DE7F;4ff9eebb-0108-1000-e000-1db0c0a813c813B7DE7F'); // 设置接收者，后面那参数是用户　　ID，多个 ID 可用分号 ";" 分割
senderAgent.sendMessage(message); //发送消息

```
/**
 * 获取 ctx 的语言信息列表
 *
 * @param ctx
 * @return
 */
public static Locale[] getContextLocales(Context ctx) {
	Locale[] locales = null;
	SolutionInfo solu = MetaDataLoaderFactory.getLocalMetaDataLoader(ctx)
			.getSolution();

	if (solu != null) {
		LanguageCollection langs = solu.getLanguages();
		if (langs != null) {

			locales = new Locale[langs.size()];

			for (int i = 0; i < langs.size(); i++) {
				locales[i] = langs.get(i).getLocale();
			}
		}
	}

	return locales;
}
```

<完>