mapp Services

1 mapp PackML: OMAC PackML standard

mapp PackML complies with the requirements of the OMAC PackML standard to generate the logic for the machine controller. This component can be used to manage the various PackML states and modes as well as incorporating a Visual Components 4 HMI application. For more information about OMAC and PackML, see:

- Official website of the OMAC organization
- PackML on Wikipedia

1.1 Concept

<table>
<thead>
<tr>
<th>User stories</th>
<th>Description</th>
</tr>
</thead>
</table>
| **End user** | • As an end user, I would like to see the same modes and states on all machines in my line.  
• As an end user, I would like to depict the state of my machine using the PackML state diagram. |
| **OEM**      | • As an OEM, I have to supply machines that follow the PackML standard.  
• As a machine operator, I would like to see the current state of my machine in graphic form. |
| **Machine operator** | • As a service technician, I would like to have the option of adjusting the configuration of the PackML model (adjust the start mode, options for switching between several modes, etc.).  
• As an application engineer, I would like to easily add the PackML state diagram to an already existing application.  
• As an application engineer, I would like to build a new application on the PackML model. |
| **Service technician** | |
| **Application engineer** | |

1.1.1 Auditing the PackML state

Events for these components can be recorded using MpAudit. For more information about how to create an entry, see section Creating entries.

The following events can occur for these mapp components:

**MpPackML mode change**

This event type allows MpPackML mode changes to be tracked.

Option "Auditing" must be enabled in the MpPackMLCore configuration in order to record events:
### Event ID

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>MpPackML components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Events</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ev</td>
<td>Event ID</td>
</tr>
<tr>
<td>idx</td>
<td>Returns the index of the currently generated event. The value is incremented on each new event. The first generated event in the event list thus has value 1; the 50th event has index 50 and the 100th has index 100.</td>
</tr>
<tr>
<td>old</td>
<td>Old PackML mode</td>
</tr>
<tr>
<td>new</td>
<td>New PackML mode</td>
</tr>
<tr>
<td>resp</td>
<td>Source of the PackML mode or state change (e.g. &quot;MpPackMLBasicUI&quot;, &quot;MpPackMLMode&quot;, &quot;PackML tag structure&quot;)</td>
</tr>
<tr>
<td>state</td>
<td>PackML state in which a mode change was initiated</td>
</tr>
</tbody>
</table>

### Examples of the new format


### Example of the old format


### MpPackML state change

This event type allows MpPackML state changes to be tracked.

Option "Auditing" must be enabled in the MpPackMLCore configuration in order to record events:
**Event ID**

<table>
<thead>
<tr>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>MpPackML components</td>
</tr>
</tbody>
</table>

**Components**

- **Ev**
  - Returns the index of the currently generated event. The value is incremented on each new event. The first generated event in the event list thus has value 1; the 50th event has index 50 and the 100th has index 100.

- **Old**
  - Old PackML state

- **Resp**
  - Source of the PackML mode or state change (e.g., "MpPackMLBasicUI", "MpPackMLMode", "PackML tag structure")

- **New**
  - New PackML state

**Examples of the new format**


**Examples of the old format**


Table 2: MpPackML state change

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>MpPackML components</td>
</tr>
</tbody>
</table>

1.1.2 Using the PackML PackTags structure

MpPackML provides the possibility of communicating using the PackTags structure specified by OMAC. This structure must be created by the user. The structure below can be used. Tags contained in the description "[Covered by MpPackMLCore PackTags]" are automatically filled out by MpPackML; the user does not have to worry about these.

The structure shown above depicts the complete PackTags structure. If the complete structure is not needed, then the minimal PackTags structure can be used. The tags that belong to the minimal PackTags structure are identified by "[Basic Type]" in the description. This makes it possible for the user to use a simplified version of PackTags.

PackTags consist of 3 areas:

- **PMLc (PackML command):** With the command structure, the administrator is able to carry out commands such as switching the PackML mode or state.

- **PMLs (PackML status):** The status structure shows status information such as the current mode.

- **PMLa (PackML administration):** The administration structure consists of statistical information such as the total runtime of a mode.

The tags below are supported by MpPackML. The required tags must be included in the respective structure.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Day</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMLc</td>
<td>UnitMode</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>UnitModeChangeRequest</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>CntrCmd</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>CmdChangeRequest</td>
<td>Required</td>
</tr>
<tr>
<td>PMLs</td>
<td>UnitModeCurrent</td>
<td>Required</td>
</tr>
</tbody>
</table>
PackTags structures

Minimal PackTags structure
This structure contains the basic structure: PackMLTagsBasic

Complete PackTags structure
This structure contains the complete PackTags structure: PackMLTagsFull

The PackTag structures listed here can be used directly in Automation Studio. This is done by downloading and opening the desired structure. In Automation Studio, the Types.typ file is opened in the desired program as text. The contents of the structure can then be copied to the type file.

Configuration
The use of PackTags is enabled in the MpPackML configuration. A variable of a suitable data type is created for each of the 3 structures.

The variable for the correct structure is then selected under "PackTags handling". In other words, the variable "PMLc" is specified as the command structure under "Command structure name", etc.

If the structure is stored locally, then the program name is specified before the structure name ("PackMLProgram:PMLc").

In this case, the structures are created globally:
In order for the variables from the different stations on the production line to communicate, they are enabled via OPC UA, for example.

**Usage**

The current status of PackTags is indicated in structure `MpPackMLCoreInfoType` under "PackTagsStatus". During initialization of PackTags, the status is "mpPACKML_PACKTAGS_INIT". "mpPACKML_PACKTAGS_OK" indicates that initialization was successful and PackTags can be used.

**Commands:**

OMAC defines recipe parameters, for example, as "Product" and parameters that are not used in production (i.e. machine parameters) as "ProcessVariables" in the command structure (PMLc). Recipe parameters can be saved as a file using MpRecipe, for example. To do so, parameter "Product" is added to MpRecipeRegPar as a process variable.

```plaintext
//Recipe parameters
MpRecipeXml_0.MpLink := ADR(gRecipeXml);
MpRecipeXml_0.Enable := TRUE;
MpRecipeXml_0.DeviceName := ADR('HD');
MpRecipeXml_0.FileName := ADR('RecipeProducts');

MpRecipeRegPar_0.MpLink := ADR(gRecipeXml);
MpRecipeRegPar_0.Enable := TRUE;
MpRecipeRegPar_0.PVName := ADR('PMLc.Command.Product');

MpRecipeRegPar_0();
MpRecipeXml_0();

END_PROGRAM
```
**Status:**
The current mode can be queried in the status structure, for example. It is shown under "PMLs.UnitModeCurrent". This query can be used as a condition for commands. If the desired mode is active, the "OpenDoor" command can be enabled, for example.

**Administration:**
Statistical information is displayed in the administration structure. Here it is possible to query things like the value "PMLa.StateCurrentTime", which displays how long the current mode is already active. This can be used for checking purposes, for example. If the value "PMLa.StateCurrentTime" reaches a user-defined maximum time, a user alarm (e.g. "ReachedMaxTime") can be triggered.

The PMLa structure should be marked as "Retain" so that the statistical information it contains is not lost. This way, the information will still be available after the controller is restarted.

If the PackTags structure is changed at runtime, the MpPackMLCore component must be re-enabled! This ensures that all changes to the structure are applied.

### 1.2 Guides

#### 1.2.1 Getting started

##### 1.2.1.1 Implementing the OMAC PackML standard

This section explains step by step how to integrate the OMAC PackML standard into a machine application using MpPackMLCore and MpPackMLMode.

##### 1.2.1.1.1 Creating a project

You must first create a new project in Automation Studio. For more information, see here.

##### 1.2.1.1.2 Adding the mapp component

**Adding the MpPackMLCore configuration**

The MpPackMLCore configuration is added to use the OMAC PackML standard.

It contains the mapp Link, which uniquely identifies the mapp component within the mapp environment. The mapp Link establishes the connection between programming and configuration.

Mode "Production" exists by default and has ID 1. All commands defined by the OMAC PackML standard such as "Execute" or "Stopped" are enabled by default and can be used for the mode.
1.2.1.3 Adding a program

The next step is to add a Ladder Diagram program. For more information, see here.

1.2.1.4 Adding and configuring the function blocks

This section explains which function blocks are added and configured.

**Adding MpPackMLCore**

In the next step, function block MpPackMLCore is added. It manages all modes defined in the configuration.

**Configuring MpPackMLCore**

**Connecting MpLink**

Using input "MpLink", we establish a connection to the configuration we created earlier named "gPackMLCore".

**Enabling the function block**

The function block is enabled using input "Enable".
Adding MpPackMLMode

Function block MpPackMLMode is added to manage mode "Production".

Configuring MpPackMLMode

Connecting MpLink

Using input "MpLink", we establish a connection to the configuration we created earlier named "gPackMLCore".

Enabling the function block

The function block is enabled using input "Enable".
Defining the mode

Finally, the mode ID of the mode to be controlled must be specified. Mode "Production", which has the ID 1, should be managed. The ID was defined in the MpPackMLCore configuration.

1.2.1.1.5 Generating the file structure

Now generate a file structure for the memory card for ARsim. For more information, see here.

1.2.1.1.6 Testing the program

After the changes are downloaded, the program can be tested. Monitor mode is enabled.

Function blocks MpPackMLCore and MpPackMLMode were added to the Watch window. To enable mode "Production", command "Activate" is set to TRUE. Depending on the status of the production line, the different commands for the mode can now be enabled or disabled via the input parameters.

In the info structure of MpPackML, you can see that mode "Production" has been started.

1.3 Configuration

1.3.1 MpPackMLCore configuration

id_Core
Settings used here are applied in all modes.

id_Modes
Allow IN: Defines whether the transition to this mode is allowed in the respective state or not
Allow OUT: Defines whether the transition to another mode is allowed in the respective state or not
General settings

General settings can be made in the first section of the configuration.

"Startup active mode" defines which mode is enabled automatically after startup. The mode ID defined in section "Modes" is specified for this. "Warning state transfer not allowed" and "Warning: mode transfer not allowed" can define whether a warning should appear on an illegal command or change in mode.

"Mode activation command level" defines whether the change in mode is triggered by a rising edge. Parameter "Startup active state" can be used to define the state after startup.

"PackTags handling" defines whether structure "PackTags" should be used. For more information, see Using the PackML PackTags structure.

"Auditing" defines whether events triggered in MpPackML should be recorded. For more information, see Auditing the PackML state.

Modes

Which modes should be used is defined in section "Modes". "Allow IN" defines whether the transition to this mode is allowed in the respective state. "Allow OUT" defines whether the transition to another mode is allowed in the respective state.

For more information about individual modes, see PackML on Wikipedia.

1.4 Use cases

This section outlines several possible use cases for MpPackML components.

1.4.1 Use case 1: Packaging machine

Requirement

A packaging machine is part of a complete production line. It is important that all machines taking part in the line can forward necessary information to the operator or shift supervisor in a standardized form. This should be achieved by using the OMAC PackML standard. For this reason, we will use the MpPackML mapp component.

Solution

Component list

- MpPackMLCore (own MpLink): Central PackML core
- MpPackMLMode (MpLink from MpPackMLCore): Controls the PackML mode, in this case the production mode
- MpPackMLMode (MpPackMLMode from MpPackMLCore): Controls the PackML mode, in this case the maintenance mode

Connection diagram

Using the mapp components

The machine is controlled using various modes. The MpPackMLCore component is the system core that manages the transitions between the modes. The following modes are needed for the packaging machine:

- Production: This mode is running while the machine, as part of the entire production line, is packaging products. All processes are automated.
- Maintenance: This mode is running while the machine is being serviced. The mechanical components of the machine can be controlled individually.
Each mode has its own state machine. For example, the machine can be put into the Execution state in *Production* mode, which means that products are being packaged. Another state is *Complete*. Here the last products are packaged after the machine has stopped.

The MpLink of the core and a unique ID (number) must be assigned to the **MpPackMLMode** component. The mode can then be enabled and the state changed in the PackML state model using standard commands.

**Helpful tips**

*Define expressive names*

When searching for errors with WebXs, it will become clear why good naming conventions are advantageous:

- The core receives its name from MpLink.
- The mode receives its name from the function block instance.

**One program for the core and two additional programs for the modes:**

This structure facilitates easy error localization. Within the program, the user can link the MpPackMLMode function block instance with the corresponding control signals for machine operation.

MpPackMLMode only accepts commands if the mode is active. This makes managing the control logic much easier.

**One ACTION call for each state in the mode program**
This practice helps in error localization and organizes the program code belonging to the individual mode states:

- The number of programs in the software configuration remains manageable.
- Possible to use local variables for the current mode

An ACTION can also be added as a reference to existing files. This can be useful if the same code should be used in several modes, e.g. resetting, clearing, etc.

The code within the ACTION can be written as follows:

The ACTION is called in one cycle and the user code is only executed if the state is actually active. A state machine within the mode state is useful and advantageous for structured programming.

There are two ways out of this state:

- **Finished state**: Calls function MpPackMLStateComplete(…)
- **Abort**: Calls function MpPackMLAbort(…)

ModelID can be set to ALL_MODES (0) if the code should be used by several modes (this ensures modularity):
1.5 Libraries

1.5.1 Function blocks

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MpPackMLAbort</td>
<td>This function triggers a PackML ABORT command.</td>
</tr>
<tr>
<td>MpPackMLBasicUI</td>
<td>This function block creates a connection to Visual Components 4 HMI applications in order to display and check the mode and state diagram.</td>
</tr>
<tr>
<td>MpPackMLCore</td>
<td>This function block forms the core of PackML logic and can be used to manage all modes.</td>
</tr>
<tr>
<td>MpPackMLMode</td>
<td>This function block manages a PackML mode.</td>
</tr>
<tr>
<td>MpPackMLModeCurrent</td>
<td>This function returns the current mode as an ID.</td>
</tr>
<tr>
<td>MpPackMLModeIsActive</td>
<td>This function indicates whether the selected mode is active.</td>
</tr>
<tr>
<td>MpPackMLStateComplete</td>
<td>This function triggers a PackML STATE COMPLETE command.</td>
</tr>
<tr>
<td>MpPackMLStateCurrent</td>
<td>This function returns the current state.</td>
</tr>
<tr>
<td>MpPackMLStateIsActive</td>
<td>This function indicates whether the selected state is active.</td>
</tr>
<tr>
<td>MpPackMLStatisticsUI</td>
<td>This function block creates a connection to Visual Components 4 HMI applications in order to visualize statistical data regarding the usage of modes and states.</td>
</tr>
</tbody>
</table>

1.5.1.1 MpPackMLAbort

This function triggers a PackML ABORT command.

**Function**

```
MpComIdentType MpLink
DINT Mode

MpPackMLAbort

BOOL Return value
```

**Interface**

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>Mode</td>
<td>DINT</td>
<td>Mode in which the command should be executed (0 = the command can be executed in all modes).</td>
</tr>
<tr>
<td>OUT</td>
<td>ReturnValue</td>
<td>BOOL</td>
<td>Function executing ABORT command. The return value is TRUE if the command was accepted.</td>
</tr>
</tbody>
</table>

**mapp concept**

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.1.1 Description

The same MpLink used by MpPackMLCore is used for this component as well.

This function can be used to trigger an ABORT command remotely. The mode is defined using input "Mode". This is specified via the corresponding index. If 0 is used as the index, the ABORT command applies to all modes.

If the ABORT command is accepted, then TRUE is given as the return value.

1.5.1.2 MpPackMLBasicUI

This function block creates a connection to Visual Components 4 HMI applications in order to display and check the mode and state diagram.
mapp Services

Function block

<table>
<thead>
<tr>
<th>Interface</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>Pointer to MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>Enable</td>
<td>BOOL</td>
<td>The function block is active as long as this input is set.</td>
</tr>
<tr>
<td>IN</td>
<td>ErrorReset</td>
<td>BOOL</td>
<td>Resets function block errors.</td>
</tr>
<tr>
<td>IN</td>
<td>UIConnect</td>
<td>Pointer to MpPackMLBasicUIConnectType</td>
<td>This structure contains the parameters needed for the connection to the HMI application.</td>
</tr>
<tr>
<td>OUT</td>
<td>Active</td>
<td>BOOL</td>
<td>Function block active.</td>
</tr>
<tr>
<td>OUT</td>
<td>Error</td>
<td>BOOL</td>
<td>Error occurred during execution.</td>
</tr>
<tr>
<td>OUT</td>
<td>StatusID</td>
<td>DINT</td>
<td>Status information.</td>
</tr>
<tr>
<td>OUT</td>
<td>Info</td>
<td>MpPackMLUIInfoType</td>
<td>Additional information about the component.</td>
</tr>
</tbody>
</table>

mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.2.1 Description

The same MpLink used by MpPackMLCore is used for this component as well. This component displays the PackML state diagram as well as the mode on the VC4-based HMI application.

MpPackMLBasicUIConnectType

Structure "UIConnect" is divided into the following areas:

- **StateControl:** Contains the various commands for navigating between the different states. It is also possible to use this substructure to control the state diagram using colors.
- **ModeControl:** Makes it possible to change the mode using a drop-down menu (if permitted).
- **ModeCurrent:** Displays the current mode.
- **StateCurrent:** Displays the current state.

1.5.1.3 MpPackMLCore

This function block forms the core of PackML logic and can be used to manage all modes.

Function block
### Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>Pointer to MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration)</td>
</tr>
<tr>
<td>IN</td>
<td>Enable</td>
<td>BOOL</td>
<td>The function block is active as long as this input is set.</td>
</tr>
<tr>
<td>IN</td>
<td>ErrorReset</td>
<td>BOOL</td>
<td>Resets function block errors.</td>
</tr>
<tr>
<td>OUT</td>
<td>Active</td>
<td>BOOL</td>
<td>Function block active.</td>
</tr>
<tr>
<td>OUT</td>
<td>Error</td>
<td>BOOL</td>
<td>Error occurred during execution.</td>
</tr>
<tr>
<td>OUT</td>
<td>StatusID</td>
<td>DINT</td>
<td>Status information.</td>
</tr>
<tr>
<td>OUT</td>
<td>ModeCurrent</td>
<td>DINT</td>
<td>Current mode.</td>
</tr>
<tr>
<td>OUT</td>
<td>StateCurrent</td>
<td>MpPackMLStateEnum</td>
<td>Current state, depends on the active mode.</td>
</tr>
<tr>
<td>OUT</td>
<td>Info</td>
<td>MpPackMLCoreInfoType</td>
<td>Additional information about the component.</td>
</tr>
</tbody>
</table>

### mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

#### 1.5.1.3.1 Description

The MpLink of an MpPackMLCore configuration is used for this function block. This mapp component forms the core necessary to use the PackML standard. It is used to manage the different modes and the transitions between them. The current mode is returned as an index on output "ModeCurrent", and the current state is checked on output "StateCurrent". Mode and state are also stored as text in the info structure.

#### 1.5.1.4 MpPackMLMode

This function block manages a PackML mode.

### Function block

![MpPackMLMode block diagram]

### Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>Pointer to MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>Enable</td>
<td>BOOL</td>
<td>The function block is active as long as this input is set.</td>
</tr>
<tr>
<td>IN</td>
<td>ErrorReset</td>
<td>BOOL</td>
<td>Resets function block errors.</td>
</tr>
<tr>
<td>IN</td>
<td>ModelID</td>
<td>DINT</td>
<td>Unique identification number for each mode.</td>
</tr>
</tbody>
</table>

Data sheet V 15
mapp Services

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Activate</td>
<td>BOOL</td>
<td>Enables this mode, if allowed.</td>
</tr>
<tr>
<td>IN</td>
<td>Start</td>
<td>BOOL</td>
<td>PackML START command.</td>
</tr>
<tr>
<td>IN</td>
<td>Stop</td>
<td>BOOL</td>
<td>PackML STOP command.</td>
</tr>
<tr>
<td>IN</td>
<td>Reset</td>
<td>BOOL</td>
<td>PackML RESET command.</td>
</tr>
<tr>
<td>IN</td>
<td>Hold</td>
<td>BOOL</td>
<td>PackML HOLD command.</td>
</tr>
<tr>
<td>IN</td>
<td>Unhold</td>
<td>BOOL</td>
<td>PackML UNHOLD command.</td>
</tr>
<tr>
<td>IN</td>
<td>Suspend</td>
<td>BOOL</td>
<td>PackML SUSPEND command.</td>
</tr>
<tr>
<td>IN</td>
<td>Unsuspend</td>
<td>BOOL</td>
<td>PackML UNSUSPEND command.</td>
</tr>
<tr>
<td>IN</td>
<td>Abort</td>
<td>BOOL</td>
<td>PackML ABORT command.</td>
</tr>
<tr>
<td>IN</td>
<td>Clear</td>
<td>BOOL</td>
<td>PackML CLEAR command.</td>
</tr>
<tr>
<td>IN</td>
<td>StateComplete</td>
<td>BOOL</td>
<td>PackML STATE COMPLETE command.</td>
</tr>
<tr>
<td>OUT</td>
<td>Active</td>
<td>BOOL</td>
<td>Function block active.</td>
</tr>
<tr>
<td>OUT</td>
<td>Error</td>
<td>BOOL</td>
<td>Error occurred during execution.</td>
</tr>
<tr>
<td>OUT</td>
<td>StatusID</td>
<td>DINT</td>
<td>Status information.</td>
</tr>
<tr>
<td>OUT</td>
<td>ChangeModeAllowed</td>
<td>BOOL</td>
<td>Indicates whether changing the mode is allowed.</td>
</tr>
<tr>
<td>OUT</td>
<td>ModeActive</td>
<td>BOOL</td>
<td>Indicates whether this mode is active.</td>
</tr>
<tr>
<td>OUT</td>
<td>StateCurrent</td>
<td>MpPackMLStateEnum</td>
<td>Current state.</td>
</tr>
<tr>
<td>OUT</td>
<td>Info</td>
<td>MpPackMLModeInfoType</td>
<td>Additional information about the component.</td>
</tr>
</tbody>
</table>

mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.4.1 Description

The same MpLink used by MpPackMLCore is used for this component as well.

The different modes can be configured in Automation Studio or via the WebXs.

This function block controls a mode. One MpPackMLMode instance is necessary for each mode. The index of the mode must be defined using input "ModeID". It is important to ensure that each mode has a unique index. The mode can be enabled using command "Activate" command as long as output "ChangeModeAllowed" is set to TRUE. Whether the mode being managed by this component is active or not can be determined using output "ModeActive".

As soon as the mode is active, it is possible to switch between individual states using the various standard commands. The current state is indicated by output "StateCurrent".

Both the mode and state are also specified as text in the info structure.

1.5.1.5 MpPackMLModeCurrent

This function returns the current mode as an ID.

Function

```
MpComIdentType
MpLink
MpPackMLModeCurrent
Return value
DINT
```

Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>OUT</td>
<td>ReturnValue</td>
<td>DINT</td>
<td>Returns the current mode as an ID.</td>
</tr>
</tbody>
</table>

mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).
1.5.1.5.1 Description
The same MpLink used by MpPackMLCore is used for this component as well. This function returns the index of the current mode.

1.5.1.6 MpPackMLModelsActive
This function indicates whether the selected mode is active.

Function

```
MpComIdentType MpPackMLModelsActive
MpLink
DINT Mode
BOOL Return value
```

Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>Mode</td>
<td>DINT</td>
<td>Mode being checked.</td>
</tr>
<tr>
<td>OUT</td>
<td>ReturnValue</td>
<td>BOOL</td>
<td>Indicates whether the mode being checked is active.</td>
</tr>
</tbody>
</table>

mapp concept
Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.6.1 Description
The same MpLink used by MpPackMLCore is used for this component as well.
This function uses the index to determine whether the mode defined on input "Mode" is active.
If the mode being checked is active, then TRUE is given as the return value.

1.5.1.7 MpPackMLStateComplete
This function triggers a PackML STATE COMPLETE command.

Function

```
MpComIdentType MpPackMLStateComplete
MpLink
DINT Mode
BOOL Return value
```

Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>Mode</td>
<td>DINT</td>
<td>Mode in which the command should be executed (0 = the command can be executed in all modes).</td>
</tr>
<tr>
<td>OUT</td>
<td>ReturnValue</td>
<td>BOOL</td>
<td>Function executing STATE COMPLETE command. The return value is TRUE if the command was accepted.</td>
</tr>
</tbody>
</table>

mapp concept
Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).
1.5.1.7.1 Description

The same MpLink used by MpPackMLCore is used for this component as well.

This function can be used to trigger a STATE COMPLETE command remotely. The mode is defined using input "Mode". This is specified via the corresponding index. If 0 is used as the index, the STATE COMPLETE command applies to all modes.

If the STATE COMPLETE command is accepted, then TRUE is given as the return value.

1.5.1.8 MpPackMLStateCurrent

This function returns the current state.

Function

```
MpComIdentType MpPackMLStateCurrent
MpLink
MpPackMLStateEnum
```

### Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>OUT</td>
<td>ReturnValue</td>
<td>MpPackMLStateEnum</td>
<td>Returns the current state.</td>
</tr>
</tbody>
</table>

### mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.8.1 Description

The MpLink for this function comes from MpPackMLCore.

This function returns the state of the current mode.

1.5.1.9 MpPackMLStateIsActive

This function indicates whether the selected state is active.

Function

```
MpComIdentType MpPackMLStateIsActive
MpLink
MpPackMLStateEnum
DINT Mode
BOOL ReturnValue
```

### Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>State</td>
<td>MpPackMLStateEnum</td>
<td>State to be checked.</td>
</tr>
<tr>
<td>IN</td>
<td>Mode</td>
<td>DINT</td>
<td>Mode in which the command should be executed (0 = the command can be executed in all modes).</td>
</tr>
<tr>
<td>OUT</td>
<td>ReturnValue</td>
<td>BOOL</td>
<td>Indicates whether the state being checked is active.</td>
</tr>
</tbody>
</table>

### mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).
For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.9.1 Description

The same MpLink used by MpPackMLCore is used for this component as well.

This function determines whether the state defined on input "State" is active. The mode can also be specified via "Mode". This makes it possible to query a certain state in a certain mode. If 0 is specified as the index for the mode, then all modes are checked.

If the state being checked is active, then TRUE is given as the return value.

1.5.1.10 MpPackMLStatisticsUI

This function block creates a connection to Visual Components 4 HMI applications in order to visualize statistical data regarding the usage of modes and states.

Function block

```
&MpComIdentType MpLink
BOOL Enable
BOOL ErrorReset
&MpPackMLStatisticsUIConnectType UIConnect
BOOL Active
BOOL Error
DINT StatusID
MpPackMLUIInfoType Info
```

Interface

<table>
<thead>
<tr>
<th>I/O</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>MpLink</td>
<td>Pointer to MpComIdentType</td>
<td>Connection to mapp (MpLink of an MpPackMLCore configuration).</td>
</tr>
<tr>
<td>IN</td>
<td>Enable</td>
<td>BOOL</td>
<td>The function block is active as long as this input is set.</td>
</tr>
<tr>
<td>IN</td>
<td>ErrorReset</td>
<td>BOOL</td>
<td>Resets function block errors.</td>
</tr>
<tr>
<td>IN</td>
<td>UIConnect</td>
<td>Pointer to MpPackMLStatisticsUIConnectType</td>
<td>This structure contains the parameters needed for the connection to the HMI application.</td>
</tr>
<tr>
<td>OUT</td>
<td>Active</td>
<td>BOOL</td>
<td>Function block active.</td>
</tr>
<tr>
<td>OUT</td>
<td>Error</td>
<td>BOOL</td>
<td>Error occurred during execution.</td>
</tr>
<tr>
<td>OUT</td>
<td>StatusID</td>
<td>DINT</td>
<td>Status information.</td>
</tr>
<tr>
<td>OUT</td>
<td>Info</td>
<td>MpPackMLUIInfoType</td>
<td>Additional information about the component.</td>
</tr>
</tbody>
</table>

mapp concept

Section mapp components explains how mapp components are structured. In addition, it provides important notes for correctly using mapp components (e.g. for downloads).

For mapp function blocks, asynchronous handling does not have to be carried out in the initialization subroutine or in an acyclic task. However, an appropriately high stack must be configured in acyclic tasks (~6 kB).

1.5.1.10.1 Description

The same MpLink used by MpPackMLCore is used for this component as well. This component manages statistical data regarding the usage of the various states and modes.

MpPackMLStatisticsUIConnectType

Structure "UIConnect" is divided into the following areas:

- **States**: Contains statistical data regarding all states.
- **Mode**: Contains statistical data regarding the mode selected using "ModeID".
- **ModeCurrent**: Displays the current mode.
- **StateCurrent**: Displays the current state.
1.5.2 Data types and enumerators

1.5.2.1 Data types

1.5.2.1.1 MpPackMLBasicUIConnectType

This data type contains all of the information necessary to establish a connection to the HMI application.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateControl</td>
<td>MpPackMLBasicUIControlType</td>
<td>Contains the commands for switching between states as well as the necessary parameters for displaying the states using colors.</td>
</tr>
<tr>
<td>ModeControl</td>
<td>MpPackMLBasicUIControlType</td>
<td>Makes it possible to switch between various modes.</td>
</tr>
<tr>
<td>ModeCurrent</td>
<td>DINT</td>
<td>Displays the current mode</td>
</tr>
<tr>
<td>StateCurrent</td>
<td>MpPackMLStateEnum</td>
<td>Display the current state.</td>
</tr>
</tbody>
</table>

VC4 connection:
- IndexDatapoint from Text (text group for PackML)

1.5.2.1.2 MpPackMLBasicUIModeControlType

This data type can be used to switch the mode.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SelectedIndex</td>
<td>UINT</td>
<td>Index of the entry currently selected in the list</td>
</tr>
<tr>
<td>MaxSelection</td>
<td>UDINT</td>
<td>Index of the last entry in the list</td>
</tr>
<tr>
<td>LockActivation</td>
<td>BOOL</td>
<td>Locks the &quot;Activate&quot; command if the selected mode is not available</td>
</tr>
<tr>
<td>Activate</td>
<td>BOOL</td>
<td>Enables the mode selected using &quot;SelectedIndex&quot;</td>
</tr>
</tbody>
</table>

VC4 connection:
- IndexDatapoint from Dropdown Listbox
- MaxDatapoint from Dropdown Listbox
- LockingDatapoint from Button / Locking: Datapoint >= Level / Level: 1
- Datapoint from Button / Type: SetDatapoint / SetValue: 1

1.5.2.1.3 MpPackMLBasicUIStateControlType

Commands for switching between various states.
### Parameter | Data type | Description
---|---|---
Start | BOOL | PackML START command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Stop | BOOL | PackML STOP command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Reset | BOOL | PackML RESET command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Hold | BOOL | PackML HOLD command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Unhold | BOOL | PackML UNHOLD command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Suspend | BOOL | PackML SUSPEND command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Unsuspend | BOOL | PackML UNSUSPEND command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Abort | BOOL | PackML ABORT command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
Clear | BOOL | PackML CLEAR command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
StateComplete | BOOL | PackML STATE COMPLETE command  
VC4 connection: Daptpoint from Button / Type: SetDatapoint / SetValue: 1
ColorIndex | MpPackMLColorIndexStatesType | Structure for managing the color and blinking for the current state

#### 1.5.2.1.4 MpPackMLColorIndexStatesType

This data type is used to manage the color display and blinking of the PackML state diagram.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Clearing | USINT | Color for the Clearing state  
VC4 connection: ColorDatapoint from Text |
| Stopped | USINT | Color for the Stopped state  
VC4 connection: ColorDatapoint from Text |
| Starting | USINT | Color for the Starting state  
VC4 connection: ColorDatapoint from Text |
| Idle | USINT | Color for the Idle state  
VC4 connection: ColorDatapoint from Text |
| Suspended | USINT | Color for the Suspended state  
VC4 connection: ColorDatapoint from Text |
| Execute | USINT | Color for the Execute state  
VC4 connection: ColorDatapoint from Text |
| Stopping | USINT | Color for the Stopping state  
VC4 connection: ColorDatapoint from Text |
| Aborting | USINT | Color for the Aborting state  
VC4 connection: ColorDatapoint from Text |
| Aborted | USINT | Color for the Aborted state  
VC4 connection: ColorDatapoint from Text |
| Holding | USINT | Color for the Holding state  
VC4 connection: ColorDatapoint from Text |
### Info

A "color map" must be created in the HMI application in order to be able to display individual states in color:

- **Held**
  - Data type: USINT
  - Description: Color for the **Held** state. VC4 connection: ColorDatapoint from Text

- **Unholding**
  - Data type: USINT
  - Description: Color for the **Unholding** state. VC4 connection: ColorDatapoint from Text

- **Suspending**
  - Data type: USINT
  - Description: Color for the **Suspending** state. VC4 connection: ColorDatapoint from Text

- **Unsuspending**
  - Data type: USINT
  - Description: Color for the **Unsuspending** state. VC4 connection: ColorDatapoint from Text

- **Resetting**
  - Data type: USINT
  - Description: Color for the **Resetting** state. VC4 connection: ColorDatapoint from Text

- **Completing**
  - Data type: USINT
  - Description: Color for the **Completing** state. VC4 connection: ColorDatapoint from Text

- **Complete**
  - Data type: USINT
  - Description: Color for the **Complete** state. VC4 connection: ColorDatapoint from Text

### 1.5.2.1.5 MpPackMLCoreInfoType

This data type provides additional information for the MpPackMLCore component.
### 1.5.2.1.6 MpPackMLDiagType

This data type is used as a substructure within the structure to hold additional information for diagnostic purposes as well as to supply additional data about the status ID.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatusID</td>
<td>MpPackMLStatusIDType</td>
<td>StatusID diagnostic structure</td>
</tr>
</tbody>
</table>

### 1.5.2.1.7 MpPackMLModeInfoType

This data type provides additional information for the MpPackMLMode component.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModeCurrent</td>
<td>STRING[50]</td>
<td>Current mode</td>
</tr>
<tr>
<td>StateCurrent</td>
<td>STRING[20]</td>
<td>Current state</td>
</tr>
<tr>
<td>ActivationAborted</td>
<td>BOOL</td>
<td>Displayed if this mode could not be enabled after an &quot;Activate&quot; command</td>
</tr>
<tr>
<td>Diag</td>
<td>MpPackMLDiagType</td>
<td>Diagnostic structure for the status ID</td>
</tr>
</tbody>
</table>

### 1.5.2.1.8 MpPackMLStatisticsUIConnectType

This data type contains all of the information necessary to establish a connection to the HMI application.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>States</td>
<td>MpPackMLStatisticsUIStatusType</td>
<td>Contains statistical data for the individual states</td>
</tr>
<tr>
<td>Mode</td>
<td>MpPackMLStatisticsUIModeType</td>
<td>Contains statistical data for the selected mode</td>
</tr>
<tr>
<td>ModeCurrent</td>
<td>DINT</td>
<td>Displays the current mode</td>
</tr>
<tr>
<td>StateCurrent</td>
<td>MpPackMLStateEnum</td>
<td>Display the current state.</td>
</tr>
</tbody>
</table>

### 1.5.2.1.9 MpPackMLStatisticsUIModeType

This data type can be used to select a mode for which statistical data can be displayed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModeID</td>
<td>DINT</td>
<td>Mode for which statistical data should be displayed. If 0 is entered, the data for the currently active mode is displayed.</td>
</tr>
<tr>
<td>CumulativeTime</td>
<td>MpPackMLStatisticsUITimeType</td>
<td>Total runtime of the current mode [seconds]</td>
</tr>
<tr>
<td>CumulativePercent</td>
<td>REAL</td>
<td>Total runtime of the current mode in relation to the total runtime of the machine [%]</td>
</tr>
<tr>
<td>CurrentTime</td>
<td>MpPackMLStatisticsUITimeType</td>
<td>Runtime of the current mode since its activation [seconds]</td>
</tr>
<tr>
<td>CurrentPercent</td>
<td>REAL</td>
<td>Runtime of the current mode since its activation in relation to the total runtime of this mode [%]</td>
</tr>
</tbody>
</table>
Info:
In order for parameter "CumulativePercent" to be displayed, you must create a text snippet (MpPackMLStatisticsUIConnect.Mode.CumulativePercent).

The created text snippet is used in the MpPackMLStatisticsText text group.

**Figure 2: Text snippet**

**Figure 3: MpPackMLStatisticsText text group**

1.5.2.1.10 MpPackMLStatisticsUIStatesType
This data type returns statistical data regarding all states.
### Parameter Data type Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clearing</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Clearing state</td>
</tr>
<tr>
<td>Stopped</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Stopped state</td>
</tr>
<tr>
<td>Starting</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Starting state</td>
</tr>
<tr>
<td>Idle</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Idle state</td>
</tr>
<tr>
<td>Suspended</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Suspended state</td>
</tr>
<tr>
<td>Execute</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Execute state</td>
</tr>
<tr>
<td>Stopping</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Stopping state</td>
</tr>
<tr>
<td>Aborting</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Aborting state</td>
</tr>
<tr>
<td>Aborted</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Aborted state</td>
</tr>
<tr>
<td>Holding</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Holding state</td>
</tr>
<tr>
<td>Held</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Held state</td>
</tr>
<tr>
<td>Unholding</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Unholding state</td>
</tr>
<tr>
<td>Suspending</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Suspending state</td>
</tr>
<tr>
<td>Unsuspending</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Unsuspending state</td>
</tr>
<tr>
<td>Resetting</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Resetting state</td>
</tr>
<tr>
<td>Completing</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Completing state</td>
</tr>
<tr>
<td>Complete</td>
<td>MpPackMLStatisticsUIStateType</td>
<td>Statistical data for the Complete state</td>
</tr>
<tr>
<td>ResetTimers</td>
<td>BOOL</td>
<td>Resets the timers for all states and modes</td>
</tr>
</tbody>
</table>

**1.5.2.11 MpPackMLStatisticsUIStateType**

This data type returns statistical data regarding a state.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CumulativeTime</td>
<td>MpPackMLStatisticsUITimeType</td>
<td>Total runtime of the current state [seconds]</td>
</tr>
<tr>
<td>CumulativePercent</td>
<td>REAL</td>
<td>Total runtime of the current state in relation to the total runtime of the machine [%]</td>
</tr>
<tr>
<td>VC4 connection: Datapoint from Button / Type: MomentaryDatapoint/ Setvalue: 1 ResetValue: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CurrentTime</td>
<td>MpPackMLStatisticsUITimeType</td>
<td>Runtime of the current state since its activation [seconds]</td>
</tr>
<tr>
<td>CurrentPercent</td>
<td>REAL</td>
<td>Runtime of the current state since its activation in relation to the total runtime of this state [%]</td>
</tr>
</tbody>
</table>
In order for parameter "CumulativePercent" to be displayed, a text snippet must be created for each state (MpPackMLStatisticsUIConnect.States.Starting.CumulativePercent).

1.5.2.1.12 MpPackMLStatisticsUITimeType
This data type provides information about how long a state/mode has been active.
### Parameter Data type Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours</td>
<td>UDINT</td>
<td>Number of hours already active. <strong>VC4 connection:</strong> Datapoint from Numeric.</td>
</tr>
<tr>
<td>Minutes</td>
<td>USINT</td>
<td>Number of minutes already active in the current hour. <strong>VC4 connection:</strong> Datapoint from Numeric.</td>
</tr>
<tr>
<td>Seconds</td>
<td>USINT</td>
<td>Number of seconds already active in the current minute. <strong>VC4 connection:</strong> Datapoint from Numeric.</td>
</tr>
</tbody>
</table>

**Info:**

In order for the time of the respective state to be displayed, you must create text snippets (e.g. `MpPackMLStatisticsUIConnect.States.Stopped.CumulativeTime.Minutes`).

---

**Figure 6: Text snippet**

**Figure 7: MpPackMLStatisticsText text group**
1.5.2.13 MpPackMLStatusIDType

This data type is used as a substructure within the structure to hold additional information for diagnostic purposes as well as to supply additional data about the status ID.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>MpPackMLErrorEnum</td>
<td>Error code for mapp component</td>
</tr>
<tr>
<td>Severity</td>
<td>MpComSeveritiesEnum</td>
<td>Describes the type of information supplied by the status ID (success, information, warning, error)</td>
</tr>
<tr>
<td>Code</td>
<td>UINT</td>
<td>Code for the status ID. This error number can be used to search for additional information in the help system.</td>
</tr>
</tbody>
</table>

1.5.2.14 MpPackMLUIInfoType

This data type provides additional information about the MpPackMLBasicUI and MpPackMLStatisticsUI components.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModeCurrent</td>
<td>STRING[50]</td>
<td>Current mode</td>
</tr>
<tr>
<td>StateCurrent</td>
<td>STRING[20]</td>
<td>Current state</td>
</tr>
<tr>
<td>Diag</td>
<td>MpPackMLDiagType</td>
<td>Diagnostic structure for the status ID</td>
</tr>
</tbody>
</table>
1.5.2.2 Enumerators

1.5.2.2.1 MpPackMLPackTagsStatusEnum

This enumerated data type contains all possible states of the PackTags.

<table>
<thead>
<tr>
<th>Enumerator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpPACKML_PACKTAGS_DISABLED</td>
<td>Use of PackTags disabled</td>
</tr>
<tr>
<td>mpPACKML_PACKTAGS_INIT</td>
<td>PackTags initializing</td>
</tr>
<tr>
<td>mpPACKML_PACKTAGS_OK</td>
<td>PackTags initialized and ready for use</td>
</tr>
<tr>
<td>mpPACKML_PACKTAGS_ERROR_TAG</td>
<td>Initialization complete, but with errors. Required tag not found.</td>
</tr>
<tr>
<td>mpPACKML_PACKTAGS_ERROR_TYPE</td>
<td>Initialization complete, but with errors. Tag has incorrect data types</td>
</tr>
<tr>
<td>mpPACKML_PACKTAGS_ERROR_STRUCT</td>
<td>Initialization complete, but with errors. Configured structure not found</td>
</tr>
</tbody>
</table>

1.5.2.2.2 MpPackMLStateEnum

This enumerated data type contains all possible states of the PackML logic.

<table>
<thead>
<tr>
<th>Enumerator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpPACKML_STATE_UNDEFINED</td>
<td>The system is inactive, no state active</td>
</tr>
<tr>
<td>mpPACKML_STATE_CLEARING</td>
<td>PackML CLEARING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_STOPPED</td>
<td>PackML STOPPED state</td>
</tr>
<tr>
<td>mpPACKML_STATE_STARTING</td>
<td>PackML STARTING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_IDLE</td>
<td>PackML IDLE state</td>
</tr>
<tr>
<td>mpPACKML_STATE_SUSPENDED</td>
<td>PackML SUSPENDED state</td>
</tr>
<tr>
<td>mpPACKML_STATE_EXECUTE</td>
<td>PackML EXECUTE state</td>
</tr>
<tr>
<td>mpPACKML_STATE_STOPPING</td>
<td>PackML STOPPING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_ABORTING</td>
<td>PackML ABORTING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_ABORTED</td>
<td>PackML ABORTED state</td>
</tr>
<tr>
<td>mpPACKML_STATE_HOLDING</td>
<td>PackML HOLDING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_HELD</td>
<td>PackML HELD state</td>
</tr>
<tr>
<td>mpPACKML_STATE_UNHOLDING</td>
<td>PackML UNHOLDING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_SUSPENDING</td>
<td>PackML SUSPENDING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_UNSUSPENDING</td>
<td>PackML UNSUSPENDING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_RESETTING</td>
<td>PackML RESETING state</td>
</tr>
<tr>
<td>mpPACKML_STATE_COMPLETING</td>
<td>PackML COMPLETING state</td>
</tr>
</tbody>
</table>

1.5.3 Status numbers

1.5.3.1 1083474944: MpPackMLCore not ready

Description:
Function block MpPackMLCore is not active.

Reaction:
The function block indicates this information on output "StatusID".

Cause/Solution:
- In order to use MpPackMLMode, MpPackMLCore must be enabled ("Active = TRUE").

Constant

| mpPACKML_INF_CORE_NOT_READY       |

These function blocks / functions can report this error:
- MpPackMLMode

1.5.3.2 1083186315: Optional tags found

Description:
Optional PackTags process variable {2:PackTagsPV} found.
mapp Services

Additional information

- {2:PackTagsPV}: Process variable "PackTags"

**Constant**

mpPACKML_INF_TAGS_OPTIONAL_FOUND

**These function blocks / functions can report this error:**

- MpPackMLCore

**1.5.3.3 -2137750523: Command not permitted**

**Description:**
This command is not currently permitted.

**Reaction:**
The function block indicates an active warning on output "StatusID".

**Cause/Solution:**

- Check the PackML state.

**Constant**

mpPACKML_WRN_CMD_NOT_ALLOWED

**These function blocks / functions can report this error:**

- MpPackMLMode

**1.5.3.4 -2137750522: Enabling currently not possible**

**Description:**
It is not currently possible to enable this mode.

**Reaction:**
The function block indicates an active warning on output "StatusID".

**Cause/Solution:**

- Check the PackML state diagram and the current mode.

**Constant**

mpPACKML_WRN_ACTIV_NOT_ALLOWED

**These function blocks / functions can report this error:**

- MpPackMLMode

**1.5.3.5 -2137750519: PackTags structure not found**

**Description:**
The specified PackTags structure was not found.

**Reaction:**
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.
Cause/Solution:
- Check the configured PackTags structure.
- Create the PackTags structure.

Constant
mpPACKML_WRN_TAGS_STR_NOT_FOUND

These function blocks / functions can report this error:
- MpPackMLCore

1.5.3.6 -2137750518: PackTags PV not found

Description:
The specified PackTags PV is required and was not found.

Reaction:
The function block indicates an active error on outputs “StatusID” and “Error”. No other functions are available during this time.

Cause/Solution:
- Check the configured PackTags PV.
- Create the PackTags PV.

Constant
mpPACKML_WRN_TAGS_NOT_FOUND

These function blocks / functions can report this error:
- MpPackMLCore

1.5.3.7 -2137750517: Incorrect PackTags data type for tag

Description:
Incorrect specification of the PackTags data type for the tag.

Reaction:
The function block indicates an active error on outputs “StatusID” and “Error”. No other functions are available during this time.

Cause/Solution:
- Check the PackTags data type.

Constant
mpPACKML_WRN_WRONG_TAGS_TYPE

These function blocks / functions can report this error:
- MpPackMLCore

1.5.3.8 -1064297340: Error creating mode configuration

Description:
Error creating the mode configuration at runtime. Error source: {1:ErrorNumber}
mapp Services

Additional information

- `{1:ErrorNumber}`: Reason for the error.

Cause/Solution:

- Check the configuration in section `Modes`.
- Use `MpComDump` to create a file that contains all information about the components and services being used. Save the Logger information as well. Contact B&R Support with this information.

Constant

`mpPACKML_ERR_MODE_CONFIG_CREATE`

These function blocks / functions can report this error:

- `MpPackMLCore`

1.5.3.9 -1064297339: No modes found

Description:
No modes could be found in the current configuration. At least one `1` mode is required.

Cause/Solution:

- Create a mode in the configuration.

Constant

`mpPACKML_ERR_NO_MODES_CONFIGURED`

These function blocks / functions can report this error:

- `MpPackMLCore`

1.5.3.10 -1064297337: Error reading configuration

Description:
Error reading mode configuration after a change. Error source: `{1:ErrorNumber}`

Additional information

- `{1:ErrorNumber}`: Reason for the error.

Cause/Solution:

- Check the specified modes in the configuration.

Constant

`mpPACKML_ERR_READ_MODE_CFG_CHANGE`

These function blocks / functions can report this error:

- `MpPackMLCore`

1.5.3.11 -1064297336: Error registering PackMLCore service

Description:
Error `{2:SysError}` registering the PackMLCore service.

Additional information

- `{2:SysError}`: Error that occurred while registering the service
Cause/Solution:

- Use MpComDump to create a file that contains all information about the components and services being used. Save the Logger information as well. Contact B&R Support with this information.

Constant

mpPACKML_ERR_REG_SERVICE

These function blocks / functions can report this error:

- MpPackMLCore

1.5.3.12 -1064297335: Error registering internal PackML service

Description:

Error (2:SysError) registering the internal PackML service.

Additional information

- {2:SysError}: Error that occurred while registering the internal service

Cause/Solution:

- Use MpComDump to create a file that contains all information about the components and services being used. Save the Logger information as well. Contact B&R Support with this information.

Constant

mpPACKML_ERR_REG_INTERNAL_SERVICE

These function blocks / functions can report this error:

- MpPackMLCore

1.5.3.13 -1064297334: Service PV not found

Description:

The service process variable was not found. It is needed to allow display of the modes with an ID and name of the function block in WebXs.

Cause/Solution:

- Use MpComDump to create a file that contains all information about the components and services being used. Save the Logger information as well. Contact B&R Support with this information.

Constant

mpPACKML_ERR_SERVICE_PV_NOT_FOUND

These function blocks / functions can report this error:

- MpPackMLCore

1.5.3.14 -1064239103: Could not create component

Description:

The mapp component could not be created and is not enabled. See the Logger for additional information.

Reaction:

The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.
Cause/Solution:

- Service for configuring the function block not available -> Problem with MpCom
- Unable to read registry -> Problem with MpCom
- Details about the cause of error in the logger

These function blocks / functions can report this error:

- MpPackMLAbort
- MpPackMLCore
- MpPackMLMode
- MpPackMLModeCurrent
- MpPackMLModelsActive
- MpPackMLStateComplete
- MpPackMLStateCurrent
- MpPackMLStatelsActive

Constant

mpPACKML_ERR_ACTIVATION

1.5.3.15 -1064239102: MpLink is null pointer

Description:
Input "MpLink" is not connected, null pointer.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:

- Check input "MpLink" on the function block.

These function blocks / functions can report this error:

- MpPackMLAbort
- MpPackMLCore
- MpPackMLMode
- MpPackMLModeCurrent
- MpPackMLModelsActive
- MpPackMLStateComplete
- MpPackMLStateCurrent
- MpPackMLStatelsActive

Constant

mpPACKML_ERR_MPLINK_NULL

1.5.3.16 -1064239101: MpLink connection not permitted

Description:
The value on input "MpLink" is not allowed.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.
Cause/Solution:

- The component is connected directly to mpCOM_MAIN or mpCOM_STANDALONE. This is not supported.

These function blocks / functions can report this error:

- MpPackMLAbort
- MpPackMLCore
- MpPackMLMode
- MpPackMLModeCurrent
- MpPackMLStateIsActive
- MpPackMLStateComplete
- MpPackMLStateCurrent
- MpPackMLStateIsActive

Constant

mpPACKML_ERR_MPLINK_INVALID

1.5.3.17 -1064239100: MpLink modified

Description:
The value on input "MpLink" was modified while the components were running ("Enable" = TRUE).

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:

- The value of input "MpLink" can only be changed while the component is inactive ("Enable" = FALSE).

These function blocks / functions can report this error:

- MpPackMLAbort
- MpPackMLCore
- MpPackMLMode
- MpPackMLModeCurrent
- MpPackMLStateIsActive
- MpPackMLStateComplete
- MpPackMLStateCurrent
- MpPackMLStateIsActive

Constant

mpPACKML_ERR_MPLINK_CHANGED

1.5.3.18 -1064239099: Invalid MpLink contents

Description:
The value of variable "MpLink" on the function block input is invalid.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.
mapp Services

Cause/Solution:

- Writing to the values in structure MpComIdentType is not permitted.
- A corresponding configuration for this mapp component must be available (see first paragraph of the description for the functions / function blocks listed below).

These function blocks / functions can report this error:

- MpPackMLAbort
- MpPackMLCore
- MpPackMLMode
- MpPackMLModeCurrent
- MpPackMLModeIsActive
- MpPackMLStateComplete
- MpPackMLStateCurrent
- MpPackMLStatelsActive

Constant

mpPACKML_ERR_MPLINK_CORRUPT

1.5.3.19 -1064239098: MpLink already in use

Description:
This MpLink is already in use.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:

- If "Overload" is the selected download mode, then the mapp component should be disabled in the exit subroutine with "Enable" = FALSE. It is also possible to use "Copy" or "One cycle" mode. In these cases, it is not necessary to disable the mapp component.
- Check which components are already using this MpLink.
- Create a new MpLink.

These function blocks / functions can report this error:

- MpPackMLAbort
- MpPackMLCore
- MpPackMLMode
- MpPackMLModeCurrent
- MpPackMLStateComplete
- MpPackMLStateCurrent
- MpPackMLStatelsActive
- MpPackMLModelsActive

Constant

mpPACKML_ERR_MPLINK_IN_USE

1.5.3.20 -1064239094: Error loading configuration

Description:
An error occurred while loading the configuration file (in the PV). See the Logger for additional information.
Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:

- Configuration file damaged
- Difference between the contents of the configuration file and the PV (e.g. different data types, data outside valid limits, etc.)

These function blocks / functions can report this error:

- MpPackMLCore

Constant

mpPACKML_ERR_CONFIG_LOAD

1.5.3.21 -1064239092: Error saving configuration

Description:
An error occurred while saving to the configuration file (from the PV). See the Logger for additional information.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:

- Configuration file damaged
- Difference between the contents of the configuration file and the PV (e.g. different data types, data outside valid limits, etc.)

These function blocks / functions can report this error:

- MpPackMLCore

Constant

mpPACKML_ERR_CONFIG_SAVE

1.5.3.22 -1064239091: Invalid configuration

Description:
Could not read configuration while creating components. See the Logger for additional information.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:

- The configuration is damaged.
- The function is not enabled in the configuration.

These function blocks / functions can report this error:

- MpPackMLMode

Constant

mpPACKML_ERR_CONFIG_INVALID
1.5.3.23 -1064008703: Cannot disable mode

Description:
Disabling the mode failed.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:
- While enabling the new mode, the previous mode could no longer be found.
- The new mode is enabled anyway.

Constant
mpPACKML_ERR_DEACTIVATION_FAIL

These function blocks / functions can report this error:
- MpPackMLMode

1.5.3.24 -1064008702: Invalid mode ID

Description:
The value on input "ModeID" is invalid.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:
- "ModeID" must be greater than 0.

Constant
mpPACKML_ERR_MODE_ID_INVALID

These function blocks / functions can report this error:
- MpPackMLMode

1.5.3.25 -1064008701: Invalid PV address

Description:
Internal error. Please contact B&R Support.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:
- Please contact B&R Support.

Constant
mpPACKML_ERR_PV_NAME_NULL

These function blocks / functions can report this error:
- MpPackMLMode
1.5.3.26 -1064008700: No valid PV address

Description:
Internal error. Please contact B&R Support.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:
- Please contact B&R Support.

Constant
mpPACKML_ERR_INVALID_PV_ADR

These function blocks / functions can report this error:
- MpPackMLMode

1.5.3.27 -1064008697: Mode ID already in use

Description:
This Mode ID is already in use.

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:
- Change the value on input "ModeID" (must be greater than 0 and unique for each mode).

Constant
mpPACKML_ERR_MODE_ID_USED

These function blocks / functions can report this error:
- MpPackMLMode

1.5.3.28 -1064008696: Missing value on UIConnect

Description:
NULL was appended to "UIConnect".

Reaction:
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

Cause/Solution:
- Input "UIConnect" forgotten

Constant
mpPACKML_ERR_MISSING_UI_CONNECT

These function blocks / functions can report this error:
- MpPackMLBasicUI
- MpPackMLStatisticsUI
1.5.3.29 -1064008695: Error reading configuration

**Description:**
Error reading configuration after a change. Error source: {1:ErrorNumber}

**Additional information**
- {1:ErrorNumber}: Reason for the error.

**Cause/Solution:**
- Check the changes made.
- If changes were made to the configuration via WebXs and then saved, the new values could not be read. In this case, use MpComDump to create a file that contains all information about the components and services being used. Save the Logger information as well. Contact B&R Support with this information.

**Constant**
mpPACKML_ERR_READ_CORE_CFG_CHANGE

**These function blocks / functions can report this error:**
- MpPackMLCore

1.5.3.30 -1064008692: Read configuration failed

**Description:**
Reading a section of the configuration failed.

**Reaction:**
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

**Cause/Solution:**
- Check the configuration.

**Constant**
mpPACKML_ERR_READ_MPPACKML_CFG

**These function blocks / functions can report this error:**
- MpPackMLCore

1.5.3.31 -1064008691: Write to configuration failed

**Description:**
Writing to a section of the configuration failed.

**Reaction:**
The function block indicates an active error on outputs "StatusID" and "Error". No other functions are available during this time.

**Cause/Solution:**
- Check the configuration.

**Constant**
mpPACKML_ERR_WRITE_MPPACKML_CFG
These function blocks / functions can report this error:

- MpPackMLCore

1.5.4 Alarms

1.5.4.1 mpPACKML_ALM_STATE_TRANSITION: Transition to this state not possible

Description:
It is not currently possible to transition to this state.

Reaction:
A mapp alarm is triggered if a MpAlarmX component is active.

Cause/Solution:
- Check the PackML state diagram.

Behavior
Persistent alarm

1.5.4.2 mpPACKML_ALM_MODE_TRANSITION: Mode transition not possible

Description:
The transition to this mode is not currently possible.

Reaction:
A mapp alarm is triggered if a MpAlarmX component is active.

Cause/Solution:
- Check the PackML state diagram and mode.

Behavior
Persistent alarm

1.5.4.3 mpPACKML_ALM_WRONG_CONFIG: Invalid configuration

Description:
The selected configuration is invalid.

Reaction:
A mapp alarm is triggered if a MpAlarmX component is active.

Cause/Solution:
- Check the configuration with PackML WebXs.

Behavior
Persistent alarm

1.6 PackML WebXs

MpPackML uses the WebXs to provide additional troubleshooting and configuration options, depending on the mapp component.
1.6.1 WebXs for MpPackMLCore

1.6.1.1 Diagnostics

The current mode and state can be viewed on the MpPackMLCore diagnostic page. It also displays static information for each mode:

- **Cumulative time**: Total runtime of the mode since the first start of the controller
- **Current time**: Runtime of the mode since the last time the controller was started

1.6.1.2 Configuration

The following parameters can be set on the configuration page of MpPackMLCore:

- **Start mode**: The mode enabled during booting can be set here.
- **Warning if a command was issued that is not allowed in the current state**
- **Warning if a mode change was requested that is not possible in the current mode/state**
- **Mode activation no longer depends on a rising edge of the command, but from the level of the command.**
1.6.2 WebXs for MpPackMLMode

1.6.2.1 Diagnostics

The current state diagram and statistical data about this mode can be viewed on the MpPackMLMode diagnostic page. The state diagram is shown as follows as long as the mode is not active:

As soon as the mode is active, the current state within the PackML state diagram is indicated:
The "Show statistics" button can be used to view precise statistical data about the individual states:

- **Cumulative time**: Total runtime of the state since the first start of the controller
- **Current time**: Runtime of the state since the last time the controller was started
1.6.2.2 Configuration

The PackML states tab on the configuration page of MpPackMLCore can be used to configure which states this mode can use. Clicking on a state changes it color and uses it. If a state is grayed out, then it is not used in this mode. The Idle state can always be used.
The *Transitions* tab can be used to define whether a mode transition is allowed. A distinction is made between transitioning to this mode and transitioning from this mode to another:

- **Transitioning to this mode**: If this arrow is enabled, it is possible to switch from another mode in the same state to this mode (as long as the other mode is configured correctly - arrow pointing up).

- **Transitioning to another mode**: If this arrow is enabled, it is possible to switch from this mode to another mode (as long as the other mode is configured correctly - arrow pointing down).