



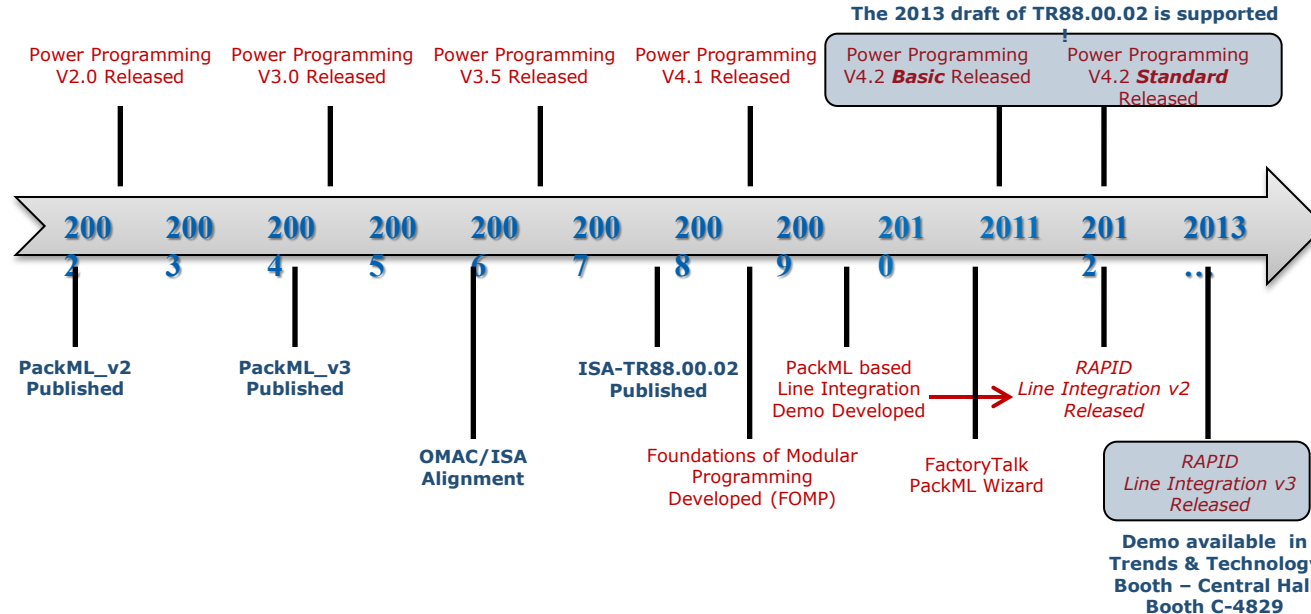
The Rockwell Automation logo consists of the words "Rockwell" and "Automation" stacked vertically in a white, bold, sans-serif font, set against a solid red rectangular background.

2015 Pack Expo Las Vegas

Rockwell Automation & PackML

John Dart – Industry Program Manager
Presenter

Rockwell Automation PackML Development History



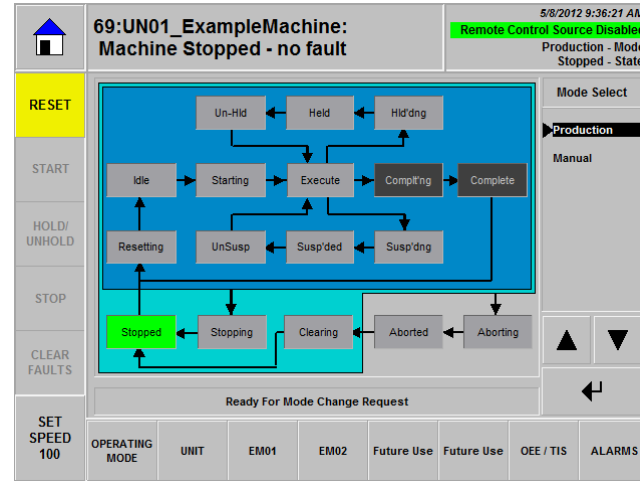
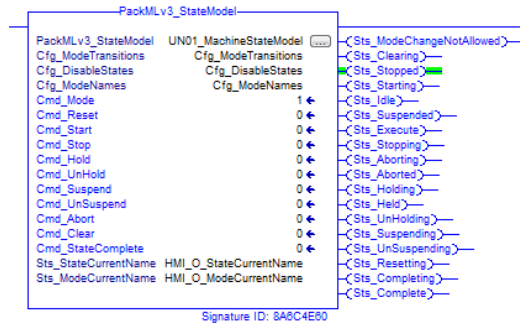
* The above dates are approximate

Rockwell Automation PackML Implementation – State Model

State transition rules defined per ISA-TR88.00.02.

Mode transition rules set by defining for each mode which states a mode transition will be allowed

For a mode to change, both current and desired mode must both be configured to allow mode transition in the current state



Rockwell Automation PackTags Implementation – Commands

- *PackTags are supported in Power Programming.*

[-] PackML_Data.Command	UDT_PackML_Commands	
[+] PackML_Data.Command.UnitName	STRING	
[+] PackML_Data.Command.UnitMode	DINT	Decimal
[-] PackML_Data.Command.UnitModeChangeRequest	BOOL	Decimal
[-] PackML_Data.Command.MachSpeed	REAL	Float
[+] PackML_Data.Command.MaterialInterlocks	DINT	Decimal
[+] PackML_Data.Command.CntrlCmd	DINT	Decimal
[-] PackML_Data.Command.CmdChangeRequest	BOOL	Decimal
[-] PackML_Data.Command.ResetPackMLTimes	BOOL	Decimal
[-] PackML_Data.Command.Heartbeat	BOOL	Decimal
[+] PackML_Data.Command.Product	DINT	Decimal

Rockwell Automation PackTags Implementation – Status

[-] PackML_Data.Status	UDT_PackML_Status	
[+] PackML_Data.Status.UnitModeCurrent	DINT	Decimal
[+] PackML_Data.Status.UnitModeCurrentName	STRING	
[-] PackML_Data.Status.UnitModeRequested	BOOL	Decimal
[-] PackML_Data.Status.UnitModeChangeInProgress	BOOL	Decimal
[+] PackML_Data.Status.StateCurrent	DINT	Decimal
[+] PackML_Data.Status.StateCurrentName	STRING	
[-] PackML_Data.Status.StateRequested	BOOL	Decimal
[-] PackML_Data.Status.StateChangeInProgress	BOOL	Decimal
[-] PackML_Data.Status.MachSpeed	REAL	Float
[-] PackML_Data.Status.CurMachSpeed	REAL	Float
[+] PackML_Data.Status.MaterialInterlocks	DINT	Decimal
[+] PackML_Data.Status.RemoteInterface	UDT_PackML_RemoteInterface[1]	
[+] PackML_Data.Status.Parameter	UDT_PackML_Descriptor[1]	
[+] PackML_Data.Status.Product	UDT_PackML_Product[1]	
[-] PackML_Data.Status.RemoteControlEnabled	BOOL	Decimal
[-] PackML_Data.Status.Heartbeat	BOOL	Decimal
[-] PackML_Data.Status.ResetPackMLTimesDone	BOOL	Decimal

Rockwell Automation

PackTags Implementation – Administration

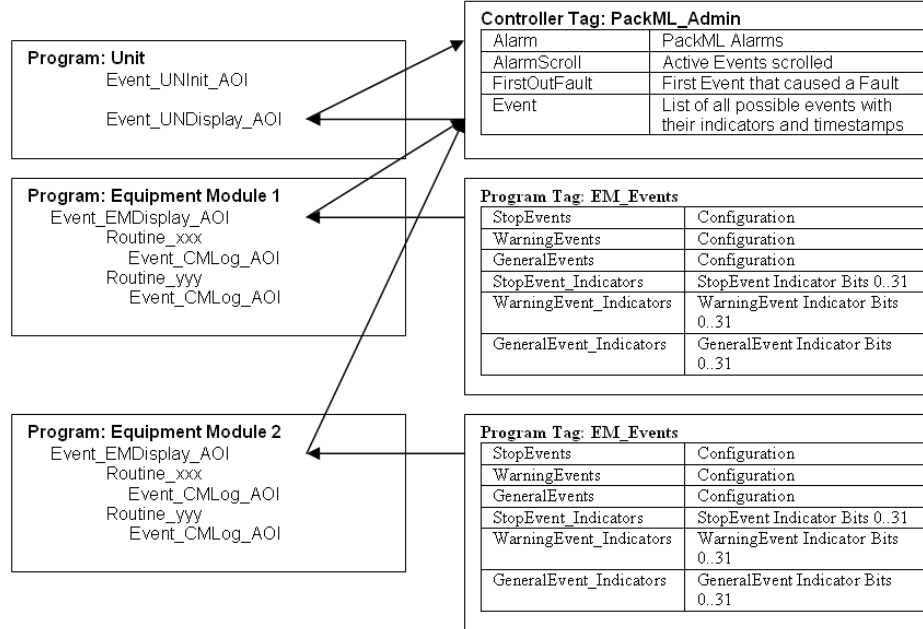
[-] PackML_Data.Admin	UDT_PackML_Administration	
[+] PackML_Data.Admin.UnitName	STRING	
[+] PackML_Data.Admin.Parameter	UDT_PackML_Descriptor[1]	
[+] PackML_Data.Admin.Alarm	UDT_Event[32]	
[+] PackML_Data.Admin.AlarmExtent	DINT	Decimal
[+] PackML_Data.Admin.FirstOutFault	UDT_Event	
[+] PackML_Data.Admin.AlarmScroll	UDT_Event	
[+] PackML_Data.Admin.Event	UDT_Event[256]	
[+] PackML_Data.Admin.EventExtent	DINT	Decimal
[+] PackML_Data.Admin.ModeCurrentTime	DINT	Decimal
[+] PackML_Data.Admin.ModeCumulativeTime	DINT[32]	Decimal
[+] PackML_Data.Admin.StateCurrentTime	DINT	Decimal
[+] PackML_Data.Admin.StateCumulativeTime	UDT_PackML_CummulativeStateTime[32]	
[+] PackML_Data.Admin.ProdConsumedCount	UDT_PackML_CntDescrip[1]	
[+] PackML_Data.Admin.ProdProcessedCount	UDT_PackML_CntDescrip[1]	
[+] PackML_Data.Admin.ProdDefectiveCount	UDT_PackML_CntDescrip[1]	
[+] PackML_Data.Admin.AccTimeSinceReset	DINT	Decimal
[-] PackML_Data.Admin.MachDesignSpeed	REAL	Float
[+] PackML_Data.Admin.AlarmHistory	UDT_Event[1]	
[+] PackML_Data.Admin.AlarmHistoryExtent	DINT	Decimal
[-] PackML_Data.Admin.PACDateTime	LINT	Decimal

Rockwell Automation PackML Implementation – Event Hierarchy

Alarms and "First Out Faults" are recorded in the PackML data structure as shown.

Alarms and Events are aggregated from the equipment level (EM) to the Unit level as shown in this diagram.

Tools are provided for import/export of event ID's & Message Strings. This tool also facilitates Language support.



Rockwell Automation

PackML Implementation – ISA-88 Modularity

Power Programming Capability Matrix	PP Standard	PP Basic	Notes
ISA-88 Modular Structure	●	●	Some Procedural Control in EM's
ISA-88 PackML State Model	●	●	Identical
ISA-88 PackML Data Model (PackTags)	●	●	Identical
Performance Tracking (OEE) Functionality	●	●	Identical
Command Condition UDT Interface to EM Level	●	○	
Mode/State Condition Interface to EM Level	○	●	
Fault Handling - Unit Level Summary	●	●	
Fault Handling - ISA-88 Modularity (CM/EM)	●	○	
Fault Handling - Multi-Language Support	●	○	
Fault Handling - Level 1 Diags/Message Hierarchy	●	○	
Global OEM Application Library Compatible	●	●	
Scalable to large or complex applications	●	●	Memory use info available
Scalable to small applications (lower cpu mem)	●	●	

Key: ● Full Capability
● Partial Capability
○ No Capability

Rockwell Automation PackML Implementation - Documentation

- Power Programming “Application Guidelines”

LISTEN.
THINK.
SOLVE.™

Power Programming V4 Standard Application Guidelines

A Modular Programming example based on ISA-TR88.00.02 (PackML)

OVERVIEW

Power Programming 'Standard' is intended to be suitable for any Logix controllers and is useful for simple to complex applications.

APPLICATION

Power Programming 'Standard' provides example HMI and PAC application software which utilizes common tools included in PPV4.2 Basic such as the AOI based PackML state model and Machine Performance Tracking. Additionally, a suite of comprehensive fault/eventAOI's have been added as well as additional structure to isolate procedural code.

Key features include:

- HMI Example with PackML state model display and mode controls
- Logix Example with PackML (V3) state model and mode management

-Application code samples:

- Fault Handling
- Equipment Interface
- Equipment Modules
- Control Modules

Application Library # OEM120xx (RA Only)

Knowledgebase ID # 66060

Allen-Bradley • Rockwell Software

LISTEN.
THINK.
SOLVE.™

Power Programming V4 Basic Application Guidelines

A Modular Programming example based on ISA-TR88.00.02 (PackML v3)

OVERVIEW

Power Programming Basic is intended to be suitable for any Logix controllers including small memory sizes and is useful for simple to moderately complex applications.

APPLICATION

Power Programming Basic provides example HMI and PAC application software which includes key features from PPV4.2 Standard such as the AOI based PackML state model and utilizes simpler methods for interfacing to equipment controls and fault handling.

Key features include:

- HMI Example with PackML state model display and mode controls
- Logix Example with PackML (V3) state model and mode management

-Application code samples:

- Fault Handling
- Equipment Interface
- Equipment Modules
- Control Modules

Application Library # OEM11015 (RA Only)

Knowledgebase ID # 66060

Allen-Bradley • Rockwell Software



Rockwell Automation PackML Implementation – Future Plans

- **Continue to work on solving customer integration challenges.**
- Most changes in the new revision are already reflected in the current version of Power Programming.
- Fully re-evaluate ISA-TR88.00.02 Revision 2 (2015).



Questions?