

LMES二次开发入门

提 纲

1. 快速入门
2. 开发模版
3. 定制业务
4. 打包发布

一、快速入门

快速开始

CMS

为OT工程师量身打造

新一代IT和OT融合的生产力工具



快速开始

参考文档

<https://cms-docs.shengyc.com/cms/>



使用指南



开发文档



API参考

软件安装

1、安装CMS基座（选择最新版本）：

<https://list.sync-cms.com/CMS/%E4%BA%A7%E5%93%81/CMS2.0>

注意：为了后续方便开发调试，不要修改CMS默认安装路径：C:\Program Files\CMS Editor

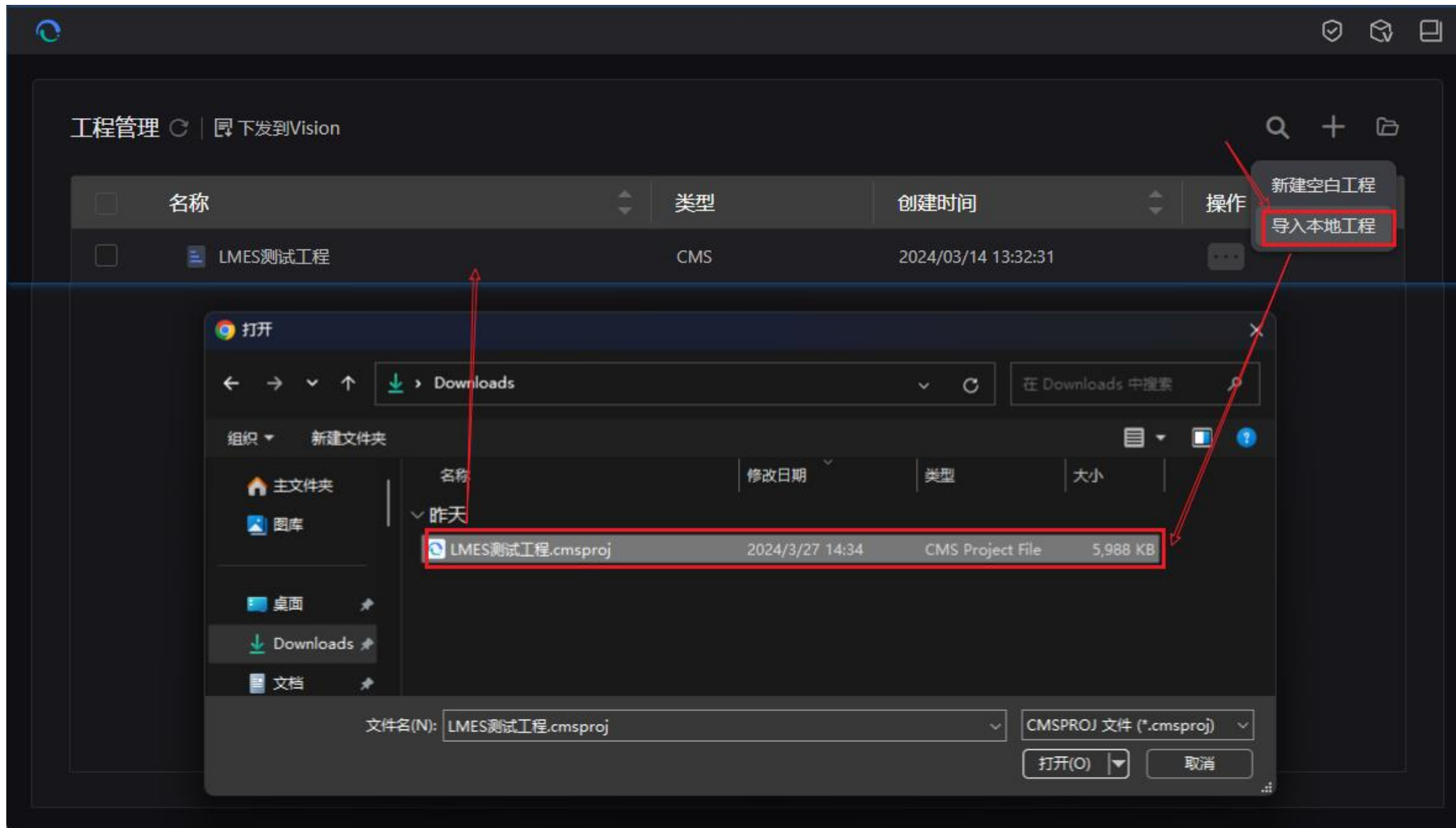
2、安装LMES插件（选择最新版本）：

<https://list.sync-cms.com/CMS/%E7%BB%84%E4%BB%B6/LMES>



导入工程

下载工程：《[LMES测试工程.cmsproj](#)》并导入到CMS



进入工程

The screenshot displays the LMES Test Engineering software interface. The main window is titled "LMES测试工程" and features a dark-themed dashboard for production monitoring. The interface includes a top toolbar with various icons for navigation and editing, a left sidebar with a project tree, and a right sidebar for styling and events.

生产监控 (Production Monitoring) Dashboard:

- 产线调试** (Production Line Debugging)
- 启动模拟数据** (Start Simulation Data)
- 启停信号** (Start/Stop Signal)
- 启动** (Start)
- 停止** (Stop)
- 初始化合格信号** (Initialize合格Signal)
- 加工类型** (Processing Type)
- 停止模拟数据** (Stop Simulation Data)
- 生成条码** (Generate Barcode)
- 触发** (Trigger)
- 清空** (Clear)
- 初始化测试数据** (Initialize Test Data)
- 进站结果** (In-station Result)

工序管理 (Process Management) Table:

工序1	工序2	工序3
产品ID	产品ID	产品ID
进站信号	进站信号	进站信号
加工类型	加工类型	加工类型
进站结果	进站结果	进站结果
进站完成	进站完成	进站完成
进站说明	进站说明	进站说明

运行工程




按钮功能:

- **初始化合格信号**: 下发PLC合格判断信号 **1**。
- **初始化测试数据**: 一键初始化, 条码, 产品, 物料, 工序, 配方, 工单, 系统配置等数据。
- **启动/停止模拟数据**: 每10秒自动模拟下发 **工序采集参数, 工序配方参数, 工序物料检测参数**。
- **启动/停止**: 启停PLC模拟流程, 模拟PLC下发进站信号, 完工信号, 完成进出站流程。
- **触发/清空**: 触发生成码流程, 按照条码规则生成条码, 清空条码。
- **启用调试模式**: 启用后支持工单下发后修改工序、配方等设置。

流程调试

步骤:

1. 初始化测试数据（表数据为空时才初始化）
2. 启动所有服务
3. 初始化合格信号（用于判断合格过站）
4. 启动模拟数据（用于过程参数采集）
5. 使用  按钮，自动运行流程，或手动触发进站信号调试。

流程日志

使用[Notepad++](#)通过流程实例ID,快速查看流程执行情况

```
FlowManagement.Services.FlowInstanceService, (=====启动进出站业务流程: 实例=2403281711608453178, PLA_工序1_PLA_工位1_10001_1, 版本=10001_1=====)
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号监听
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号监听 监听信号=InStationSignal1, CompareValue=1, IsProcessed=False, CanExit=0
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号监听 InStationSignal1 变量值发生变化, 旧值=>新值True, TraceId=
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号监听
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号监听 监听信号=InStationSignal1, CompareValue=1, IsProcessed=True, CanExit=1
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号监听 耗时=0.0025毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 读取产品码
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 读取产品码 ProductID1=20240328_0000000158, TraceId=4bc194f8-8022-4240-3281-711608453178
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 读取产品码 耗时=2.0917毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站初始化
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站初始化 耗时=4.263毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 判断加工类型
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 判断加工类型 返修加工判断, 耗时=36毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 判断加工类型 正常加工, 产品型号=默认产品, 工单号=Order_1
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 判断加工类型 正常加工判断, 耗时=19毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 判断加工类型 耗时=57.3081毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 正常加工
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 正常加工 耗时=0.0544毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站校验
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站校验 耗时=0.014毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 产品状态校验
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 产品状态校验 采集物料检测参数 BindMaterialVariable TraceId=c86463cb-ea7f-4240-3281-711608453178
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 产品状态校验 校验结果: True, 返修标记: False
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 产品状态校验 耗时=7.4554毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 重码校验
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 重码校验 校验结果: True
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 重码校验 耗时=4.7078毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 漏工序校验
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 漏工序校验 校验结果: True
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 漏工序校验 耗时=4.892毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站结果下发
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站结果下发 下发信号 ProcessType1=1,AbnormalCause1=,InStationResult1=1,CompletedSignal1=1, TraceId=b648bc1c-b761-4240-3281-711608453178
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站结果下发 下发校验 CompletedSignal1=1,InStationResult1=1,AbnormalCause1=,ProcessType1=1, TraceId=79cd3a00-b2dd-4240-3281-711608453178
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站结果下发 发布流程事件 EventSynchronous=False --Begin
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站结果下发 发布流程事件 EventSynchronous=False --end
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站结果下发 耗时=0.3803毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号复位监听
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号复位监听 监听信号=InStationSignal1, CompareValue=0, IsProcessed=False, CanExit=0
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号复位监听 InStationSignal1 变量值发生变化, 旧值True=>新值False, TraceId=
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号复位监听
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号复位监听 监听信号=InStationSignal1, CompareValue=0, IsProcessed=True, CanExit=1
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站信号复位监听 耗时=0.0023毫秒
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站完成
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站完成 下发信号 ProcessType1=0,AbnormalCause1=,InStationResult1=0,CompletedSignal1=0, TraceId=ba7542fc-af2e-4240-3281-711608453178
Processflow, 执行流程: 实例=2403281711608453178 -> PLA_工序1_PLA_工位1_10001_1 -> 进站完成 下发校验 CompletedSignal1=0,InStationResult1=0,AbnormalCause1=,ProcessType1=0, TraceId=e56014df-ab77-4240-3281-711608453178
```

二、开发模板

开发环境

后端: [.NET](#) + [Visual Studio 2022](#)

前端: [Node.js](#) + [Visual Studio Code](#)

插件模板

安装模板:

```
dotnet new install CMS.Plugin.Template.Ddd
```

卸载模板:

```
dotnet new uninstall CMS.Plugin.Template.Ddd
```

创建插件: 创建一个名为“MyPluginName”的插件解决方案, 并添加名为“MyEntityName”实体代码

```
dotnet new cms-ddd -n MyPluginName -e MyEntityName
```

选项:

-n: 插件的名称

-e: 实体的名称

实体模板

安装模板:

```
dotnet new install CMS.Plugin.Template.Ddd.Entity
```

卸载模板:

```
dotnet new uninstall CMS.Plugin.Template.Ddd.Entity
```

创建插件: 为名为“MyPluginName”的插件解决方案, 添加名为“MyEntityName”实体代码

```
dotnet new cms-ddd-entity -n MyPluginName -e MyEntityName
```

选项:

-n: 插件的名称

-e: 实体的名称

创建插件

The image shows a Windows PowerShell terminal window and a File Explorer window. The PowerShell window displays the following commands and output:

```
PS D:\> dotnet new install CMS.Plugin.Template.Ddd --nuget-source https://nexus.sycdev.com/repository/nuget-group/index.json
将安装以下模板包:
  CMS.Plugin.Template.Ddd

已安装 CMS.Plugin.Template.Ddd (版本 2.8.7), 它将替换为 最新版本。
已成功卸载 CMS.Plugin.Template.Ddd::2.8.7。
成功: CMS.Plugin.Template.Ddd::2.8.7 已安装以下模板:
模板名                短名称  语言  标记
-----
CMS.Plugin.Template.Ddd cms-ddd [C#]  Web/WebAPI

PS D:\> dotnet new cms-ddd -n DemoManagement -e Demo
已成功创建模板“CMS.Plugin.Template.Ddd”。

PS D:\>
```

The File Explorer window shows the directory structure of the newly created project:

名称	修改日期	类型	大小	说明
server	2024/3/28 15:57	文件夹		后端代码
web	2024/3/28 15:57	文件夹		前端代码
build.ps1	2024/3/28 15:57	PowerShell 源文件	1 KB	构建脚本

检查输出文件

The image displays two screenshots of a Windows File Explorer window, showing the file paths and contents for a CMS plugin.

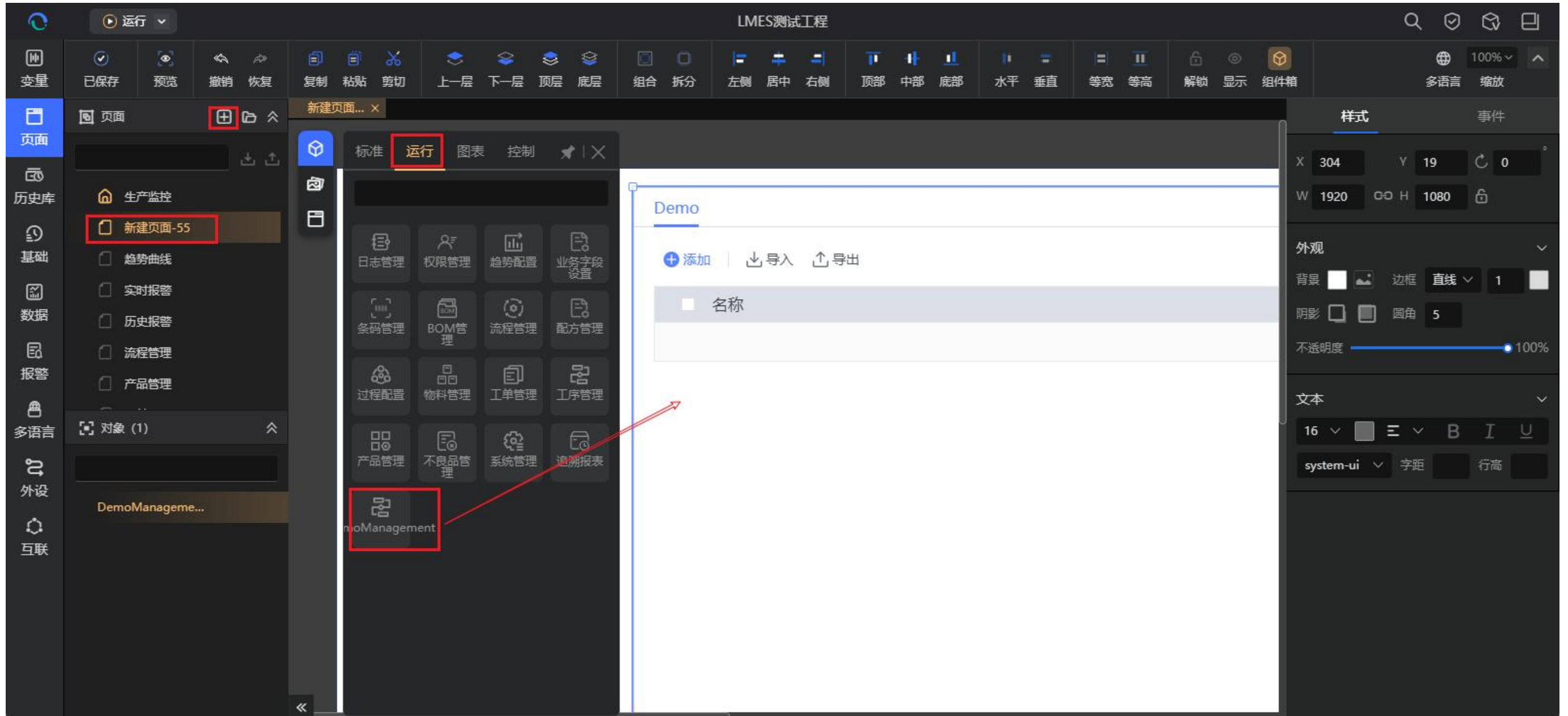
The top screenshot shows the path: 此电脑 > Windows (C:) > Program Files > CMS Editor > plugins > cms.plugin.demomanagement > net6.0. The file list contains the following items:

名称	修改日期	类型	大小
CMS.Plugin.DemoManagement.deps....	2024/3/28 16:00	JSON 源文件	363 KB
CMS.Plugin.DemoManagement.dll	2024/3/28 16:00	应用程序扩展	35 KB
CMS.Plugin.DemoManagement.pdb	2024/3/28 16:00	VisualStudio.pd...	36 KB
CMS.Plugin.DemoManagement.runti...	2024/3/28 16:00	JSON 源文件	1 KB
CMS.Plugin.DemoManagement.xml	2024/3/28 16:00	Microsoft Edge ...	10 KB

The bottom screenshot shows the path: 此电脑 > Windows (C:) > Program Files > CMS Editor > host > wwwroot > widgets > DemoManagement. The file list contains the following item:

名称	修改日期	类型	大小
index.js	2024/3/28 16:01	JavaScript 源文件	686 KB

重启CMS



初始化数据

生产监控 追溯报表 工序管理 工单管理 流程管理 产品管理 条码管理 配方管理 质量管理 物料管理 BOM管理 系统配置 过程配置

产线调试 启动模拟数据 启停信号 启动 停止 初始化合格信号 加工类型: 0=复位状态, 1=正常加工, 2=返修加工, 3=点检加工 物料
停止模拟数据 生成条码 触发 清空 初始化测试数据 进站结果: 0=复位状态, 1=正常, 2=产品状态异常, 3=重码, 4=漏工序

Demo

+ 添加 | ↓ 导入 | ↑ 导出

名称	编号	备注
1 Demo_Name1	Demo_Code1	Demo_Remark1
2 Demo_Name2	Demo_Code2	Demo_Remark2
3 Demo_Name3	Demo_Code3	Demo_Remark3
4 Demo_Name4	Demo_Code4	Demo_Remark4
5 Demo_Name5	Demo_Code5	Demo_Remark5

三、定制业务

新增流程

[生产监控](#)[追溯报表](#)[工序管理](#)[工单管理](#)[流程管理](#)[产品管理](#)[条码管理](#)[配方管理](#)[质量管理](#)[物料管理](#)[BOM管理](#)[系统配置](#)[过](#)

流程列表

筛选

序号	流程名称	交互类型	功能项	关联工序	
1	普通工序流程	批量配置	进出站交互	产品条码的获取方式, 产品加工状态的判断...	PLA_工序1 PLA_工序10 PLA_工序1_副本 +1
2	普通工序进站流程	批量配置	进站交互	进站条码获取方式, 进站时的产品状态检测...	PLA_工序2 PLA_工序3 PLA_工序4 +38
3	普通工序出站流程	批量配置	出站交互	出站条码获取方式, 产品加工状态的判断方...	PLA_工序2 PLA_工序3 PLA_工序4 +35
4	生成码流程	批量配置	其他交互	依据生成规则生成条码	PLA_工序1 PLA_工序1_副本 PLA_工序1_副本_副本
5	物料检测流程	批量配置	其他交互	当前工序的装配物料检测	PLA_工序10
6	PLC模拟流程	批量配置	其他交互	-	PLA_工序1 PLA_工序2 PLA_工序3 +39
7	普通工序流程 (PLC触发 - LMES复位)	批量配置	进出站交互	产品条码的获取方式, 产品加工状态的判断...	-
8	普通工序进站流程 (PLC触发 - LMES复位)	批量配置	进站交互	进站条码获取方式, 进站时的产品状态检测...	-
9	普通工序出站流程 (PLC触发 - LMES复位)	批量配置	出站交互	出站条码获取方式, 产品加工状态的判断方...	-

预览流程

生产监控 追溯报表 工序管理 工单管理 **流程管理** 物料管理 BOM管理 系统配置 过程配置

流程列表

筛选

序号	流程名称	交互类型	操作
1	普通工序流程	批量配置	查看
2	普通工序进站流程	批量配置	查看
3	普通工序出站流程	批量配置	查看
4	生成码流程	批量配置	查看
5	物料检测流程	批量配置	查看
6	普通工序流程 (PLC触发 - LMES复位)	批量配置	查看
7	普通工序进站流程 (PLC触发 - LMES复位)	批量配置	查看
8	普通工序出站流程 (PLC触发 - LMES复位)	批量配置	查看
9	PLC模拟流程	批量配置	查看
10	Demo测试流程	批量配置	查看

Demo测试流程

```
graph TD; Start[启动] --> Listen[触发信号监听]; Listen --> Read[变量读取]; Read --> Expand[扩展步骤]; Expand --> Reset[触发信号复位监听]; Reset --> Common[通用步骤]; Common --> Write[变量下发]; Write --> End[结束];
```

取消 确认

			操作
PLA_工序10			查看
PLA_工序3	PLA_工序4	+35	查看
PLA_工序3	PLA_工序4	+35	查看
			查看
			查看
			查看
			查看
			查看
PLA_工序2	PLA_工序3	+37	查看
			查看

功能配置

FlowInfo.xml
<Abilities/>

```
FlowInfo.xml 50001_1.pfd
1301 <FlowInfo>...</FlowInfo>
1506 <FlowInfo>...</FlowInfo>
1679 <FlowInfo>
1680   <Type>50001</Type>
1681   <Name>Demo测试流程</Name>
1682   <Description></Description>
1683   <Version>1</Version>
1684   <FilePath>50001_1.pfd</FilePath>
1685   <Business Type>
1686     <Name>Other</Name>
1687     <Value>0</Value>
1688     <Description>其他交互</Description>
1689   </Business Type>
1690   <Abilities> "功能配置" 数据源
1691     <Ability>
1692       <Code>Demo_Ability</Code> 功能编码
1693       <Name>Demo功能</Name>
1694       <Description>Demo功能</Description>
1695       <DefaultValue>1</DefaultValue>
1696       <Options>
1697         <Option>
1698           <Name>Disable</Name>
1699           <Value>0</Value>
1700           <Description>禁用</Description>
1701         </Option>
1702         <Option>
1703           <Name>Enable</Name>
1704           <Value>1</Value>
1705           <Description>启用</Description>
1706         </Option>
1707       </Options>
1708     </Ability>
1709   </Abilities>
1710   <Fields>...</Fields>
1740 </FlowInfo>
1741 </Root>
```

Demo工位

* 工位名称: Demo工位

* 所属工序: Demo工序

关联流程: Demo测试流程

看板IP地址: 请输入看板IP地址

SOP信号: 请选择SOP信号

更新码: 请选择更新码

备注: 请输入

参数信号配置

功能配置 | 采集参数 | 配方参数 | 不良原因 | 物料检测 | Demo交互

序号	功能名称	功能描述	* 功能选项
1	Demo功能	Demo功能	<div style="border: 1px solid red; padding: 2px;">启用 禁用 启用</div>

字段配置

FlowInfo.xml
<Fields/>

```
FlowInfo.xml 50001_1.pfd
1301 <FlowInfo>...</FlowInfo>
1506 <FlowInfo>...</FlowInfo>
1679 <FlowInfo>
1680 <Type>50001</Type>
1681 <Name>Demo测试流程</Name>
1682 <Description></Description>
1683 <Version>1</Version>
1684 <FilePath>50001_1.pfd</FilePath>
1685 <BusinessType>...</BusinessType>
1690 <Abilities>...</Abilities>
1710 <Fields> 字段配置数据源
1711 <Field>
1712 <Group>Demo交互</Group> 归属Tab
1713 <Name>触发信号</Name>
1714 <Type>
1715 <Name>Variable</Name>
1716 <Value>1</Value>
1717 <Description>变量</Description>
1718 </Type>
1719 <Description>触发信号</Description> 字段标识(唯一)
1720 <ObjectName>Demo_TriggerSignal</ObjectName>
1721 <ObjectValue>
1722 </ObjectValue>
1723 <IsRequired>True</IsRequired>
1724 </Field>
1725 <Field>
1726 <Group>Demo交互</Group>
1727 <Name>条码变量</Name>
1728 <Type>
1729 <Name>Variable</Name>
1730 <Value>1</Value>
1731 <Description>变量</Description>
1732 </Type>
1733 <Description>条码变量</Description>
1734 <ObjectName>Demo_BarcodeVariable</ObjectName>
1735 <ObjectValue>
1736 </ObjectValue>
1737 <IsRequired>True</IsRequired>
1738 </Field>
1739 </Fields>
1740 </FlowInfo>
1741 </Root>
```

Demo工位

* 工位名称: Demo工位

* 所属工序: Demo工序

关联流程: Demo测试流程

看板IP地址: 请输入看板IP地址

SOP信号: 请选择SOP信号

更新码: 请选择更新码

备注: 请输入

参数信号配置

功能配置 采集参数 配方参数 不良原因 物料检测 **Demo交互**

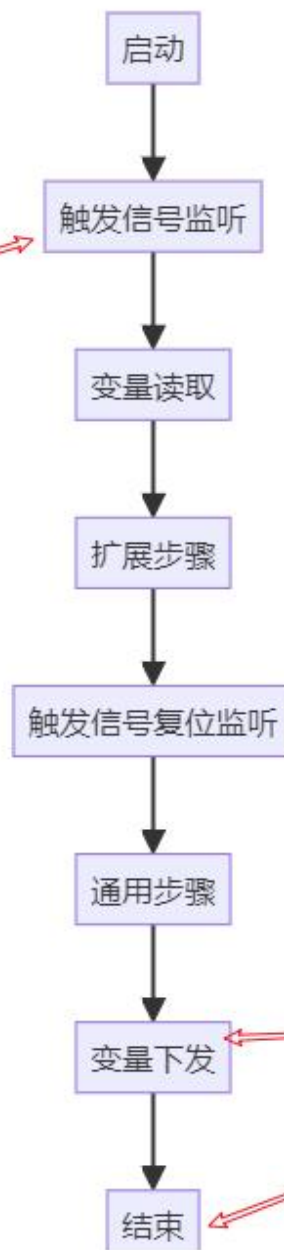
序号	功能字段	描述	变量规则
1	*触发信号	触发信号	请输入 <input type="button" value="选择"/>
2	*条码变量	条码变量	请输入 <input type="button" value="选择"/>

流程配置

50001_1.pfd
<ProcessflowBuilder/>

Demo测试流程

```
50001_1.pfd  x FlowInfo.xml
1 <ProcessflowDefine Version="1">
2 <Define>
3 <ProcessflowBuilder xmlns:xsi="http://www.w3.org/2001/XMLSchema-ns
4 <Parameters/>
5 <Name>Demo业务流程</Name>
6 <Description></Description>
7 <Root xsi:type="Activity">...</Root>
17 <Activities> 定义流程步骤集合
18 <Activity xsi:type="VariableMonitorActivity">
19 <Alias>变量监听</Alias>
20 <Name>触发信号监听</Name>
21 <EnterMode>Automatic</EnterMode>
22 <ExitMode>Automatic</ExitMode>
23 <JoinType>JoinXOR</JoinType>
24 <SplitType>SplitXOR</SplitType>
25 <NeedTagChange>true</NeedTagChange>
26 <TaskDelay>0</TaskDelay> 监听信号, 当变量为1时流程往下流转
27 <MonitorVariableName>Demo_TriggerSignal</MonitorVariableName>
28 <CompareVariableValue>1</CompareVariableValue>
29 <ExtendedProperty></ExtendedProperty>
30 </Activity>
31 <Activity xsi:type="VariableReadAct">...</Activity>
50 <Activity xsi:type="DemoManagementA">...</Activity>
61 <Activity xsi:type="VariableMonitor">...</Activity>
74 <Activity xsi:type="BusinessActivit">...</Activity>
86 <Activity xsi:type="VariableWriteAc">...</Activity>
105 <Activity xsi:type="EndActivity">...</Activity>
114 </Activities>
115 <Transitions>...</Transitions>
152 </ProcessflowBuilder>
153 </Define>
154 <Appearance>...</Appearance>
158 </ProcessflowDefine>
```



```
50001_1.pfd  x FlowInfo.xml
1 <ProcessflowDefine Version="1">
2 <Define>
3 <ProcessflowBuilder xmlns:xsi="http://www.w3.org/2001/XMLSchema-ns
4 <Parameters/>
5 <Name>Demo业务流程</Name>
6 <Description></Description>
7 <Root xsi:type="Activity">...</Root>
17 <Activities>...</Activities>
115 <Transitions> 定义迁移集合
116 <Transition>
117 <Label>启动->触发信号监听</Label>
118 <Source>启动</Source>
119 <Sink>触发信号监听</Sink>
120 </Transition>
121 <Transition>
122 <Label>触发信号监听->变量读取</Label>
123 <Source>触发信号监听</Source>
124 <Sink>变量读取</Sink>
125 </Transition>
126 <Transition>...</Transition>
131 <Transition>...</Transition>
136 <Transition>...</Transition>
141 <Transition>...</Transition>
146 <Transition>
147 <Label>变量下发->结束</Label>
148 <Source>变量下发</Source>
149 <Sink>结束</Sink>
150 </Transition>
151 </Transitions>
152 </ProcessflowBuilder>
153 </Define>
154 <Appearance>...</Appearance>
158 </ProcessflowDefine>
```

确认

流程调试

更新完流程文件后，需要重启流程服务

LMES测试工程

浮点型 时间 字符串 所有变量 Demo_ 清除 批量下发 停止调试

序号	变量名	变量描述	变量组	数据类型	作用域	初始值	当前值	质量戳	最新变...
1	Demo_TriggerS...		-	二进制变量	服务端	无	true	0	15:16:51
2	Demo_Barcode...		-	字符串	服务端	无	AABBCC	0	15:16:49

files\CMS Editor\host\Log\2024-03-29.Log (匹配25次)

```
2024-03-29 15:16:30.9386,Info,9,CMS.Plugin.FlowManagement.Services.FlowInstanceService,《=====启动Demo业务流程:实例=2403291711696474727, Demo工序_Demo工位_50001_1, 版本=50001_1=====》
2024-03-29 15:16:30.9386,Info,9,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号监听 Enter
2024-03-29 15:16:30.9386,Info,9,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号监听 监听信号=Demo_TriggerSignal, CompareValue=1, IsProcessed=False, CanExit=0
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号监听 Demo_TriggerSignal 变量值发生变化, 旧值False->新值True, TraceId=
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号监听 监听信号=Demo_TriggerSignal, CompareValue=1, IsProcessed=True, CanExit=1
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号监听 Exit, 耗时=0.0025毫秒
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量读取 Enter
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量读取 Demo_BarcodeVariable=AABBCC, TraceId=5e132746-5dfc-4240-3291-711696474727
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量读取 Exit, 耗时=0.1466毫秒
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 扩展步骤 Enter
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 扩展步骤 DemoManagementActivity
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 扩展步骤 Exit, 耗时=0.0311毫秒
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号复位监听 Enter
2024-03-29 15:16:51.8105,Info,235,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号复位监听 监听信号=Demo_TriggerSignal, CompareValue=0, IsProcessed=False, CanExit=0
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号复位监听 Demo_TriggerSignal 变量值发生变化, 旧值True->新值False, TraceId=
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号复位监听 监听信号=Demo_TriggerSignal, CompareValue=0, IsProcessed=True, CanExit=1
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 触发信号复位监听 Exit, 耗时=0.0026毫秒
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 通用步骤 Enter
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 通用步骤 Exit, 耗时=0.004毫秒
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量下发 Enter
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量下发 下发信号 Demo_BarcodeVariable=, TraceId=0a33707f-53db-4240-3291-711696474727
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量下发 下发校验 Demo_BarcodeVariable=, TraceId=a075845f-7257-4240-3291-711696474727
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 变量下发 Exit, 耗时=0.1801毫秒
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 结束 Enter
2024-03-29 15:17:23.2448,Info,81,CMS.Plugin.Processflow,执行流程:实例=2403291711696474727 -> Demo工序_Demo工位_50001_1 -> 结束
```

扩展业务：IFlowProcessor

The image shows a Visual Studio IDE with two windows. The left window is the Solution Explorer, showing a project structure for 'CMS.Plugin.DemoManagement'. The file 'C# DemoManagementFlowProcessor.cs' is selected and highlighted with a red box. A red arrow points from this file to the right window, which is the code editor.

The code editor displays the implementation of the `IFlowProcessor` interface in the `DemoManagementFlowProcessor` class. The class is defined as follows:

```
9  /// <summary>
10  /// 流程处理器
11  /// </summary>
12  public class DemoManagementFlowProcessor : IFlowProcessor, ISingletonDependency
13  {
14      private readonly ILogger<DemoManagementFlowProcessor> _logger;
15
16      /// <summary>
17      /// Initializes a new instance of the <see cref="FlowProcessor"/> class.
18      /// </summary>
19      public DemoManagementFlowProcessor(ILogger<DemoManagementFlowProcessor> logger)
20      {
21          _logger = logger;
22      }
23
24      /// <inheritdoc />
25      public Task OnEnterAsync(FlowProcessor flowProcessor, object sender, EventArgs args)
26
27
28
29
30      /// <inheritdoc />
31      public Task OnExecuteAsync(FlowProcessor flowProcessor, object sender, EventArgs args)
32      {
33          if (sender is Activity activity)
34          {
35              var eventArgs = args as ProcessflowEventArgs;
36              var procID = eventArgs.DataItems["Instance_ProcID"];
37              if (activity?.Name == "第三方系统合格判断")
38                  通过判断当前步骤名称，处理对应业务
39              {
40                  _logger.LogInformation($"执行流程：实例={procID} -> {flowProcessor.ProcName} -> {activity.Name} ProductionFlowProcessor");
41              }
42          }
43
44          return Task.CompletedTask;
45      }
46
47      /// <inheritdoc />
48      public Task OnExitAsync(FlowProcessor flowProcessor, object sender, EventArgs args)
49
50
51  }
```

The code highlights the `IFlowProcessor` interface and the `OnExecuteAsync` method. A red box highlights the `if (activity?.Name == "第三方系统合格判断")` condition, with a red annotation: "通过判断当前步骤名称，处理对应业务".

定制步骤：BusinessActivity

The image displays a Visual Studio IDE with three main panes illustrating the implementation of a custom BusinessActivity:

- Left Pane (Solution Explorer):** Shows the project structure for 'CMS.Plugin.DemoManagement'. The file 'DemoManagementActivity.cs' is highlighted in the 'src' folder.
- Middle Pane (Code Editor):** Shows the C# code for 'DemoManagementActivity : BusinessActivity'. The class is marked as [Serializable] and has a public string property 'MyProperty'. The 'ProcessAsync' method is overridden to perform business logic, including logging and returning 'Task.CompletedTask'.

```
4 using CMS.Plugin.OrderManagement.Abstractions.Models;
5 using CMS.Plugin.ProcessManagement.Abstractions.Models;
6 using CMS.Plugin.TraceManagement.Abstractions.Models.Traces;
7 using Microsoft.Extensions.Logging;
8 using SYC.Flow.Kernel;
9
10 namespace CMS.Plugin.DemoManagement.FlowBusinessExtension
11 {
12     /// <summary>
13     /// DemoManagement 业务步骤
14     /// </summary>
15     [Serializable]
16     public class DemoManagementActivity : BusinessActivity
17     {
18         /// <summary>
19         /// 配置属性 (XML: <MyProperty>Value</MyProperty>)
20         /// </summary>
21         [DataMember]
22         public string MyProperty { get; set; }
23
24         /// <summary>
25         /// 工艺流程处理
26         /// </summary>
27         public override Task ProcessAsync(ProcessflowEvent Args args)
28         {
29             // 工艺模型
30             var processModel = Flow.DataItems.ApplicationData as ProcessM
31
32             // 工单模型
33             var orderModel = Flow.DataItems[FlowItemCollection.OrderMod
34
35             // 产品模型
36             var productModel = Flow.DataItems[FlowItemCollection.Product
37
38             // 追溯模型
39             var traceModel = Flow.DataItems[FlowItemCollection.TraceMod
40
41             Flow.Logger.LogInformation(message: $"执行流程: 实例={Flow.Ins
42
43             return Task.CompletedTask;
44         }
45     }
```
- Right Pane (XML Editor):** Shows the XML definition for the '50001_1.pfd' process. The 'DemoManagementActivity' is defined as an activity within the 'ProcessflowBuilder'.

```
1 <ProcessflowDefine Version="1">
2 <Define>
3 <ProcessflowBuilder xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
4 <Parameters/>
5 <Name>Demo业务流程</Name>
6 <Description></Description>
7 <Root xsi:type="Activity">...</Root>
17 <Activities>
18 <Activity xsi:type="VariableMonitor">...</Activity>
31 <Activity xsi:type="VariableReadAct">...</Activity>
50 <Activity xsi:type="DemoManagementActivity">
51 <Alias>变量监听</Alias>
52 <Name>扩展步骤</Name>
53 <EnterMode>Automatic</EnterMode>
54 <ExitMode>Automatic</ExitMode>
55 <JoinType>JoinXOR</JoinType>
56 <SplitType>SplitXOR</SplitType>
57 <TaskDelay>0</TaskDelay>
58 <MyProperty>MyProperty1</MyProperty>
59 <ExtendedProperty></ExtendedProperty>
60 </Activity>
61 <Activity xsi:type="VariableMonitor">...</Activity>
74 <Activity xsi:type="BusinessActivity">
75 <Alias>通用步骤</Alias> 也可使用通用步骤, 配合IFlowProcessor
76 <Name>通用步骤</Name>
77 <EnterMode>Automatic</EnterMode>
78 <ExitMode>Automatic</ExitMode>
79 <JoinType>JoinXOR</JoinType>
80 <SplitType>SplitXOR</SplitType>
81 <TaskDelay>0</TaskDelay>
82 <ExtendedProperty>
83 <MyExtendedProperty>MyExtendedProperty1</MyExtendedProperty>
84 </ExtendedProperty>
85 </Activity>
86 <Activity xsi:type="VariableWriteAc">...</Activity>
105 <Activity xsi:type="EndActivity">...</Activity>
114 </Activities>
115 <Transitions>...</Transitions>
152 </ProcessflowBuilder>
153 </Define>
154 <Appearance>...</Appearance>
158 </ProcessflowDefine>
```

订阅事件：ProcessFlowEto

The image displays a Visual Studio development environment with three main windows:

- Left Window (Solution Explorer):** Shows the project structure for 'CMS.Plugin.DemoManagement'. The file 'C# DemoManagementEventHandler.cs' is highlighted with a red box.
- Middle Window (Code Editor):** Shows the implementation of `DemoManagementEventHandler`. The class is annotated with `IDistributedEventHandler<ProcessFlowEto>` and `TransientDependency`. The `HandleEventAsync` method processes flow items for various models like `ProcessModel`, `AssociationProductModel`, `TraceModel`, and `OrderModel`. A red box highlights the `if (eventData.Activity.Equals("步骤名称"))` condition, with a red arrow pointing to the BPMN XML.
- Right Window (Code Editor):** Shows the BPMN XML configuration for the process flow. The `ProcessflowDefine` element includes a `ProcessflowBuilder` with several activities. A red box highlights the `publishFlowEvent>true` attribute on the `BusinessActivity` element, which is linked by a red arrow from the C# code's `if` statement.

```
public class DemoManagementEventHandler : IDistributedEventHandler<ProcessFlowEto>, TransientDependency
{
    private readonly ILogger _logger;
    private readonly IServiceProvider _serviceProvider;

    /// <summary>
    /// Initializes a new instance of the <see cref="DemoManagementEventHandler"/> class.
    /// </summary>
    /// <param name="logger">The logger.</param>
    /// <param name="serviceProvider">The service provider.</param>
    public DemoManagementEventHandler(ILogger<DemoManagementEventHandler> logger, IServiceProvider service

    /// <summary>
    /// Handler handles the event by implementing this method.
    /// </summary>
    /// <param name="eventData">Event data</param>
    public Task HandleEventAsync(ProcessFlowEto eventData)
    {
        var serialNumber = eventData?.FlowItems[FlowItemCollection.SerialNumber]?.ToString();

        // 工艺模型
        var process = eventData?.FlowItems[FlowItemCollection.ApplicationData] as ProcessModel;

        // 产品模型
        var product = eventData?.FlowItems[FlowItemCollection.ProductModel] as AssociationProductModel;

        // 追溯模型
        var trace = eventData?.FlowItems[FlowItemCollection.TraceModel] as TraceModel;

        // 工单模型
        var order = eventData?.FlowItems[FlowItemCollection.OrderModel] as OrderModel;

        if (eventData.Activity.Equals("步骤名称"))
        {
            // TODO: 业务逻辑
        }
    }
}
```

```
<ProcessflowDefine Version="1">
  <Define>
    <ProcessflowBuilder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      <Parameters/>
      <Name>MyPluginName业务流程</Name>
      <Description/>
      <Root xsi:type="Activity">...</Root>
      <Activities>
        <Activity xsi:type="VariableMonitor">...</Activity>
        <Activity xsi:type="VariableReadAct">...</Activity>
        <Activity xsi:type="MyPluginNameAct">...</Activity>
        <Activity xsi:type="VariableMonitor">...</Activity>
        <Activity xsi:type="BusinessActivity">
          <Alias>通用步骤</Alias>
          <Name>通用步骤</Name>
          <EnterMode>Automatic</EnterMode>
          <ExitMode>Automatic</ExitMode>
          <JoinType>JoinXOR</JoinType>
          <SplitType>SplitXOR</SplitType>
          <PublishFlowEvent true</PublishFlowEvent>
          <TaskDelay>0</TaskDelay>
          <ExtendedProperty>
            <MyExtendedProperty>MyExtendedProperty1</MyExtendedProperty>
          </ExtendedProperty>
        </Activity>
        <Activity xsi:type="VariableWriteAc">...</Activity>
        <Activity xsi:type="EndActivity">...</Activity>
      </Activities>
      <Transitions/>
    </ProcessflowBuilder>
  </Define>
  <Appearance>
    <ProcessflowAppearance xmlns:xsi="http://www.w3.org/2001/XMLSchema-inst"
    </ProcessflowAppearance>
  </Appearance>
</ProcessflowDefine>
```

四、打包发布

打包发布

参阅 《[CMSv2插件打包工具说明.pdf](#)》 ， [插件打包工具](#)。

谢谢！