

企业如果上了ERP系统，比如SAP、用友、金蝶或者E10等等，只需要ERP里面提供相应的接口，则可以直接将PMC创建的工单信息抛转至 MTS 系统，当该工单生产完成之后，MTS 将完成数据回传至 ERP，方便货仓进行入仓与扣数，这篇文章介绍 MTS 与 SAP 之间如何对接。

1.SAP里面必须有一个Function，类似 MSSQL里面的存储过程一样，根据用户传入的参数，返回所需要的数据，这一部分程序通过 ABAP 完成，不在本文的讲述范围之类，这里略过。

2.MTS连接到 SAP，Call对应的 Function，拿到数据后进行处理，这里需要准备以下几个参数。

- SAP服务器的IP地址
- 访问SAP的用户名
- 访问SAP的密码
- SAP系统的实例编号
- SAP系统的客户端编号

为了方便多个地方调用，这里将Call SAP 的代码写成一个方法，避免重复代码，代码如下：

```
using SAP.Middleware.Connector;

public class SapHelper

{

    public IRfcTable GetSapData(string FunctionName, string TableName,
List<string> Key, List<string> Value)

    {

        string Server = ConfigurationManager.AppSettings["SapServer"];

        string Client = ConfigurationManager.AppSettings["SapClient"];

        string User = ConfigurationManager.AppSettings["SapUser"];

        string Password = ConfigurationManager.AppSettings["SapPwd"];
```

```
string SystemNumber =  
ConfigurationManager.AppSettings["SapSystemNumber"];  
  
RfcConfigParameters rfcPar = new RfcConfigParameters();  
  
rfcPar.Add(RfcConfigParameters.Name, "CON");  
  
rfcPar.Add(RfcConfigParameters.AppServerHost, Server);  
  
rfcPar.Add(RfcConfigParameters.Client, Client);  
  
rfcPar.Add(RfcConfigParameters.User, User);  
  
rfcPar.Add(RfcConfigParameters.Password, Password);  
  
rfcPar.Add(RfcConfigParameters.SystemNumber, SystemNumber);  
  
rfcPar.Add(RfcConfigParameters.Language, "EN");  
  
RfcDestination dest = RfcDestinationManager.GetDestination(rfcPar);  
  
RfcRepository rfcrep = dest.Repository;  
  
IRfcFunction myfun = null;  
  
myfun = rfcrep.CreateFunction(FunctionName);  
  
IRfcTable IrfTable = myfun.GetTable(TableName);  
  
if (Key.Count != Value.Count)  
{  
  
    return null;  
  
}  
  
for (int i = 0; i < Key.Count; i++)
```

```
{  
  
    myfun.SetValue(Key[i].ToString(), Value[i].ToString());  
  
}  
  
myfun.Invoke(dest);  
  
return IrfTable;  
  
}  
  
}
```

当用户输入需要下载的工单号码之后，系统连接到 SAP 开始调用函数：

```
string Order = Request["LotName"];  
  
if (string.IsNullOrEmpty(Order))  
  
{  
  
    return;  
  
}
```

//这里强烈建议将参数以key value的形式传进去，方便很多。

```
List<string> Key1 = new List<string>();  
  
List<string> Value1 = new List<string>();  
  
  
Key1.Insert(0, "ORDER_NUMBER");  
  
Value1.Insert(0, Order);
```

```
IRfcTable IrfTable_Order = new SapHelper().GetSapData("function", "table",  
Key1, Value1);
```

```
DataTable dt_Order = new DataTable();
```

```
dt_Order.Columns.Add("LotName");
```

```
dt_Order.Columns.Add("ProductName");
```

```
dt_Order.Columns.Add("LotQuantity");
```

```
dt_Order.Columns.Add("Customer");
```

```
dt_Order.Columns.Add("Plant");
```

```
for (int i = 0; i < IrfTable_Order.Count; i++)
```

```
{
```

```
IrfTable_Order.CurrentIndex = i;
```

```
DataRow dr = dt_Order.NewRow();
```

```
dr["LotName"] = IrfTable_Order.GetString("AUFNR");
```

```
dr["ProductName"] = IrfTable_Order.GetString("STLBEZ");
```

```
dr["LotQuantity"] =  
Math.Round(Convert.ToDouble(IrfTable_Order.GetString("GAMNG")),  
0).ToString();
```

```
dr["Customer"] = IrfTable_Order.GetString("PSPEL");
```

```
dr["Plant"] = IrfTable_Order.GetString("WERKS");
```

```
dt_Order.Rows.Add(dr);
```

```
}  
  
var json = Newtonsoft.Json.JsonConvert.SerializeObject(dt_Order);  
  
var totalRecordCount = dt_Order.Rows.Count.ToString();  
  
json = "{\"total\":\"" + totalRecordCount + "\",\"rows\":\"" + json + "\"}";  
  
Response.Write(json);  
  
Response.End();
```

注意，从SAP里面得到的数据是一张 rfc table，并不是我们常用的 DataTable，所以这里需要进行转换后才能使用，最后将数据转换为 JSON 输出即可：

