

当表或列名更新时同步引用

- 1. 描述
- 2. 同步引用



1. 描述

当表或列名更新时，引用表或列的插件的属性值应该一起重命名。因此，插件需要实现IReferenceTable接口，不仅可以同步重命名属性，而且可以找到表或列的所有引用。



2. 同步引用

下面的示例中，展示了如何使用这个接口。

```
public class MyListCellType: CellType, IReferenceTable
{
    public string TableName
    {
        get; set;
    }
    public string TextColumn
    {
        get; set;
    }

    public object QueryCondition
    {
        get; set;
    }
    public object SortCondition
    {
        get; set;
    }

    public ComplexProperty ComplexProperty
    {
        get; set;
    }
    /// <summary>
    ///
    /// QueryCondition generated using builtin query condition dialog
    has implemented the interface 'IReferenceTable';
    /// SortCondition generated using builtin sort condition dialog has
    implemented the interface 'IReferenceTable';
    /// </summary>
    /// <param name="location"></param>
    /// <returns></returns>
    public IEnumerable<LocatedObject<TableCheckInfo>>
    GetTableInfo(LocationIndicator location)
    {
        var result = new List<LocatedObject<TableCheckInfo>>();
```

```

        TableCheckInfo tableInfo = new TableCheckInfo(this.TableName);
        tableInfo.AddColumns(this.TextColumn);
        //Add direct table info
        result.Add(new LocatedObject<TableCheckInfo>(tableInfo,
location.AppendProperty("TableName")));
        //Add QueryCondition table info
        if (this.QueryCondition is IReferenceTable)
        {
            result.AddRange((this.QueryCondition as
IReferenceTable).GetTableInfo(location.AppendProperty("QueryCondition"))
);
        }
        //Add SortCondition table info
        if (this.SortCondition is IReferenceTable)
        {
            result.AddRange((this.SortCondition as
IReferenceTable).GetTableInfo(location.AppendProperty("SortCondition")))
;
        }
        //Add sub-property table info
        if (this.ComplexProperty != null)
        {
            //Add releted table info
            TableCheckInfo relatedTableInfo = new
TableCheckInfo(this.ComplexProperty.ReletedTableName);
            result.Add(new
LocatedObject<TableCheckInfo>(relatedTableInfo,
location.AppendProperty("ComplexProperty").AppendProperty("ReletedTableN
ame")));
            //Add SortCondition releted table info
            if (this.ComplexProperty.SortCondition is IReferenceTable)
            {
                result.AddRange((this.ComplexProperty.SortCondition as
IReferenceTable).GetTableInfo(location.AppendProperty("ComplexProperty")
.AppendProperty("SortCondition")));
            }
        }
        return result;
    }
    /// <summary>
    ///
    /// </summary>
    /// <param name="tableName"></param>
    /// <param name="oldName"></param>
    /// <param name="newName"></param>
    public void RenameTableColumnName(string tableName, string oldName,
string newName)
    {
        if (string.Equals(this.TableName, tableName))
        {
            if (string.Equals(this.TextColumn, oldName))
            {
                this.TextColumn = newName;
            }
        }
        if (this.QueryCondition is IReferenceTable)
        {

```

```

        (this.QueryCondition as
IReferenceTable).RenameTableColumnName(tableName, oldName, newName);
    }
    if (this.SortCondition is IReferenceTable)
    {
        (this.SortCondition as
IReferenceTable).RenameTableColumnName(tableName, oldName, newName);
    }
    if (this.ComplexProperty != null)
    {
        if (this.ComplexProperty.SortCondition is IReferenceTable)
        {
            (this.ComplexProperty.SortCondition as
IReferenceTable).RenameTableColumnName(tableName, oldName, newName);
        }
    }
}
/// <summary>
///
/// </summary>
/// <param name="oldName"></param>
/// <param name="newName"></param>
public void RenameTableName(string oldName, string newName)
{
    if (string.Equals(this.TableName, oldName))
    {
        this.TableName = newName;
    }
    if (this.QueryCondition is IReferenceTable)
    {
        (this.QueryCondition as
IReferenceTable).RenameTableName(oldName, newName);
    }
    if (this.SortCondition is IReferenceTable)
    {
        (this.SortCondition as
IReferenceTable).RenameTableName(oldName, newName);
    }
    if (this.ComplexProperty != null)
    {
        if (string.Equals(this.ComplexProperty.RelatedTableName,
oldName))
        {
            this.ComplexProperty.RelatedTableName = newName;
        }
        if (this.ComplexProperty.SortCondition is IReferenceTable)
        {
            (this.ComplexProperty.SortCondition as
IReferenceTable).RenameTableName(oldName, newName);
        }
    }
}
}

public class ComplexProperty
{
    public string RelatedTableName
    {

```

```
        get; set;
    }
    public object SortCondition
    {
```

```
    get; set;
}
}
```

[回到顶部](#)