

2015 Pack Expo Las Vegas

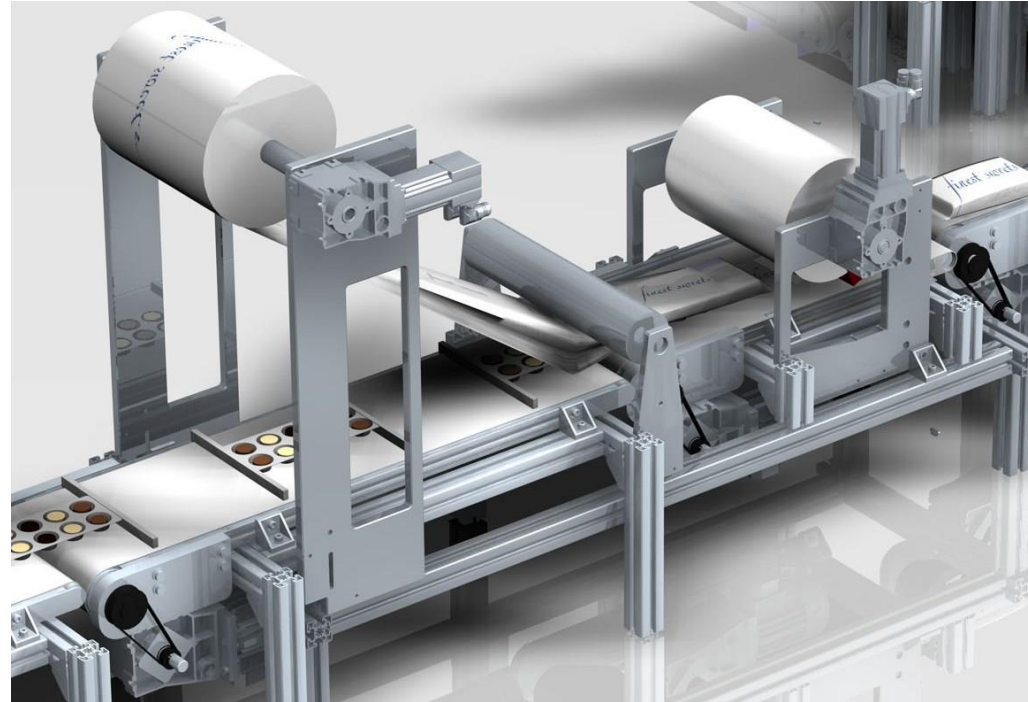
Lenze Application Template

Daniel Repp

Controller-based Automation

Lenze OMAC-PackML Standard

80% of the engineering
in 20% of the time -
Added value based
on standardization
and reuse



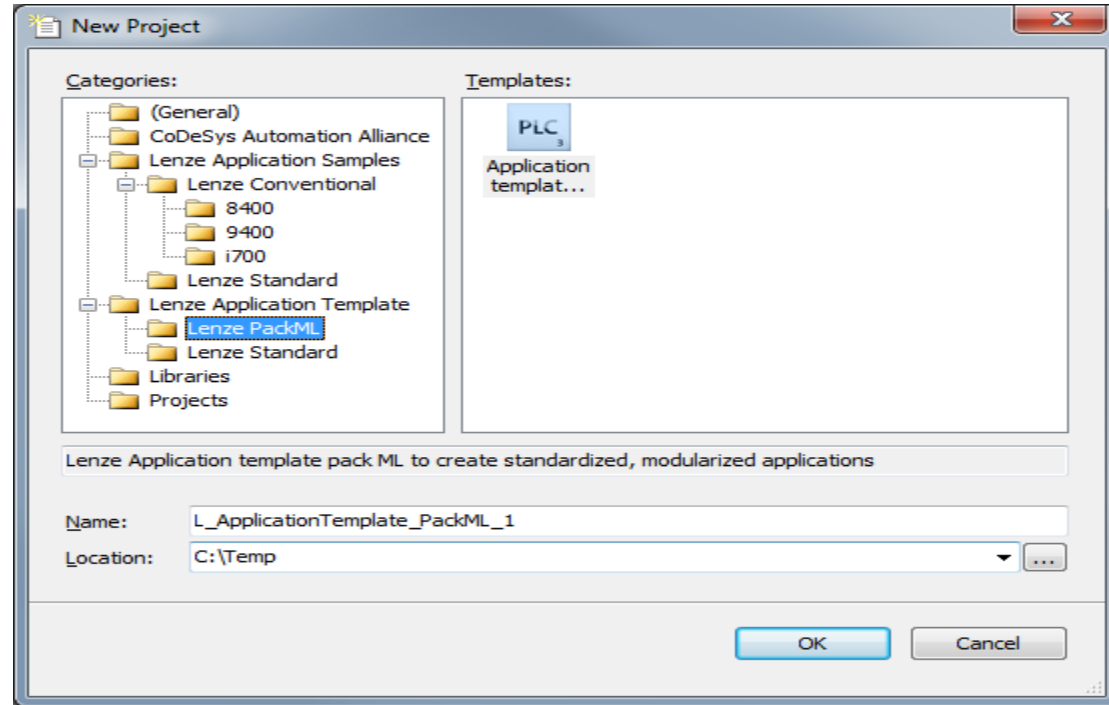
Lenze Application Template

It's not only a description of a standard
it's a Lenze product!

Lenze Application Template PackML

Technical survey

The Lenze Application Template is included in The PLC Designer installation

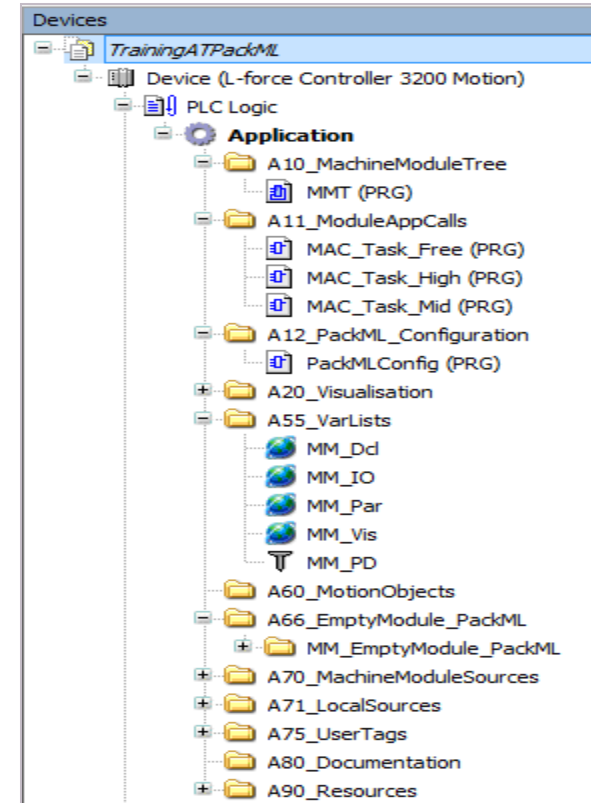


Lenze Application Template PackML

Technical survey

Framework and software structure

- The Application Template PackML supports our clients in terms of the implementation of a modular software structure to their machines.
- It offers functions that simplify the implementation of a machine application in a PLC:
 - provides the software foundation and the framework (Machine modules, Technology modules, Motion Control e.g.) based on norms and standards,
 - offers automatic code generation and provides the management of state, mode and errors,
 - guarantees the data consistency between the tasks in case of multitasking,
 - provides diagnosis functions for the individual machine modules.

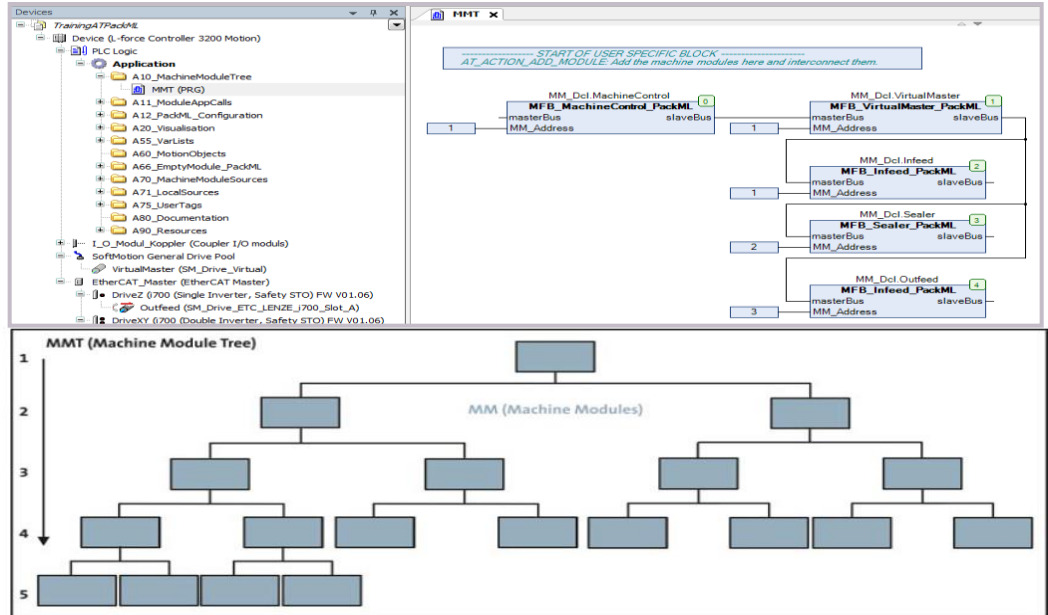


Lenze Application Template PackML

Technical survey

Modular Machine Tree

- The modular machine tree (machine structure) is compiled in a graphical editor.
- Illustration of the modular machine concept.
- Support of 2–5 hierarchic levels of machine modules.
- Support of up to 30 machine modules.
- The tree always starts with a MachineControl module.

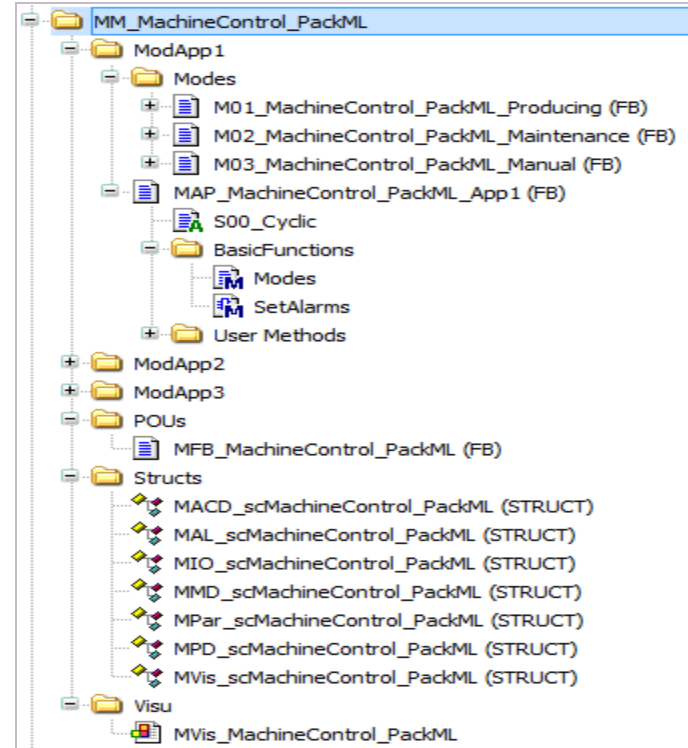
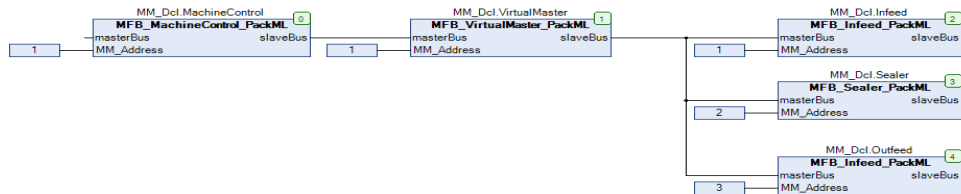


Lenze Application Template PackML

Technical survey

Functions of the machine modules:

- State machine
- Switch of operating modes
- Error handling
- Communication with higher-level und lower-level modules
- Communication with IO, drives, visualization, ...

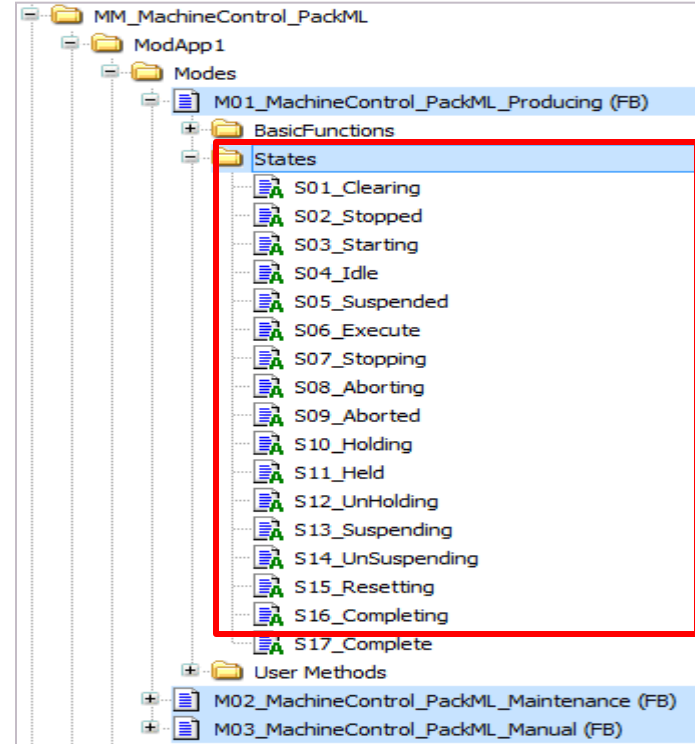


Lenze Application Template PackML

Technical survey

State machine and mode

- Every machine module contains a standard state machine.
- The state machine represents the existing states of the machine for every mode.
- Both fixed and user-specific modes can be used.
- The error response of every single machine module can be parameterized via the state machine.

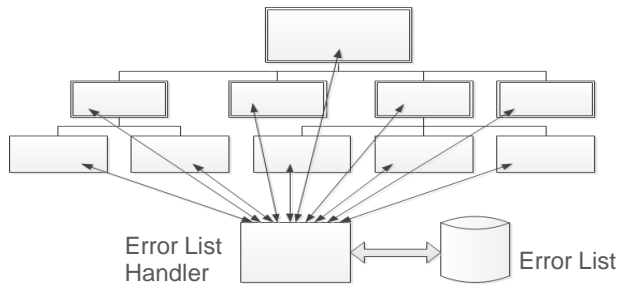


Lenze Application Template PackML

Technical survey

Error handling

- Machine modules can cause errors in the application
- Standard error handling in other machine modules via the state machine and the PackTags
- Module specific error responses can be implemented



```

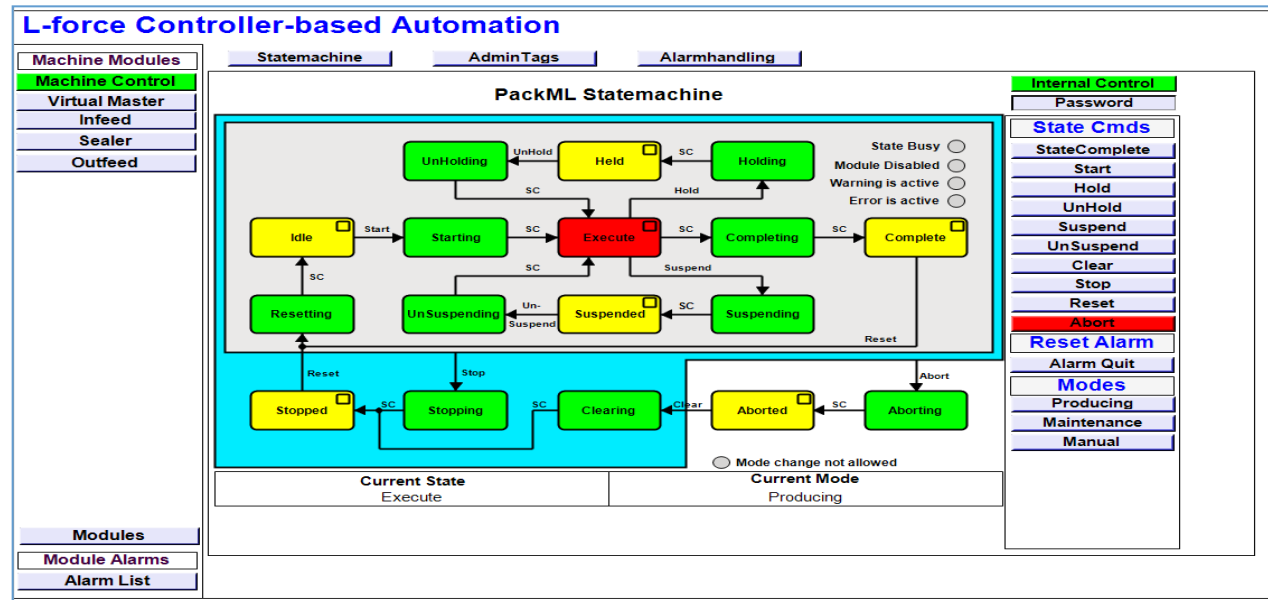
sOMAL.AlarmsA
  L_EATP_Alarm_PackML
    xAlarm1Trigger
    wAlarm1ID
    dwAlarm1Value
    sAlarm1Message
    dwAlarm1Category
    L_EATP_AlarmReactionType.Abort
    FALSE
    xAlarm1AckNeeded
    xAlarm2Trigger
    wAlarm2ID
    dwAlarm2Value
    sAlarm2Message
    dwAlarm2Category
    L_EATP_AlarmReactionType.Abort
    TRUE
    xAlarm2AckNeeded
    xAlarm3Trigger
    wAlarm3ID
    dwAlarm3Value
    sAlarm3Message
    dwAlarm3Category
    L_EATP_AlarmReactionType.Stop
    TRUE
    xAlarm3AckNeeded
    xAlarm4Trigger
    wAlarm4ID
    dwAlarm4Value
    sAlarm4Message
    dwAlarm4Category
    L_EATP_AlarmReactionType.Stop
    FALSE
    xAlarm4AckNeeded
    xAlarm1Active - Vis.dwAlarmsActive.0
    xAlarm2Active - Vis.dwAlarmsActive.1
    xAlarm3Active - Vis.dwAlarmsActive.2
    xAlarm4Active - Vis.dwAlarmsActive.3
  
```

Lenze Application Template PackML

Technical survey

Standard diagnosis and operating

The visualization in
the PLC designer
provides an error list
and a status overview
of all machine modules



Lenze Application Template PackML

Technical survey

Standard diagnosis and operating

- Module and alarm list

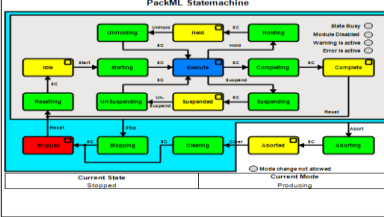
L-force Controller-based Automation

Machine Modules	MM Addr.	MM Name	ST	CPL
1	1	MachineControl	●	PML_3C
2	1.1	VirtualMaster	●	PackML
3	1.1.1	Infeed	●	PackML
4	1.1.2	Sealer	●	PackML
5	1.1.3	Outfeed	●	PackML
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Machine Module Details

MM Address	1
MM Name	MachineControl
MM ComplID	2C00
MM Version	0
Master MM Address	
Master MM Name	
Number of Slaves	1

PackML StateMachine



Up Down StateMachine Alarmhandling

L-force Controller-based Automation

Machine Modules	MM Addr.	MM Inst.	ID	Value	Category	Message	Reaction
Machine Control	1.1.1	Infeed	11100	183	0	Fault TM ElectricalShaft	Fault
Virtual Master	1.1.1	Infeed	11102	20	0	Fault Axis	Stop
Infeed			0	0	0		None
Sealer			0	0	0		None
Outfeed			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None
			0	0	0		None

Modules Create CSV file

Module Alarms

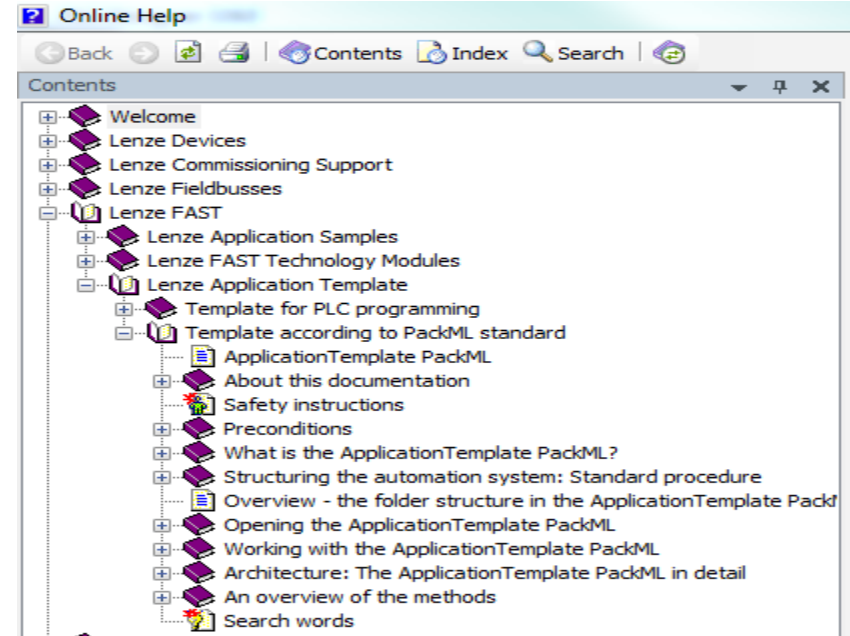
Alarm List

Lenze Application Template PackML

Support survey

Online help

- Overview
- Architecture in detail
- Example



Lenze Application Template PackML


Support survey

Getting started

- Project and documentation for an example with step by step programming and the solution project

Home > Products & Solutions > Application Knowledgebase > Automation Software > GettingStarted for ApplicationTemplate PackML (3200C, p500)

German article: GettingStarted für ApplicationTemplate PackML (3200C, p500)

Document ID	201400244
Title	GettingStarted for ApplicationTemplate PackML (3200C, p500)
Languages	English
Article Content	<p>Description: GettingStarted addresses to all persons who want to know how to handle and use the ApplicationTemplate with PackML State model by means of a sample application for the i700 Lenze teachware case, which is to be configured step by step.</p> <p>The AT_PackML_GettingStarted_V38a.ZIP (3.85MB) file includes three files:</p> <ul style="list-style-type: none"> • GettingStarted_AT_PackML_VMasterEShaft_en.pdf The software manual describes how to implement a sample project on the basis of the ApplicationTemplate with PackML State model. • L_GettingStarted_AT_PackML_VMaster&EShaft.project Project template which can be used as a basis to implement exercises from the software manual. • L_GettingStarted_AT_PackML_VMaster&EShaft_Final.project Executable project in which all solutions described in the software manual are implemented. <p>System requirements:</p> <ul style="list-style-type: none"> - PLC Designer V3.8 - Installed libraries: <ul style="list-style-type: none"> - L_TT1P_TechnologyModules_SM3 V3.8.0.23 - L_EATP_ApplicationTemplate V3.8.0.7 - 3200C / p500 firmware V3.7
Comments	
Remarks/Links	
Author	Mahnken, Eckehard - Lenze
Editor	Mittelstädt, Jan - Lenze 
Approver	

Thank you for your attention!

Less means more!

Focused on the essentials: the new i500 inverter series.



Visit Lenze at Booth #S-6033

PACK EXPO Las Vegas
September 28 – 30, 2015

As easy as that.

Lenze