单据列表插件示例代码

修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. No** | **日期** | **编制\修订** | **校对** | **批准** | **修改的章节号** |
| V1.0 | 20150421 | 吴亮 | 张晋博 | 赖碧云 | 初始版本 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

问题和意见

如果你对文档有任何意见、问题或想法，或者你的问题未在此文档中找到答案，请通过电子邮件联系我们

jinbo\_zhang@kingdee.com

目录

[单据列表插件示例代码 1](#_Toc417395379)

[单据列表插件概要说明 4](#_Toc417395380)

[公共属性 4](#_Toc417395381)

[ListModel 4](#_Toc417395382)

[语法 4](#_Toc417395383)

[常用属性 5](#_Toc417395384)

[常用方法 5](#_Toc417395385)

[案例 – 汇率列表，对汇率按精度格式化显示 6](#_Toc417395386)

[ListView 8](#_Toc417395387)

[语法 8](#_Toc417395388)

[常用属性 9](#_Toc417395389)

[常用方法 9](#_Toc417395390)

[案例 – 销售订单列表对订单进行BOM配置 9](#_Toc417395391)

[事件 11](#_Toc417395392)

[BatchCopyData 11](#_Toc417395393)

[案例 –暂缺 11](#_Toc417395394)

[AfterBatchCopyData 11](#_Toc417395395)

[案例 –暂缺 12](#_Toc417395396)

[BeforeButtonClick 12](#_Toc417395397)

[AfterButtonClick 12](#_Toc417395398)

[AfterGetData 12](#_Toc417395399)

[案例 –网上银行，银行类型列表，刷新取数 12](#_Toc417395400)

[AfterMenuClick 13](#_Toc417395401)

[BeforeMenuClick 14](#_Toc417395402)

[CellDbClick 14](#_Toc417395403)

[CellFormat 14](#_Toc417395404)

[CreateFilterEditorControl 14](#_Toc417395405)

[EntryHyperlinkButtonClick 14](#_Toc417395406)

[FormatCellValue 14](#_Toc417395407)

[案例 –股权关系列表，持股比例精度按参数设置控制 15](#_Toc417395408)

[OnFormatRowConditions 16](#_Toc417395409)

[案例 –暂缺 17](#_Toc417395410)

[ListCreateColumns 17](#_Toc417395411)

[案例 –暂缺 17](#_Toc417395412)

[ListInitialize 17](#_Toc417395413)

[案例 –调整分录模板列表初始化时获取用户参数 18](#_Toc417395414)

[ListRowDoubleClick 19](#_Toc417395415)

[案例 –采购价目表列表界面双击 19](#_Toc417395416)

[OnGetConvertRule 20](#_Toc417395417)

[案例 –简单生产入库下推简单生产领料订单 20](#_Toc417395418)

[OnShowConvertOpForm 22](#_Toc417395419)

[案例 –采购收料单列表界面下推筛选掉资产卡片 23](#_Toc417395420)

[OnShowTrackResult 24](#_Toc417395421)

[案例 –生产领料单上查替换物料清单为生产订单 24](#_Toc417395422)

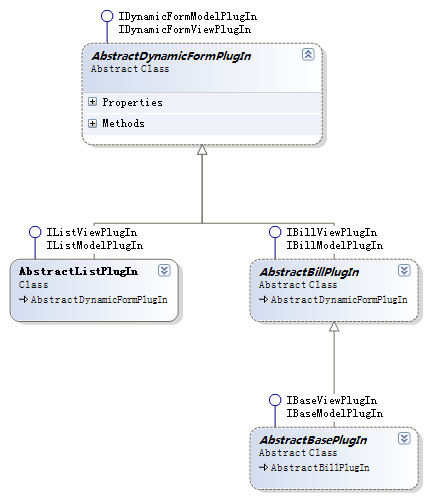
[PrepareFilterParameter 27](#_Toc417395423)

[案例 –信用检查规则查找列表数据筛选 27](#_Toc417395424)

[PrepareFuncPermissionDataRule 28](#_Toc417395425)

# 单据列表插件概要说明

平台插件继承体系图如下



如上图所示，列表插件基类AbstractListPlugIn继承自动态表单基类插件AbstractDynamicFormPlugIn，并实现了IlistViewPlugIn和IlistModelPlugIn两个插件接口，分别是视图层和模型层的插件接口。

# 公共属性

## ListModel

模型层访问接口，Model属性做强制转换

### 语法

**属性定义**：

|  |  |
| --- | --- |
| C# |  |
| public virtual IListModel ListModel  {  get  {  if (this.Model != null && this.Model is IListModel)  {  return (IListModel)this.Model;  }  else  {  return null;  }  }  } | |

**备注**：

无

### 常用属性

|  |  |
| --- | --- |
| **名称** | **描述** |
| BillBusinessInfo | 关联单据的业务逻辑元数据 |
| FieldKeyMap | 列表上显示的字段，其字段名FieldName和字段标识Key的对应关系 |
| FilterParameter | 列表过滤条件参数对象 |
| GlobalParameter | 单据参数配置中配置的单据全局参数 |
| ParameterData | 用户参数数据，一般是选项菜单界面的配置数据 |
| Header | 列表表头对象 |
| FieldKeyMap | 字段和列表所选字段key的映射关系 |
| QueryInfo | 列表查询元数据，包括组织隔离字段、可供查询的实体、字段列表等等 |
| Limit | 每页行数 |
| StartRow | 开始行索引 |

### 常用方法

|  |  |
| --- | --- |
| **名称** | **描述** |
| GetData(int startRow, int limit, bool needRefreshTree) | 获取列表数据 |
| GetData(int start, int rows, bool needRefreshTree, string orderInfo) | 获取列表数据，带排序 |
| GetRow(IDataRow r) | 获取选中行对象 |
| GetGroupInfo(string groupRows) | 获取分组信息 |
| GetHeadFieldSummaryData(List<Field> summaryHeadFields) | 获取单据头字段的汇总数据 |
| GetSummaryData(List<Field> summaryFields) | 获取汇总数据（单据体） |
| GetPagingRowCount() | 列表总行数，为了性能，仅在最后一页取数 |
| GetPKData() | 获取当前列表查询的所有主键数据 |
| GetRowCount() | 获取列表总行数和单据记录数 |
| Refresh() | 刷新列表 |
| RefreshByFilter() | 根据过滤条件，重新取数，刷新列表 |

### 案例 – 汇率列表，对汇率按精度格式化显示

**代码来源**

汇率管理列表插件

Kingdee.K3.BD.Common.Business.PlugIn.RateList,

Kingdee.K3.BD.Common.Business.PlugIn

**需求背景**

汇率列表界面的直接汇率和间接汇率列的显示值，需要根据所在汇率类型中定义的精度进行格式化显示

**实现方案**

捕获FormatCellValue事件，对直接汇率和间接汇率进行格式化。先需要加载当前行的汇率数据，以拿到精度信息，然后进行格式化。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS;  using Kingdee.BOS.Core.DynamicForm.PlugIn.Args;  using Kingdee.BOS.Core.List;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.List.PlugIn.Args;  using Kingdee.BOS.Core.Metadata;  using Kingdee.BOS.Core.SqlBuilder;  using Kingdee.BOS.Orm.DataEntity;  using Kingdee.BOS.ServiceHelper;  using Kingdee.BOS.Util;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Linq;  namespace Kingdee.K3.BD.Common.Business.PlugIn  {  /// <summary>  /// 汇率管理列表插件  /// </summary>  public class RateList : AbstractListPlugIn  {  /// <summary>  /// 直接汇率和间接汇率精度集 rateDecimal[主键PKID][0：直接汇率，1：间接汇率]=精度  /// </summary>  private IDictionary<long, int[]> rateDecimal = new Dictionary<long, int[]>();  public override void FormatCellValue(FormatCellValueArgs args)  {  base.FormatCellValue(args);  string filedName = args.Header.FieldName;//字段名  //直接汇率或间接汇率列  if (filedName.Equals("FExchangeRate", StringComparison.OrdinalIgnoreCase)  || filedName.Equals("FReverseExRate", StringComparison.OrdinalIgnoreCase))  {  long pkId = Convert.ToInt64  (this.ListModel.GetRow(args.DataRow).PrimaryKeyValue);  //暂存汇率字段中，不存在当前分录的记录  if (!rateDecimal.ContainsKey(pkId))  {  //加载直接汇率和间接会精度  QueryBuilderParemeter para = new QueryBuilderParemeter()  {  FormId = "BD\_Rate",  SelectItems = SelectorItemInfo.CreateItems(  "FRateID,FRATETYPEID.FDIGITS,FRATETYPEID.FREVERSEDIGITS"),  FilterClauseWihtKey = "FRateID=@FRateID"  };  List<SqlParam> lstSqlPara = new List<SqlParam>();  lstSqlPara.Add(new SqlParam("@FRateID", DbType.Int64, pkId));  DynamicObjectCollection obj = QueryServiceHelper.GetDynamicObjectCollection(this.Context, para, lstSqlPara);  if (obj != null && obj.Count > 0)  {  DynamicObject dy = obj.First();  rateDecimal.Add(pkId, new int[2]);  rateDecimal[pkId][0] = Convert.ToInt32(dy["FRATETYPEID\_FDIGITS"]);  rateDecimal[pkId][1] = Convert.ToInt32(dy["FRATETYPEID\_FREVERSEDIGITS"]);  }  }  //格式化值  if (filedName.Equals("FExchangeRate", StringComparison.OrdinalIgnoreCase))  {  args.FormateValue = string.Format("{0:F" + rateDecimal[pkId][0] + "}", args.Value);  }  else if (filedName.Equals("FReverseExRate", StringComparison.OrdinalIgnoreCase))  {  args.FormateValue = string.Format("{0:F" + rateDecimal[pkId][1] + "}", args.Value);  }  }  }  }  } | |

## ListView

视图层访问接口，View属性做强制转换

### 语法

**属性定义**：

|  |  |
| --- | --- |
| C# |  |
| public virtual IListView ListView  {  get  {  if (null != this.View && this.View is IListView)  {  return (IListView)this.View;  }  else  {  return null;  }  }  } | |

**备注**：

无

### 常用属性

|  |  |
| --- | --- |
| **名称** | **描述** |
| BillBusinessInfo | 对应单据的逻辑元数据 |
| BillLayoutInfo | 对应单据的布局元数据 |
| Model | 列表对应的模型，强转为IListModel |
| OpenParameter | 页面打开时传入的参数 |
| SelectedRowsInfo | 当前列表上被选中的行记录，通过复选框勾选的 |
| CurrentSelectedRowInfo | 当前列表上当前选择行记录，直接点击选中的 |
| CurrentPageRowsInfo | 当前列表所有单据的行信息 |
| CanDealWithRows | 可以处理的行数据，可能被插件排除了某些单据ID |

### 常用方法

|  |  |
| --- | --- |
| **名称** | **描述** |
| SetRowFontColor(int row, string color) | 设置列表的行文本颜色 |
| FormatCellValue(ListHeader header, IDataRow dr, object value) | 格式化单元格的值，应用于列表格式化、引出数据、打印等 |

### 案例 – 销售订单列表对订单进行BOM配置

**代码来源**

销售订单列表插件

Kingdee.K3.SCM.Sal.Business.PlugIn.SaleOrderList,

Kingdee.K3.SCM.Sal.Business.PlugIn

**需求背景**

销售订单列表菜单项【配置】，点击时进行订单的BOM配置，配置只能对一张订单的唯一一行分录进行。

**实现方案**

捕获菜单点击事件，如果是点击的【配置】，则判断是否符合BOM配置的条件，符合，则进行配置，否则，直接退出菜单操作。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS.Core.DynamicForm.PlugIn.Args;  using Kingdee.BOS.Core.List;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Util;  using System;  using System.ComponentModel;  using System.Linq;  namespace Kingdee.K3.SCM.Sal.Business.PlugIn  {  /// <summary>  /// 销售订单叙事薄列表插件  /// </summary>  public class SaleOrderList : AbstractListPlugIn  {  public override void BarItemClick(BarItemClickEventArgs e)  {  ListSelectedRowCollection selectRows = this.ListView.SelectedRowsInfo;  string objectID = this.ListView.BillBusinessInfo.GetForm().Id;  string entiryKey = this.ListView.Model.BillBusinessInfo.GetField("FQty").EntityKey;  switch (e.BarItemKey.ToUpperInvariant())  {  case "TBBOMCONFIG":  //分是否显示单据体处理  if (selectRows.Count() != 1 || this.ListView.SelectedRowsInfo[0].EntryEntityKey != "FSaleOrderEntry"  || this.ListView.SelectedRowsInfo[0].EntryPrimaryKeyValue.IsNullOrEmptyOrWhiteSpace())  {  e.Cancel = true;  this.View.ShowMessage("请选择唯一一行单据分录进行配置操作！");  break;  }  //私有函数，进行BOM配置，具体逻辑略去  BomConfigViewEdit();  break;  //其他case分支处理逻辑略去  }  }  }  } | |

# 事件

## BatchCopyData

**触发时机**

调用批量复制操作，获取目标数据包时触发

**应用场景**

批量复制单据时，调整复制的单据主键或自定义复制的目标数据包。如果仅调整主键，那么平台会根据新的主键值去加载目标数据包；如果定义了目标数据包，则平台会直接使用。

**关键字**

批量复制

**备注**

当BatchCopyData事件中直接给出了目标数据包，后面不会再触发AfterBatchCopyData

### 案例 –暂缺

## AfterBatchCopyData

**触发时机**

批量复制操作，获取到平台加载的的目标数据包之后触发。

**应用场景**

插件对平台加载的目标数据包再做修改、调整

**关键字**

批量复制

**备注**

模型层事件

如果BatchCopyData事件中直接给出了目标数据包，不会再触发本事件

### 案例 –暂缺

## BeforeButtonClick

已失效，功能被ButtonClick替代

参考参考《动态表单插件示例代码.docx》文档中AButtonClick事件

## AfterButtonClick

参考《动态表单插件示例代码.docx》文档中AfterButtonClick事件

## AfterGetData

**触发时机**

列表取数完成后，在数据发送到前端之前触发

**应用场景**

**关键字**

取数完成

**备注**

视图层事件

### 案例 –网上银行，银行类型列表，刷新取数

**代码来源**

财务领域，网上银行，银行类型列表插件

Kingdee.K3.FIN.WB.Business.PlugIn.BankTypeList,

Kingdee.K3.FIN.WB.Business.PlugIn

**需求背景**

网上银行，银行类型列表取数后，如果发现没有数据，则需要后台从银企平台数据库同步银行类型数据到Cloud数据库。

**实现方案**

捕获AfterGetData事件，判断当前也的数据行数，如果为0，说明列表没有数据，调用服务端事件接口，从银企平台同步数据过来。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Util;  using Kingdee.K3.FIN.WB.ServiceHelper;  namespace Kingdee.K3.FIN.WB.Business.PlugIn  {  public class BankTypeList : AbstractListPlugIn  {  /// <summary>  /// 组织ID  /// </summary>  public string OrgID { get; set; }  public override void AfterGetData()  {  base.AfterGetData();  if (this.ListView.CurrentPageRowsInfo.Count < 1)  {  //说明没有数据，需要重新刷新下  if (this.OrgID.IsNullOrEmptyOrWhiteSpace())  {  return;  }  //财务，网上银行服务接口，同步银企平台银行类型数据  Kingdee.K3.FIN.WB.ServiceHelper.EbankAutoFillServiceHelper.AutoFillBankList(this.Context, OrgID);  }  }  }  } | |

## AfterMenuClick

已失效

## BeforeMenuClick

已失效

## CellDbClick

已失效

## CellFormat

已失效

## CreateFilterEditorControl

已失效

## EntryHyperlinkButtonClick

已失效，前端不会触发，请使用EntryButtonCellClick事件

## FormatCellValue

**触发时机**

列表显示，在获取数据之后，数据发送到前端之前触发，用来格式化列表上的单元格数据

**应用场景**

插件在加载列表数据时格式化字段值，满足列表显示多样化的需求

**关键字**

格式化

**备注**

视图层插件事件

设计器中，表单整体属性中有【列表条件格式化】属性，通过配置这个属性，可以对列表做一些格式化显示配置。如果该属性不能完全满足需求，那么可以通过该事件做更丰富的列表格式化显示设置。

### 案例 –股权关系列表，持股比例精度按参数设置控制

**代码来源**

财务合并报表：股权关系-列表插件

Kingdee.K3.FIN.CR.Business.PlugIn.InvestList,

Kingdee.K3.FIN.CR.Business.PlugIn

**需求背景**

【合并报表参数设置】中有对持股比例精度进行控制。股权关系列表中，在显示持股比例值的时候，需要获取该参数，并且对持股比例值的小数位数参照该精度进行控制

**实现方案**

插件在初始化表单事件OnInitialize中获取系统参数；捕获FormatCellValue事件，在对持股比例值进行格式化的时候，使用该参数格式化数值。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Linq;  using System.Text;  using Kingdee.BOS.Core.DynamicForm.PlugIn.Args;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.List.PlugIn.Args;  using Kingdee.K3.FIN.CR.Common.Core;  using Kingdee.K3.FIN.CR.ServiceHelper;  namespace Kingdee.K3.FIN.CR.Business.PlugIn  {  [Description("股权关系-列表插件")]  public class InvestList : AbstractListPlugIn  {  /// <summary>  /// 持股比例精度  /// </summary>  private int \_ratePrecision = 0;  private string \_formartString = string.Empty;  /// <summary>  /// 持股比例精度 格式化字符串  /// </summary>  private string FormartString  {  get  {  if (!string.IsNullOrWhiteSpace(\_formartString))  return \_formartString;  char[] sp = new char[\_ratePrecision];  for (int i = 0; i < \_ratePrecision; i++)  {  sp[i] = '0';  }  \_formartString = string.Format("#.{0}", new string(sp));  return \_formartString;  }  }  public override void OnInitialize(InitializeEventArgs e)  {  base.OnInitialize(e);  //调用系统服务帮助类，获取系统参数【合并报表参数设置】中的持股比例精度控制的值  \_ratePrecision = CRSystemParamerServiceHelper.GetSystemParamter4CR(this.Context, 0, CRConst.Key\_InvestRatePrecision, 2);  }  public override void FormatCellValue(FormatCellValueArgs args)  {  base.FormatCellValue(args);  switch(args.Header.FieldName.ToUpperInvariant())  {  case "FRATE":  case "FHISTORYRATE":  if(args.Value == null) break;  args.FormateValue = Convert.ToDecimal(args.Value).ToString(FormartString);  break;  }  }  }  } | |

## OnFormatRowConditions

**触发时机**

列表显示，在获取数据之后，数据发送到前端之前触发；在FormatCellValue事件之后，设计时配置的列表条件格式化生效之前触发。对列表行数据进行格式化

**应用场景**

列表的行数据，不同的行个性显示，譬如奇偶行分色显示等等

**关键字**

格式化

**备注**

视图层插件

### 案例 –暂缺

## ListCreateColumns

**触发时机**

列表创建列数据之后，数据发送到前端之前触发

**应用场景**

列表创建列信息之后，插件可以修改列信息

**关键字**

无

**备注**

视图层插件事件

### 案例 –暂缺

## ListInitialize

**触发时机**

列表初始化时触发

**应用场景**

初始化时获取参数

**关键字**

初始化

**备注**

无

### 案例 –调整分录模板列表初始化时获取用户参数

**代码来源**

财务——合并报表——调整分录模板列表插件：

Kingdee.K3.FIN.CR.Business.PlugIn.RptAdjTempList,

Kingdee.K3.FIN.CR.Business.PlugIn

**需求背景**

调整分录表单界面，点击菜单【引入分录模板】，打开调整分录模板列表时，会添加一个自定义参数“OpenByAdjust”，标识来自调整分录表单。而在调整分录模板列表界面，需要获取这个参数，在操作之前判断这个参数，如果参数为true且操作是返回数据，则走返回调整分录模板的逻辑

**实现方案**

调整分录表单插件，ShowForm的时候，添加自定义参数“OpenByAdjust”。调整分录模板列表界面，捕获ListInitialize事件，获取参数。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS.Core.List.PlugIn;  using System;  using System.ComponentModel;  namespace Kingdee.K3.FIN.CR.Business.PlugIn  {  /// <summary>  /// 调整分录模板列表插件  /// </summary>  [Description("调整分录模板列表插件")]  public class RptAdjTempList : AbstractListPlugIn  {  private bool blnOpenByAdjust;  public override void ListInitialize(ListInitializeEventArgs e)  {  base.ListInitialize(e);  blnOpenByAdjust = Convert.ToBoolean(e.Paramter.GetCustomParameter("OpenByAdjust"));  }  }  } | |

## ListRowDoubleClick

**触发时机**

双击列表行时触发。如果插件未撤销后续操作，会继续调用内置的行双击处理函数。

**应用场景**

双击列表时，打开单据详情、穿透明细列表、返回数据

**关键字**

列表双击

**备注**

插件可以通过设置参数ListRowDoubleClickArgs的Cancel属性，撤销后续操作。如果没有撤销，则会调用内置的行双击处理函数：如果是查找列表，则返回数据；如果是普通列表，则调用列表配置的行双击事件。

### 案例 –采购价目表列表界面双击

**代码来源**

供应链采购价目表列表插件

Kingdee.K3.SCM.Purchase.Business.PlugIn.PriceListList,

Kingdee.K3.SCM.Business.PlugIn

**需求背景**

采购价目表列表界面，如果是普通查看列表界面，双击时进入价目表明细；如果是采购订单（或其他采购类单据）在查询价格的时候进入，则双击不要做任何操作。

**实现方案**

采购订单在点击采购明细表体的菜单项【查询价格】时，会调用【查询价格】操作，在这个操作中，通过ShowForm的方式显示采购价目表列表，并添加了一个自定义参数“IsPriceListQuery”标识是在进行价格查询。在采购价目表的列表插件中，捕获重写ListRowDoubleClick事件，获取并判断自定义参数“IsPriceListQuery”，如果有值则撤销后续操作；如果没有值，则继续调用BOS平台的列表双击处理函数。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.List.PlugIn.Args;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Kingdee.K3.SCM.Purchase.Business.PlugIn  {  public class PriceListList : AbstractListPlugIn  {  public override void ListRowDoubleClick(ListRowDoubleClickArgs e)  {  object isPriceListQuery = this.View.OpenParameter.GetCustomParameter("IsPriceListQuery");  if (isPriceListQuery != null)  {  e.Cancel = true;  }  }  }  } | |

## OnGetConvertRule

**触发时机**

选单/下推过程，获取单据转换规则时触发

**应用场景**

弃用操作框里指定的转换规则，通过插件来获取新的单据转换规则

**关键字**

选单、下推、单据转换

**备注**

视图层插件

选单/下推过程中，确定单据转换规则

### 案例 –简单生产入库下推简单生产领料订单

**代码来源**

供应链简单生产入库列表插件

Kingdee.K3.SCM.Stock.Business.PlugIn.SP.SpInStockList,

Kingdee.K3.SCM.Stock.Business.PlugIn

**需求背景**

简单生产入库在下推简单生产领料单时，需要使用BOM正向展开下推简单生产领料单的转换规则进行下推。

**实现方案**

插件捕获OnGetConvertRule事件，判断下推目标单是简单生产领料，则获取BOM正向展开下推简单生产领料单的转换规则，调整转换规则的相关属性，赋值到事件参数中。

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS;  using Kingdee.BOS.Core.DynamicForm;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.List.PlugIn.Args;  using Kingdee.BOS.Core.Metadata.ConvertElement;  using Kingdee.BOS.ServiceHelper;  using System.Collections.Generic;  using System.ComponentModel;  using System.Linq;  namespace Kingdee.K3.SCM.Stock.Business.PlugIn.SP  {  [Description("简单生产入库单-列表插件")]  public class SpInStockList : AbstractListPlugIn  {  public override void OnGetConvertRule(GetConvertRuleEventArgs e)  {  base.OnGetConvertRule(e);  if (e.ConvertOperation == FormOperationEnum.Push  && (e.TargetFormId == "SP\_PickMtrl"))  {  ConvertRuleElement rule = ConvertServiceHelper.GetConvertRules(  this.Context,"ENG\_BomExpandBill",e.TargetFormId).FirstOrDefault();  if (rule != null)  {  ConvertFilterPolicyElement filterPolicy = rule.Policies.Where(p => p is ConvertFilterPolicyElement)  .Select(p => p as ConvertFilterPolicyElement).FirstOrDefault();  filterPolicy.AlertMessage = new LocaleValue("1.用户需有入库物料BOM上发料组织下简单生产领料单的新增权限。\r\n2.简单生产入库单领料标识应该为“否”。",  this.Context.LogLocale.LCID);  List<DefaultConvertPolicyElement> ruleFieldsMap = rule.Policies.Where(w => w is DefaultConvertPolicyElement).  Select(w => w as DefaultConvertPolicyElement).ToList();  var fRuleMap = ruleFieldsMap.FirstOrDefault();  if (fRuleMap != null)  {  KeyValuePair<string, string>[] fieldMaps = new KeyValuePair<string, string>[]  {  new KeyValuePair<string, string>("FStockOrgId", "FSupplyOrgId"), // 物料清单分录的供应组织  new KeyValuePair<string, string>("FPrdOrgId", "fprdorgid\_reg"),  new KeyValuePair<string, string>("FWorkShopId", "fworkshopid\_reg"),  new KeyValuePair<string, string>("FOwnerTypeId0", "fownertypeid\_reg"),  new KeyValuePair<string, string>("FOwnerId0", "fownerid\_reg"),  new KeyValuePair<string, string>("FOwnerTypeId", "fownertypeid\_reg"),  new KeyValuePair<string, string>("FOwnerId", "fownerid\_reg"),  new KeyValuePair<string, string>("FKeeperId", "FSupplyOrgId"), // 物料清单分录的供应组织  new KeyValuePair<string, string>("FParentOwnerTypeId", "fownertypeid\_reg"),  new KeyValuePair<string, string>("FParentOwnerId", "fownerid\_reg"),  };  foreach (var keyValuePair in fieldMaps)  {  var fieldMapElement = fRuleMap.FieldMaps.FirstOrDefault(w => w.TargetFieldKey == keyValuePair.Key);  if (fieldMapElement != null)  {  fieldMapElement.SourceFieldKey = keyValuePair.Value;  }  }  }  e.Rule = rule;  }  }  }  }  } | |
|  | |

## OnShowConvertOpForm

**触发时机**

下推过程中，展示可选目标单据

**应用场景**

捕获此事件，调整可选的单据类型

**关键字**

下推

**备注**

无

### 案例 –采购收料单列表界面下推筛选掉资产卡片

**代码来源**

采购收料单列表插件

Kingdee.K3.SCM.Purchase.Business.PlugIn.PurchaseReceiveList,

Kingdee.K3.SCM.Purchase.Business.PlugIn

**需求背景**

采购收料单下推过程，可选的单据列表需要筛选掉资产卡片

**实现方案**

捕获OnShowConvertOpForm事件，修改事件参数的Bills属性，筛选掉资产卡片

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS.Core.DynamicForm;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.Metadata.ConvertElement;  using Kingdee.BOS.Util;  using System;  using System.Collections.Generic;  using System.Linq;  using Kingdee.BOS.Core.List.PlugIn.Args  namespace Kingdee.K3.SCM.Purchase.Business.PlugIn  {  public class PurchaseReceiveList : AbstractListPlugIn  {  public override void OnShowConvertOpForm(ShowConvertOpFormEventArgs e)  {  //下推操作，用来筛选掉资产卡片  if (e.ConvertOperation == FormOperationEnum.Push)  {  var bills = e.Bills as List<ConvertBillElement>;  if (bills != null && bills.Count > 0)  {  e.Bills = bills.Where(c => !c.FormID.Equals("FA\_CARD", StringComparison.OrdinalIgnoreCase)).ToList();  }  }  }  }  } | |

## OnShowTrackResult

**触发时机**

上/下查过程，获得上/下查结果

**应用场景**

插件捕获此事件，调整上/下查结果，调整结果显示

**关键字**

上查、下查

**备注**

无

### 案例 –生产领料单上查替换物料清单为生产订单

**代码来源**

生产领料单列表插件

Kingdee.K3.MFG.PRD.Business.PlugIn.Bill.PickMtrlList,

Kingdee.K3.MFG.PRD.Business.PlugIn

**需求背景**

在生产领料单列表界面，上查的时候，需要将上查结果中的物料清单列表替换成生产订单列表

**实现方案**

生产领料单的列表插件捕获OnShowTrackResult事件，修改上查结果

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using Kingdee.BOS.BusinessEntity.BillTrack;  using Kingdee.BOS.Core.DynamicForm;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.List.PlugIn.Args;  using Kingdee.BOS.Orm.DataEntity;  using Kingdee.BOS.Util;  using Kingdee.K3.MFG.ServiceHelper.PRD;  using System;  using System.Collections.Generic;  using System.Linq;  namespace Kingdee.K3.MFG.PRD.Business.PlugIn.Bill  {  /// <summary>  /// 生产领料单序时薄插件  /// </summary>  public class PickMtrlList : AbstractListPlugIn  {  public override void OnShowTrackResult(ShowTrackResultEventArgs e)  {  base.OnShowTrackResult(e);  switch (e.TrackOperation)  {  case FormOperationEnum.TrackUp: //上查  if (e.TrackResult != null)  {  BillNode trackResult = e.TrackResult as BillNode;  if (!trackResult.FormKey.Equals(e.TargetFormKey, StringComparison.OrdinalIgnoreCase))  {  e.TrackResult = e.TargetFormKey.EqualsIgnoreCase("PRD\_PREPAREMTRL") ?  GetReplacedTrackResultForMoveMtrl(trackResult, e.TargetFormKey) : GetReplacedTrackResult(trackResult, e.TargetFormKey);  }  }  break;  }  }  /// <summary>  /// 替换上下查结果,把用料清单列表替换成生产挪料单列表  /// </summary>  /// <param name="trackResult"></param>  /// <param name="targetFormKey"></param>  /// <returns></returns>  private BillNode GetReplacedTrackResultForMoveMtrl(  BillNode trackResult, string targetFormKey)  {  if (trackResult.FormKey.Equals("PRD\_PPBOM", StringComparison.OrdinalIgnoreCase)  && targetFormKey.Equals("PRD\_PREPAREMTRL", StringComparison.OrdinalIgnoreCase))  {  List<long> moveMtrlEntryIds = MoveMtrlServiceHelper.GetMoEntryIdsByPPBomEntryId(this.View.Context, trackResult.LinkIds.Select(o => Convert.ToInt64(o)).ToList());  trackResult = BillNode.Create("PRD\_PREPAREMTRL", "");  trackResult.AddLinkCopyData(moveMtrlEntryIds.Select(o => o.ToString()).ToList());  return trackResult;  }  else  {  return trackResult;  }  }  /// <summary>  /// 替换上下查结果,把用料清单列表替换成生产订单列表  /// </summary>  /// <param name="trackResult"></param>  /// <param name="targetFormKey"></param>  /// <returns></returns>  private BillNode GetReplacedTrackResult(  BillNode trackResult, string targetFormKey)  {  if (trackResult.FormKey.Equals("PRD\_PPBOM", StringComparison.OrdinalIgnoreCase)  && targetFormKey.Equals("PRD\_MO", StringComparison.OrdinalIgnoreCase))  {  DynamicObjectCollection moDataEntitys = MOServiceHelper.GetMoEntityIdsByPPBomEntityId(this.View.Context, trackResult.LinkIds.Select(o => Convert.ToInt64(o)).ToList());  trackResult = BillNode.Create("PRD\_MO", "");  trackResult.AddLinkCopyData(moDataEntitys.Select(o => o["FMOENTRYID"].ToString()).ToList());  return trackResult;  }  else  {  return trackResult;  }  }  }  } | |

## PrepareFilterParameter

**触发时机**

打开单据列表，构建取数用的SqlBuilder对象前触发

**应用场景**

打开单据列表之前，插件直接修改过滤条件及排序子句等内容，筛选、处理数据，列表显示的时候显示这些被筛选、处理过的数据

**关键字**

修改列表过滤条件、排序子句

**备注**

模型层插件事件

通过这个事件修改的过滤条件在所有列表都生效，且在过滤界面修改过滤条件、排序字段等内容时不会影响到此事件的修改内容

### 案例 –信用检查规则查找列表数据筛选

**代码来源**

供应链——信用检查规则列表插件

Kingdee.K3.SCM.Credit.Business.PlugIn.CheckRuleList,

Kingdee.K3.SCM.Credit.Business.PlugIn

**需求背景**

信用档案在【检查规则】字段上F7打开信用检查规则列表时，需要列表仅显示列表可见的检查规则。

**实现方案**

插件捕获PrepareFilterParameter事件，判断父页面业务对象是信用档案，则添加额外的过滤条件flistdisable<>'1'

**示例代码**

|  |  |
| --- | --- |
| C# |  |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using Kingdee.BOS.Core.Bill.PlugIn;  using Kingdee.BOS.Core.List.PlugIn;  using Kingdee.BOS.Core.List.PlugIn.Args;  namespace Kingdee.K3.SCM.Credit.Business.PlugIn  {  public class CheckRuleList : AbstractListPlugIn  {  public override void PrepareFilterParameter(FilterArgs e)  {  if (this.View.ParentFormView.UserParameterKey != "CRE\_CustArchives")  {  if (string.IsNullOrWhiteSpace(e.FilterString) == false)  {  e.FilterString += " and";  }  e.FilterString += " flistdisable<>'1' ";  }  }  }  } | |

## PrepareFuncPermissionDataRule

已废弃，不会触发