EUROMAP 91	OPC UA interfaces for plastics and rubber machinery - Winder

Release Candidate 1.0.0, 2024-10-01

EUROMAP 91 (Release Candidate 1.0.0) is identical with

OPC 40091 (Release Candidate 1.0.0) and VDMA 40091:2024-10

Contents

Figures

Tables

OPC Foundation / EUROMAP

_____

AGREEMENT OF USE

COPYRIGHT RESTRICTIONS

- This document is provided "as is" by the OPC Foundation and EUROMAP.
- Right of use for this specification is restricted to this specification and does not grant rights of use for referred documents.
- Right of use for this specification will be granted without cost.
- This document may be distributed through computer systems, printed or copied as long as the content remains unchanged and the document is not modified.
- OPC Foundation and EUROMAP do not guarantee usability for any purpose and shall not be made liable for any case using the content of this document.
- The user of the document agrees to indemnify OPC Foundation and EUROMAP and their officers, directors and agents harmless from all demands, claims, actions, losses, damages (including damages from personal injuries), costs and expenses (including attorneys' fees) which are in any way related to activities associated with its use of content from this specification.
- The document shall not be used in conjunction with company advertising, shall not be sold or licensed to any party.
- The intellectual property and copyright is solely owned by the OPC Foundation and EUROMAP.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OPC or EUROMAP specifications may require use of an invention covered by patent rights. OPC Foundation or EUROMAP shall not be responsible for identifying patents for which a license may be required by any OPC or EUROMAP specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OPC or EUROMAP specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

WARRANTY AND LIABILITY DISCLAIMERS

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OPC FOUDATION NOR EUROMAP MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OPC FOUNDATION NOR EUROMAP BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you.

RESTRICTED RIGHTS LEGEND

This Specification is provided with Restricted Rights. Use, duplication or disclosure by the U.S. government is subject to restrictions as set forth in (a) this Agreement pursuant to DFARs 227.7202-3(a); (b) subparagraph (c)(1)(i) of the Rights in Technical Data and Computer Software clause at DFARs 252.227-7013; or (c) the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 subdivision (c)(1) and (2), as applicable. Contractor / manufacturer are the OPC Foundation, 16101 N. 82nd Street, Suite 3B, Scottsdale, AZ, 85260-1830

COMPLIANCE

The combination of EUROMAP and OPC Foundation shall at all times be the sole entities that may authorize developers, suppliers and sellers of hardware and software to use certification marks, trademarks or other special designations to indicate compliance with these materials as specified within this document. Products developed using this specification may claim compliance or conformance with this specification if and only if the software satisfactorily meets the certification requirements set by EUROMAP or the OPC Foundation. Products that do not meet these requirements may claim only that the product was based on this specification and must not claim compliance or conformance with this specification.

TRADEMARKS

Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

GENERAL PROVISIONS

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of Germany.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, this specification.

Forewords

OPC UA is a machine to machine communication technology to transmit characteristics of products (e.g. manufacturer name, device type or components) and process data (e.g. temperatures, pressures or feed rates). To enable vendor unspecific interoperability the description of product characteristics and process data has to be standardized utilizing technical specifications, the OPC UA companion specifications.

This specification was created by a joint working group of the OPC Foundation and EUROMAP. It is adopted identically as VDMA Specification.

EUROMAP

EUROMAP is the European umbrella association of the plastics and rubber machinery industry which accounts for annual sales of around 13.5 billion euro and a 40 per cent share of worldwide production. Almost 75 per cent of its European output is shipped to worldwide destinations. With global exports of 10.0 billion euro, EUROMAP's around 1,000 machinery manufacturers are market leaders with nearly half of all machines sold being supplied by EUROMAP members.

EUROMAP provides technical recommendations for plastics and rubber machines. In addition to standards for machine descriptions, dimensions and energy measurement, interfaces between machines feature prominently. The provision of manufacturer independent interfaces ensures high levels of machine compatibility.

OPC Foundation

OPC is the interoperability standard for the secure and reliable exchange of data and information in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors. The OPC Foundation is responsible for the development and maintenance of this standard.

OPC UA is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework. This multi-layered approach accomplishes the original design specification goals of:

– Platform independence: from an embedded microcontroller to cloud-based infrastructure

– Secure: encryption, authentication, authorization and auditing

– Extensible: ability to add new features including transports without affecting existing applications

– Comprehensive information modelling capabilities: for defining any model from simple to complex

1   Scope

OPC 40091 describes the data model for winders (rewinder or unwinder) for data exchange during operation. While this specification was developed in the scope of plastic and rubber machines (e.g. winding of plastic film or tubes), this interface may also be suitable for other industries as well (e.g. printing machines). The target of OPC 40091 is to provide a unique interface for the communication of winders with control systems or other components of the production line from different manufacturers to ensure compatibility.

The following functionalities are covered:

– General information about the winding system (manufacturer, model, serial number…), current configuration and status.

– Process information like tensions, forces, speeds or web width at different measuring points

– Process information about the winding process like roll diameter, remaining time or quality

– Cyclic information about the winding process and roll quality

Following functions are not included:

– Safety related signals like emergency stop

2   Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments and errata) applies

OPC 10000-1, *OPC Unified Architecture - Part 1: Overview and Concepts*

> http://www.opcfoundation.org/UA/Part1/

OPC 10000-3, *OPC Unified Architecture - Part 3: Address Space Model*

> http://www.opcfoundation.org/UA/Part3/

OPC 10000-5, *OPC Unified Architecture - Part 5: Information Model*

> http://www.opcfoundation.org/UA/Part5/

OPC 10000-6, *OPC Unified Architecture - Part 6: Mappings*

> http://www.opcfoundation.org/UA/Part6/

OPC 10000-7, *OPC Unified Architecture - Part 7: Profiles*

> http://www.opcfoundation.org/UA/Part7/

OPC 10000-7, *OPC Unified Architecture - Part 16: State Machines*

> http://www.opcfoundation.org/UA/Part16/

OPC 10000-100, *OPC Unified Architecture - Part 100: Devices*

> http://www.opcfoundation.org/UA/Part100/

OPC 40001-1, *OPC UA for Machinery - Part 1: Basic Building Blocks*

> http://www.opcfoundation.org/UA/Machinery/

OPC 40001-2, *OPC UA for Machinery - Part 2: Process Values*

> http://www.opcfoundation.org/UA/Machinery/ProcessValues

OPC 40083: *OPC UA interfaces for plastics and rubber machinery – General Type definitions*

> http://www.opcfoundation.org/UA/PlasticsRubber/GeneralTypes

3   Terms, definitions and conventions

3.1   Overview

It is assumed that basic concepts of OPC UA information modelling are understood in this specification. This specification will use these concepts to describe the OPC 40091 Information Model. For the purposes of this document, the terms and definitions given in the documents referenced in Clause 2 apply.

Note that OPC UA terms and terms defined in this specification are *italicized* in the specification.

3.2   Conventions used in this document

The conventions described in OPC 40083 apply.

The same applies to the Container Concept described in OPC 40083, especially the use of the *Property NodeVersion:*

"Several objects can occur several times in the parent object (e.g. several moulds in one machine). For these, container objects are modelled. The benefit is that all instances are collected in one object so that changes can be easily recognized by using a Property NodeVersion which can be subscribed by clients. According to OPC UA, Part 3, the instances of the container objects shall also trigger a *GeneralModelChangeEvent.*"

3.3   Abbreviations

MES   Manufacturing Execution System

UUID   Universally unique identifier

4   General information to OPC UA interfaces for plastics and rubber machinery and OPC UA

For general information on OPC UA interfaces for plastics and rubber machinery and OPC UA see OPC 40083.

5   Use cases

The following functionalities are covered:

– General information about the winding system (manufacturer, model, serial number…), current configuration and status.

– Process information like tensions, forces, speeds or web width at different measuring points

– Process information about the winding process like roll diameter, remaining time or quality

– Cyclic information about the winding process and roll quality

– Mapping of the material path by linking the process values with the preceding ones.

Following functions are not included:

– Safety related signals like emergency stop

6   OPC 40091 Information Model Overview

The information model structure of OPC 40091 consists of the root *ObjectType Winder_InterfaceType*, which contains instances of all other *ObjectTypes* defined in this specification.

The *Winder_InteraceType* is defined in chapter 7. The top level objects are separated by use cases:

-    Identification and machine state
-    Information about the winding process in the *Winding Object*
-    Information about the wound rolls in the *WindingRolls Object*
-    Information about the drives *Drives Object*
-    Information about errors in the *ErrorHandling Object*

7   Winder_InterfaceType

7.1   Winder_InterfaceType definition

This OPC UA *ObjectType* is used for the root *Object* representing a rewinder or unwinder and is described in Table 1.

The instance(s) of *Winder_InterfaceType* shall be located under the *Machines Object* of the Server (see OPC UA for Machinery).
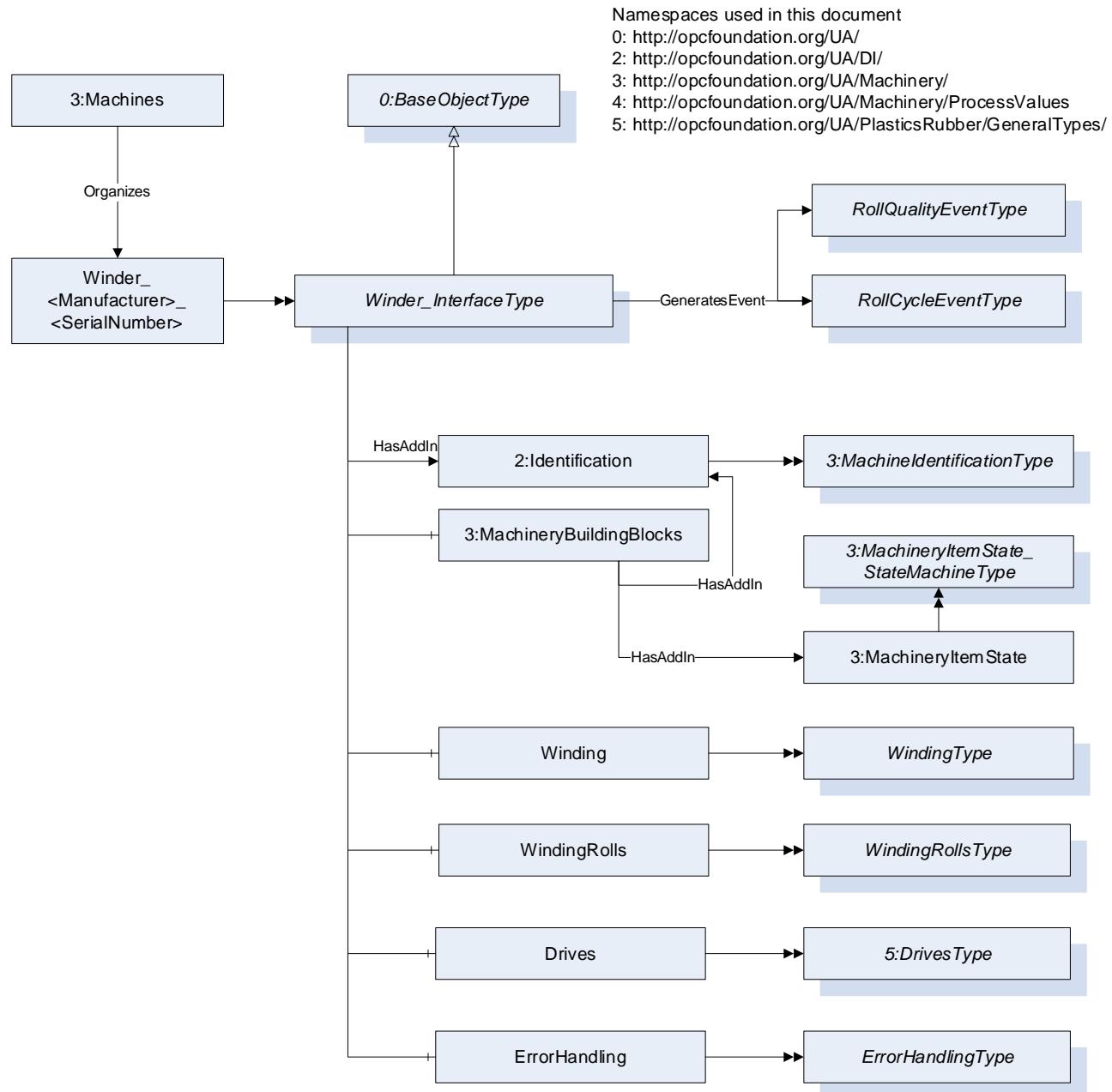
Figure 1 – Overview Winder_InterfaceType

Table 1 – Winder_InterfaceType Definition

Attribute	Value				
BrowseName	Winder_InterfaceType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:BaseObjectType defined in OPC 10000-5					
0:HasAddIn	Object	2:Identification		3:MachineIdentificationType	M
0:HasComponent	Object	3:MachineryBuildingBlocks		0:FolderType	M
0:HasComponent	Object	Winding		WindingType	M
0:HasComponent	Object	WindingRolls		WindingRollsType	M
0:HasComponent	Object	Drives		5:DrivesType	O
0:HasComponent	Object	ErrorHandling		ErrorHandlingType	M
0:GeneratesEvent	ObjectType	RollCycleEventType	Defined in 21		
0:GeneratesEvent	ObjectType	RollQualityEventType	Defined in 22		
Conformance Units					
OPC 40091 Basic					
OPC 40091 Drives					
OPC 40091 RollCycleEvent					
OPC 40091 RollQualityEvent					

7.2   Identification and MachineryBuildingBlocks

The *MachineIdentificationType* is defined in OPC UA for Machinery (OPC 40001-1) and provides basic information on a machine/device.

For the *InstanceDeclaration* the *ModellingRules* of the *Properties Model* and *DeviceClass* are overridden to mandatory and the optional *Property ControllerName* is added.

Table 2 – Winder_InterfaceType Additional Subcomponents

BrowsePath	References	NodeClass	BrowseName	DataType	TypeDefinition	Other
2:Identification	0:HasProperty	Variable	2:Model	0:LocalizedText	0:PropertyType	M, RO
2:Identification	0:HasProperty	Variable	2:DeviceClass	0:String	0:PropertyType	M, RO
2:Identification	0:HasProperty	Variable	ControllerName	0:String	0:PropertyType	O, RO
3:MachineryBuilding Blocks	0:HasAddIn	Object	2:Identification		3:MachineIdentificationType	M
3:MachineryBuilding Blocks	0:HasAddIn	Object	3:MachineryItem State		3:MachineryItemState_State MachineType	M

The *DeviceClass Property* in the *Identification Object* shall have the value "Rewinder" or "Unwinder".

The *ControllerName Property* represents the name of the machine controller (e.g. "CP22xx").

7.3   Winding

This *Object* provides information about the current winding process and contains process values like web tensions, nip forces, web widths, web speeds, web thickness at different measuring points. Provides also information about the winding mode and the state of the process steps unloading, loading, cutting, splicing and winding. The *WindingType* is defined in chapter 8.

7.4   WindingRolls

This *Object* provides information about the current wound-up rolls. It contains process values like roll diameter, winding length or remaining time until roll end. The *WindingRollsType* is defined in chapter 19.

7.5   Drives

This *Object* provides information about the drives used in the winder. The *DrivesType* is defined in OPC 40083.

7.6   ErrorHandling

This *Object* provides a minimal error handling for devices without alarm support. The *ErrorHandlingType* is defined in chapter 23.

8   WindingType

8.1   WindingType definition

The *WindingType* provides information about the current winding process and is defined in Table 3.
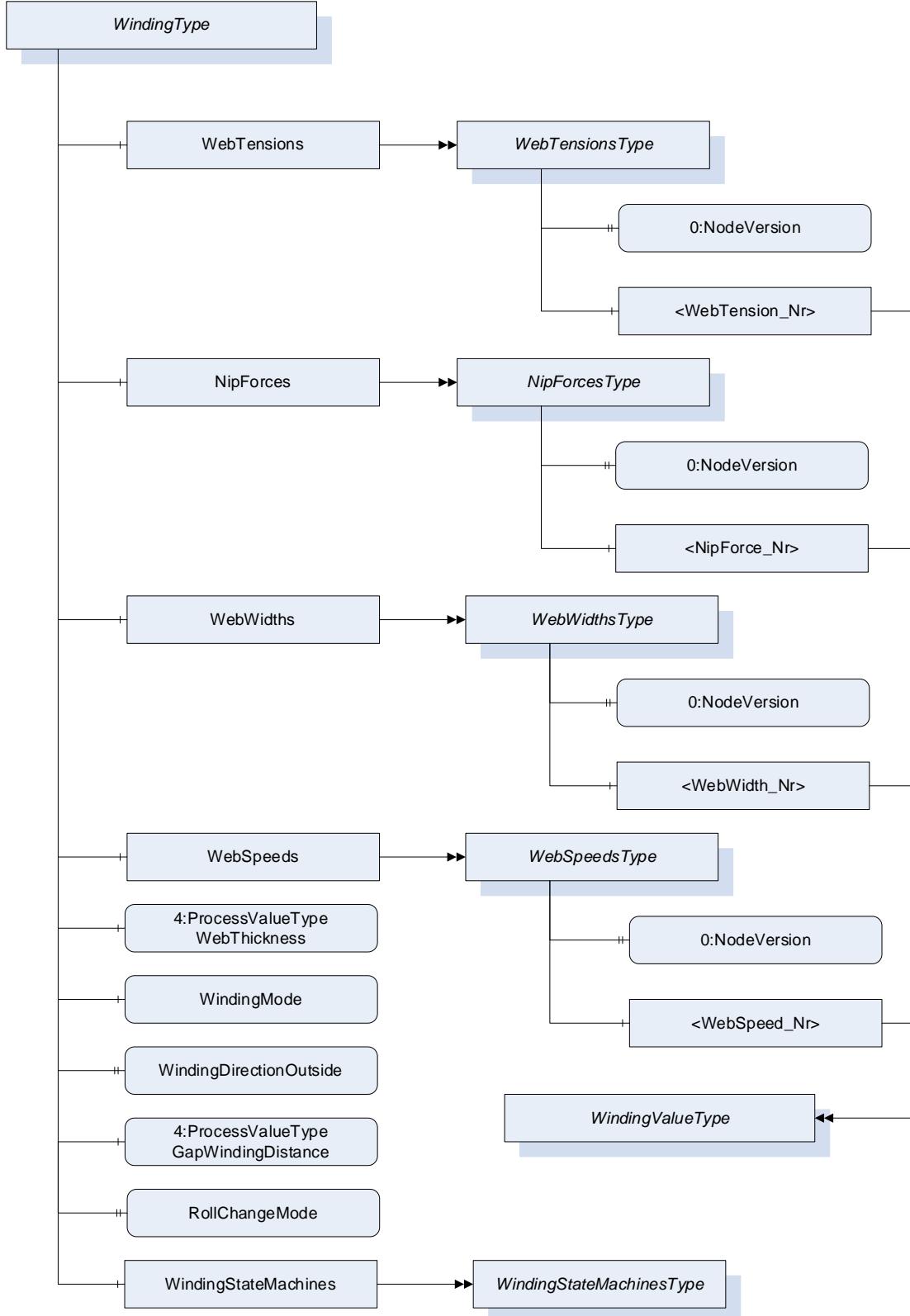
Figure 2 – WindingType Overview

Table 3 – WindingType Definition

Attribute	Value				
BrowseName	WindingType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasComponent	Object	WebTensions		WebTensionsType	M
0:HasComponent	Object	NipForces		NipForcesType	M
0:HasComponent	Object	WebWidths		WebWidthsType	M
0:HasComponent	Object	WebSpeeds		WebSpeedsType	M
0:HasComponent	Object	WebThickness		4:ProcessValueType	O
0:HasProperty	Variable	WindingMode	WindingModeEnumeration	0:PropertyType	M
0:HasProperty	Variable	WindingDirectionOutside	0:Boolean	0:PropertyType	M
0:HasComponent	Object	GapWindingDistance		4:ProcessValueType	O
0:HasProperty	Variable	RollChangeMode	RollChangeModeEnumeration	0:PropertyType	M
0:HasComponent	Object	WindingStateMachines		WindingStateMachinesType	M
Conformance Units					
OPC 40091 Basic					
OPC 40091 WebThickness					
OPC 40091 GapDistance					

8.2 WebTensions

Provides information about all monitored web tensions in the winding system. The *WebTensionsType* is defined in 9.

8.3 NipForces

Provides information about all monitored nip forces in the winding system. The *NipForcesType* is defined in 10.

8.4 WebWidths

Provides information about all monitored web widths in the winding system. The *WebWidhtsType* is defined in 11.

8.5 WebSpeeds

Provides information about all monitored web speeds in the winding system. The *WebSpeedsType* is defined in 12.

8.6 WebThickness

Provides the thickness of the web(film), typically measured with a thickness measuring device in the upstream production line. It can be also a value which comes with the roll as an information (applies mainly to the unwinder roll). The measured thickness gives an information about the thickness profile across the width and about the overall quality of the wound roll.

Additionally it can be used for the winding density. It can be also used for the calculation of the remaining film length on the roll at the unwinder. The web thickness is necessary, if the winding tension is set as a specific tension N/mm² (direction to cross area), instead of a typical N/m (direction to the web width).

Default unit: µm

8.7 WindingMode

Provides information about the current winding mode. The *WindingModeEnumeration* is defined in Table 4.

Table 4 – WindingModeEnumeration Definition

Name	Value	Description
Surface	0	Surface winding
Splat	1	Wind on splat
CenterSurface	2	Center Surface winding
Center	3	Center drive winding

8.8   WindingDirectionOutside

Provides information, which side of the material is outside while winding. Table 5 provides more information.

Table 5 – WindingDirectionOutside additional information

Value	Film	Tube
TRUE		
FALSE		
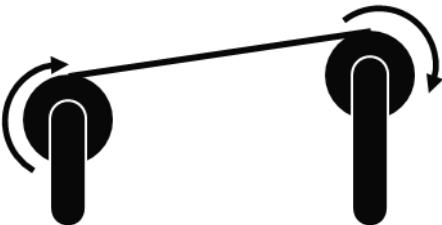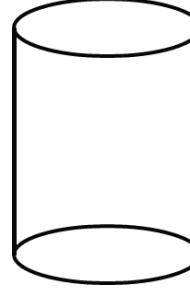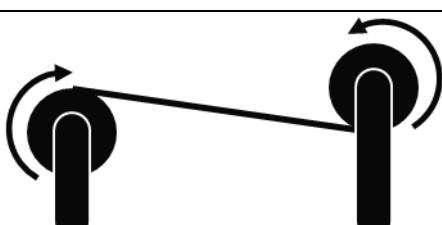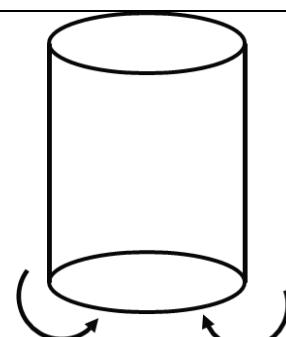
8.9   GapWindingDistance

Provides information about the winding gap and can be used to avoid a collision of the pressure carriage through setpoint specification.

Default unit: mm

8.10   RollChangeMode

Provides information about the roll change mode. The *RollChangeModeEnumeration* is defined in Table 6.

Table 6 – RollChangeModeEnumeration Definition

Name	Value	Description
RunLength	0	Roll is changed after a certain run length
Diameter	1	Roll is changed after a certain outside diameter of the wound roll
Hand	2	Roll is changed manually

A server can provide manufacturer specific values with *EnumValues* ≥ 100.

8.11   WindingStateMachines

Provides information about the current state of winding process steps like unloading, loading, cutting, splicing or winding. The *WindingStateMachinesType* is defined in 16.

9   WebTensionsType

9.1   WebTensionsType definition

This *Object* contains information about all monitored parameters related to web tensions in the winding system and is formally defined in Table 7.

Table 7 – WebTensionsType Definiton

Attribute	Value				
BrowseName	WebTensionsType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<WebTension_Nr>		WindingValueType	MP
Conformance Units					
OPC 40091 Basic					

The <WebTension_Nr> provides a placeholder for the *WebTensions*. When instances are created the *BrowseNames* shall be "<WebTension_Nr>" where "Nr" is a three-digit number with leading zeros, starting with "001". The *WindingValueType* is defined in 13.

Allowed values for *WindingValueKind* in the *WindingValueType* when instances of *WebTensions* are created are listed in the *WindingValueKindEnumeration* in Table 12 in the *EnumValue* range between 100 and 199.

10   NipForcesType

10.1   NipForcesType definition

This *Object* contains information about all monitored nip forces in the winding system and is formally defined in Table 8.

Table 8 – NipForcesType Definition

Attribute	Value				
BrowseName	NipForcesType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<NipForce_Nr>		WindingValueType	MP
Conformance Units					
OPC 40091 Basic					

The <NipForces_Nr> provides a placeholder for the *NipForces*. When instances are created the *BrowseNames* shall be "<NipForce_Nr>" where "Nr" is a three-digit number with leading zeros, starting with "001". The *WindingValueType* is defined in 13.

Allowed values for *WindingValueKind* in the *WindingValueType* when instances of *NipForces* are created are listed in the *WindingValueKindEnumeration* in Table 12 in the *EnumValue* range between 200 and 299.

11 WebWidthsType

11.1 WebWidthsType definition

This *Object* contains information about all monitored web widths in the winding system and is formally defined in Table 9.

Table 9 – WebWidthsType definition

Attribute	Value				
BrowseName	WebWidthsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<WebWidth_Nr>		WindingValueType	MP
Conformance Units					
OPC 40091 Basic					

The <WebWidth_Nr> provides a placeholder for the *WebWidths*. When instances are created the *BrowseNames* shall be "<WebWidth_Nr>" where "Nr" is a three-digit number with leading zeros, starting with "001". The *WindingValueType* is defined in 13.

Allowed values for *WindingValueKind* in the *WindingValueType* when instances of *WebWidths* are created are listed in the *WindingValueKindEnumeration* in Table 12 in the *EnumValue* range between 300 and 399.

12 WebSpeedsType

12.1 WebSpeedsType definition

This *Object* contains information about all monitored web speeds in the winding system and is formally defined in Table 10.

Table 10 – WebSpeedsType Definition

Attribute	Value				
BrowseName	WebSpeedsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<WebSpeed_Nr>		WindingValueType	MP
Conformance Units					
OPC 40091 Basic					

The <WebSpeed_Nr> provides a placeholder for the *WebSpeeds*. When instances are created the *BrowseNames* shall be "<WebSpeed_Nr>" where "Nr" is a three-digit number with leading zeros, starting with "001". The *WindingValueType* is defined in 13.

Allowed values for *WindingValueKind* in the *WindingValueType* when instances of *WebSpeeds* are created are listed in the *WindingValueKindEnumeration* in Table 12 in the *EnumValue* range between 400 and 499.

13 WindingValueType

13.1 WindingValueType definition

The *WindingValueType* represents one winding value (WebTension, NipForce, WebWidth, WebSpeed) and is formally defined in Table 11.
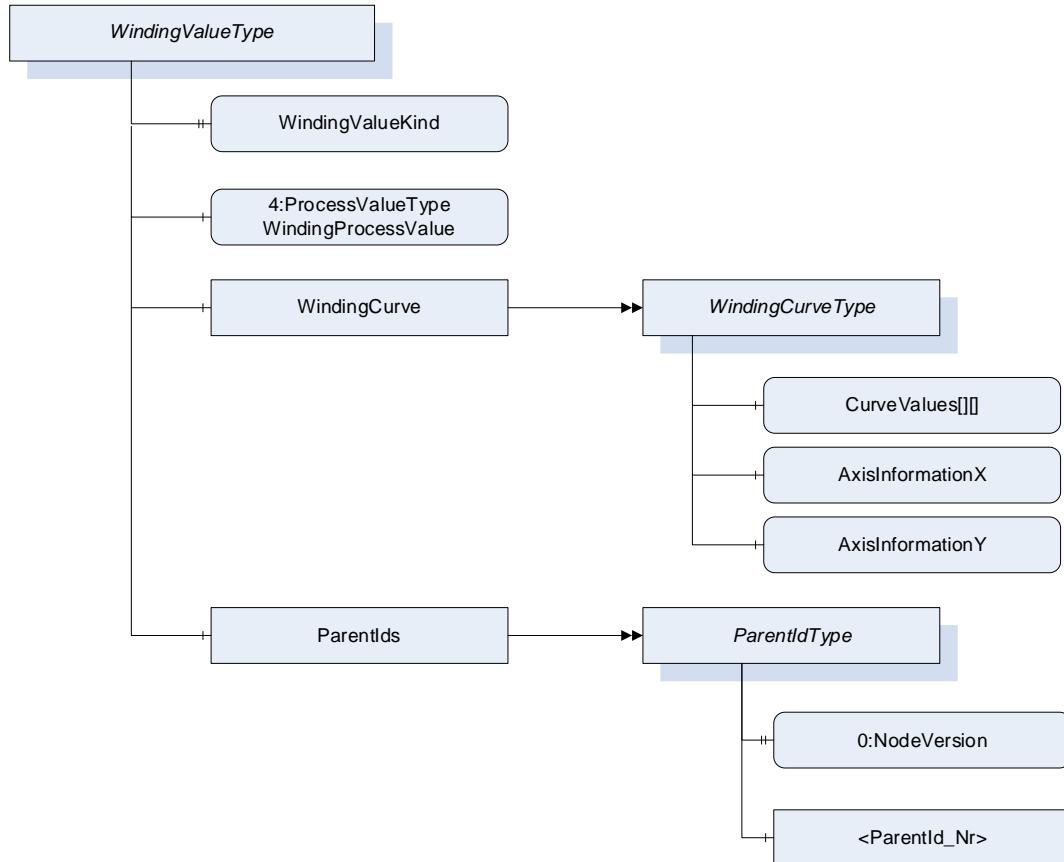
Figure 3 – WindingValueType Overview

Table 11 – WindingValueType Definition

Attribute	Value				
BrowseName	WindingValueType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	WindingValueKind	WindingValueKindEnumeration	0:PropertyType	M
0:HasComponent	Object	WindingProcessValue		4:ProcessValueType	M
0:HasComponent	Object	WindingCurve		WindingCurveType	O
0:HasComponent	Object	ParentIds		ParentIdType	M
Conformance Units					
OPC 40091 Basic					
OPC 40091 WindingCurve					

A name of the winding value shall be given by the *Node Attribute DisplayName*.

13.2 WindingValueKind

This variable provides information on what kind of process value it is. The *WindingValueKindEnumeration* is defined in Table 12.

Table 12 – WindingValueKindEnumeration Definition

Name	Value	Description
TensionDraw	100	Web tension at draw
ActualFilmTension	101	The current film tension in the active winding process
ActualPositionDancer	102	The current position of a dancer in front of the winder
TensionContactRollerDrive	103	Web tension at contact roller
LeadTrimDrawOffRoll	104	Lead of the trim draw-off roll
RewindTensionSpecificCenterDrive	105	Specific web tension at the rewinder
LeadSpreaderRoller	106	Lead of spreader roller
RewindTensionCenterDrive	107	Web tension at the rewinder
TensionCenterDrive	108	Web tension at center roller
ContactForce	200	Contact force roll on contact roller
PressureRiseWinder	201	Pressure reduction / increase on the winder
ForceContactLayOnRoller	202	Contact force lay-on roller on roll
WebWidthAfterLongCut	300	Web width after longitudinal cut
WebWidth	301	Web width before any longitudinal cut
WebSpeed	400	Speed of web
WebSpeedDraw	401	Speed of web at draw

A server can provide manufacturer specific values with *EnumValues* ≥ 500.

Further information regarding some of the Parameters of *WindingValueKindEnumeration* are given below.

TensionDraw: The so-called "draw" means a speed or torque difference between two or more driven rollers. This can be required, to either increase or typically decrease the elastic elongation (strain) in the web for a good winding without excessive embedded tension. Such a unit is typically located in the preparation section in front of the winder. The value of such a tension difference in N or in N/m (Newton per linear meter web width) can be either measured with tension measuring rollers or calculated by the motor torques difference of the draw rollers. It is also called "tension separation", because the tension of the production part is often higher than the required tension for rewinding.

TensionContactRollerDrive: At winders with driven contact rollers, it is possible to apply an additional tension onto the web by the contact roller drive. This is required for products (webs) where the needed rewind tension applied by the center drive (core / spindle) is not sufficient. Typically, the applied tension from the contact roller drive is calculated by the torque of the contact roller motor. The system of winding with driven contact roller is called "surface winding". In combination with a center drive of the core/spindle it is called "center/surface winding".

The taper ratio of this process value is given by the *WindingCurve*: This feature only applies to a driven contact roller or driven surface roller, which then enables to add an additional tension to the surface web layers. In order to avoid an overstretching of webs and to keep the pressure between the layers to an adequate range it is required to reduce the rewind tension with increasing roll diameter. This is called "taper tension" or "taper ratio" and follows the theory of the "constant torque winding". The start tension at core diameter corresponds to the "set point rewind tension rewinder surface drive" and is typically called 100 %. The reduction to the max. rewind diameter can be in a simple way as a straight way to an end point, e.g. 60 % of rewind surface tension at max. diameter. Another typical adjustment is done with several support points. With this system a set percentage at a specific diameter can be freely defined. With such an adjustment (often 12 or 14 support points) it possible to freely select a curve of rewind tension decay (e.g. with a hyperbolic shape).

Default unit: N

RewindTensionSpecificCenterDrive: The process value for the desired specific web tension (referred to the cross-web area) at the rewinder (RW). Typically, the tension is set in N as an absolute value or mostly in N/m, the specific web tension takes the thickness into consideration and gives the possibility to set it in N/mm². This means, that the set point tension takes the web width into

consideration as a specific value tension per meter web width and also the web thickness and realizes to set a specific web tension in N/mm² as a tension per cross area. The tension will be different according to the web thickness and web type and can be stored in a recipe.

LeadSpreaderRoller: The speed of the spreader roller, also called "banana roller", is set in percentage in relation to the last upstream driven roller. This is only possible, if the spreader roller (banana roller) has an individual drive. With the speed difference, the web can be hold under tension for a good flatness. If a value of 1,001 is adjusted, the spreader roller will have a speed, which is 0,1 % faster than the last driven roller upstream.

RewindTensionCenterDrive: The web tension at the rewinder (RW). Typically, the tension is set in N as an absolute value or mostly in N/m. This means, that the set point tension takes the web width into consideration as a specific value tension per meter web width. The tension will be different according to the web thickness and web type and can be stored in a recipe.

TensionCenterDrive: The web tension at the center drive of the winder is the applied tension by the driven rewind core. It can be determined either by a tension measuring system just before the wound roll or calculated by the applied torque of the motor minus the friction loss of the winder.

To avoid an overstretching of webs and to keep the pressure between the layers to an adequate range it is required to reduce the rewind tension with increasing roll diameter. This is called "taper tension" or "taper ratio" and follows the theory of the "constant torque winding". The start tension at core diameter corresponds to the "set point rewind tension rewinder center drive" and is typically called 100 %. The reduction to the max. rewind diameter can be in a simple way as a straight way to an end point, e.g. 60 % of rewind tension at max. diameter. Another typical adjustment is done with several support points. With this system a set percentage at a specific diameter can be freely defined. With such an adjustment (often 12 or 14 support points) it possible to freely select a curve of rewind tension decay (e.g. with a hyperbolic shape). The taper ratio curve can be given by the corresponding *WindingCurve*, while the minimum and maximum values (low and high limits) can be given in the *WindingProcessValue*.

ContactForce: The contact roller applies a force onto the wound roll in order to control / reduce the amount of air in between the film layers to ensure a compact wound roll with a straight edge. The actual contact force / contact pressure is either measured by force transducers or it is calculated by the pneumatic pressure of the cylinders (typically air cylinders). The taper ratio can be given by the *WindingCurve.*

Default unit: N

WebWidthAfterLongCut: Current web width (active/passive roller) from one side to the other side. The minimum and maximum values given by the range in the *WindingProcessValue* define the minimum and maximum product width, which determine the adjustment range of edge spreaders, edge loading, contact aids.

WebWidth: Width of film web before any longitudinal cut (MD = main direction). It is the width coming from the production line before it is transported to the preparation section for winding, where e.g. the edge trim is cut before winding. Example: If a web has a width of 8500 mm coming from the production and the two edge trims of 150 mm each are separated, the *WebWdith* is 8500 mm and the *WebWidthAfterlongCut* is 8200 mm. Although the winder itself is supposed to wind the 8200 mm only, it is required to consider the total width of 8500 mm, because typically the overall web is transported to the winder, when the line will be started. (this refers to the width of the core and the position of individual units, such as edge spreaders etc.)

Default unit: mm

WebSpeed: This is the speed of the web, either measured or calculated by motor revolution and roller diameter, coming from the upstream machine. It is important for an eventual adjustment of a "draw" between rollers (means a speed difference of two driven rollers), for a characteristic adjustment of winding tension and contact roller force and for a calculation of the remaining time of the running reel. The set point for the web speed is coming from the upstream line and

ensures, that the winder will follow when the line speed is changed. If there are driven rollers installed right before the winder for rewind preparation (so called "tension separation" or "draw"), the speed signal will be typically sent from line to the tension zone and then to the winder. The *Range Property* provides information about the physical limits of the value.

Default unit: m/min

WebSpeedDraw: If there is a tension separation located in front of the winder (driven rollers) the indicated speed of the driven roller(s) gives an information, whether the web elongation is increased or reduced between the production line and the tension separation and the tension separation and the winder. The difference is called "draw". The speed of these roller(s) is typically calculated by the motor revolutions and the circumference of the roller(s).

Default unit: m/min

13.3   WindingProcessValue

Provides information about the process value. The *ProcessValueType* is defined in OPC 40001-2 and contains the actual value, set value and two high and low limits with alarms.

13.4   WindingCurve

The *WindingCurve* provides a curve related to the process value (e.g. taper ratio as a function of the roll diameter). The *WindingCurveType* is defined in14.

13.5   ParentIds

The *PartentIds Object* provides information about the process values located upstream of the current value. This allows the material path to be mapped. The object is designed as a container object in order to model web divisions and mergers. The *ParentIdType* is defined in 15.

14   WindingCurveType

14.1   WindingCurveType definition

Provides a characteristic curve of a winding process value (e.g. taper ratio as a function of roll diameter). The *WindingCurveType* is formally defined in Table 13.

Table 13 – WindingCurveType

Attribute	Value				
BrowseName	WindingCurveType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	CurveValues	0:Double[][2]	0:PropertyType	M
0:HasProperty	Variable	AxisInformationX	0:AxisInformation	0:PropertyType	M
0:HasProperty	Variable	AxisInformationY	0:AxisInformation	0:PropertyType	M
ConformanceUnits					
OPC 40091 WindingCurve					

14.2   CurveValues

Provides the curve values. The *Variable* is a two-dimensional array with the size of the second dimension fixed to 2: X-Axis (abscissa) and Y-Axis (ordinate). The number of curve points, provided by the first dimension, is not limited.

14.3   AxisInformationX

Provides information like the unit about the X-Axis (e.g. diameter) of the curve.

14.4   AxisInformationY

Provides information like the unit about the Y-Axis (e.g. Force) of the curve.

15   ParentIdType

15.1   ParentIdType definition

The *ParentIdType* provides a container for all *ParentId_Nr* located upstream of the current value. This allows the material path to be mapped. The object is designed as a container object in order to model web divisions and mergers

Table 14 – ParentIdType definition

Attribute	Value				
BrowseName	ParentIdType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M
0:HasProperty	Variable	<ParentId_Nr>	0:String	0:PropertyType	MP
Conformance Units					
OPC 40091 Basic					

The ParentId references to the parents of the actual parameter. If the ParentId is empty, the actual parameter is the parent. The ParentIds are equal to either <WebTension_Nr>, <NipForce_Nr> , <WebWidth_Nr> or <WebSpeed_Nr> of the parent.

16 WindingStateMachinesType

16.1 WindingStateMachinesType definition

The *WindingStateMachinesType* provides information about the state of different winding components. A schematic is shown in Figure 4 and the formal definition in Table 15. It is a container object for all *<WindingState_Nr>*.
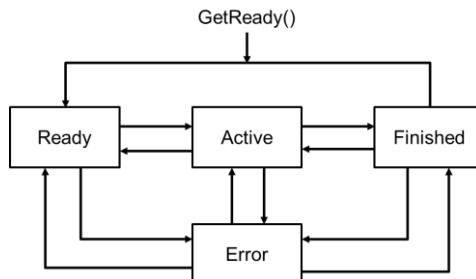
Figure 4 – WindingStateMachinesType schematic
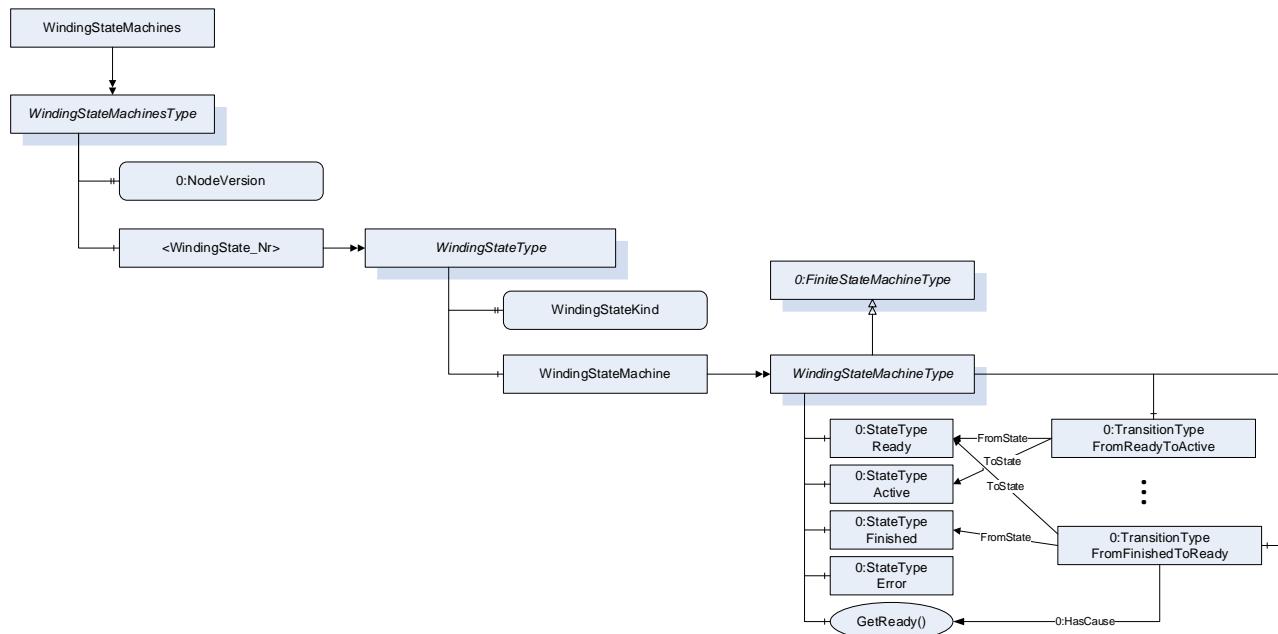
Figure 5 – WindingStateMachinesType Overview

Table 15 – WindingStateMachinesType Definition

Attribute	Value				
BrowseName	WindingStateMachinesType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<WindingState_Nr>		WindingStateType	MP
Conformance Units					
OPC 40091 Basic					

The *WindingStateType* is defined in 17.

17   WindingStateType

17.1   WindingStateType definition

The WindingStateType provides information about the state of a single process step and is defined in Table 16.

Table 16 – WindingStateType

Attribute	Value				
BrowseName	WindingStateType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	WindingStateKind	WindingStateKindEnumeration	0:PropertyType	M
0:HasComponent	Object	WindingStateMachine		WindingStateMachineType	M
Conformance Units					
OPC 40091 Basic					

A name shall be given by the *Node Attribute DisplayName*.

17.2   WindingStateKind

The *WindingStateKind Variable* provides information about which process step or component is involved. The *WindingStateKindEnumeration* is given in Table 17.

Table 17 – WindingStateKindEnumeration

Name	Value	Description
Unloading	0	Unloading process
Loading	1	Loading process
Cutting	2	Cutting process
Splicing	3	Splicing process
Winding	4	Winding process

A server can provide manufacturer specific values with *EnumValues* ≥ 100.

17.3   WindingStateMachine

Provides information about the state of the process step or component. The *WindingStateMachineType* is defined in 18.

18 WindingStateMachineType

18.1 WindingStateMachineType definition

The *WindingStateMachineType* provides information about the state of a process step or component. It is a *Subtype* of the *FiniteStateMachineType* and is formally defined in Table 18. It consists of the 4 states: *Ready*, *Active*, *Finished* and *Error*. There are *Transitions* between all states except from *Ready* to *Finished*. The *Transition* from *Finished* to *Ready* can be triggered by the *GetReady Method.*

Table 18 – WindingStateMachineType

Attribute	Value				
BrowseName	WindingStateMachineType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*FiniteStateMachineType* defined in OPC 10000-16					
0:HasProperty	Variable	0:DefaultInstanceBrowseName	0:QualifiedName	0:PropertyType	
0:HasComponent	Object	Ready		0:StateType	
0:HasComponent	Object	Active		0:StateType	
0:HasComponent	Object	Finished		0:StateType	
0:HasComponent	Object	Error		0:StateType	
0:HasComponent	Object	FromReadyToActive		0:TransitionType	
0:HasComponent	Object	FromReadyToError		0:TransitionType	
0:HasComponent	Object	FromActiveToFinished		0:TransitionType	
0:HasComponent	Object	FromActiveToError		0:TransitionType	
0:HasComponent	Object	FromFinishedToReady		0:TransitionType	
0:HasComponent	Object	FromFinishedToError		0:TransitionType	
0:HasComponent	Object	FromErrorToReady		0:TransitionType	
0:HasComponent	Object	FromErrorToActive		0:TransitionType	
0:HasComponent	Object	FromErrorToFinished		0:TransitionType	
0:HasComponent	Method	GetReady			O
Conformance Units					
OPC 40091 Basic					
OPC 40091 GetReady					

Table 19 – WindingStateMachineType Attribute Values for Child Nodes

SourceBrowsePath		Value	Description
0:DefaultInstanceBrowseName		WindingStateMachine	The default BrowseName for instances of the type
Ready		-	The process step or component is ready to perform any action
Active		-	The process step or component is currently performing an action
Finished		-	The process step or component has finished performing an action
Error		-	An error has occurred which led to a standstill
FromReadyToActive		-	Transition from state Ready to state Active
FromReadyToError		-	Transition from state Ready to state Error
FromActiveToFinished		-	Transition from state Active to state Finished
FromActiveToError		-	Transition from state Active to state Error
FromFinishedToReady		-	Transition from state Finished to state Ready
FromFinishedToError		-	Transition from state Finished to state Error
FromErrorToReady		-	Transition from state Error to state Ready
FromErrorToActive		-	Transition from state Error so state Active
FromErrorToFinished		-	Transition from state Error to state Finished
Ready 0:StateNumber		0	-
Active 0:StateNumber		1	
Finished 0:StateNumber		2	
Error 0:StateNumber		3	
FromReadyToActive 0:TransitionNumber		0	
FromReadyToError 0:TransitionNumber		1	
FromActiveToFinished 0:TransitionNumber		2	
FromActiveToError 0:TransitionNumber		3	
FromFinishedToReady 0:TransitionNumber		4	
FromFinishedToError 0:TransitionNumber		5	
FromErrorToReady 0:TransitionNumber		6	
FromErrorToActive 0:TransitionNumber		7	
FromErrorToFinished 0:TransitionNumber		8	

The components of *WindingStateMachineType* have additional *References* which are defined in Table 20.

Table 20 – WindingStateMachineType Additional References

SourceBrowsePath	ReferenceType	IsForward	TargetBrowsePath
FromReadyToActive	0:FromState	True	Ready
	0:ToState	True	Active
FromReadyToError	0:FromState	True	Ready
	0:ToState	True	Error
FromActiveToFinished	0:FromState	True	Active
	0:ToState	True	Finished
FromActiveToError	0:FromState	True	Active
	0:ToState	True	Error
FromFinishedToReady	0:FromState	True	Finished
	0:ToState	True	Ready
	0:HasCause	True	GetReady
FromFinishedToError	0:FromState	True	Finished
	0:ToState	True	Error
FromErrorToReady	0:FromState	True	Error
	0:ToState	True	Ready
FromErrorToActive	0:FromState	True	Error
	0:ToState	True	Active
FromErrorToFinished	0:FromState	True	Error
	0:ToState	True	Finished

18.2 GetReady

Method to request the transition from *Finished* State *to Ready* State. The *FromFinishedToReady* Transition has a *HasCause* Reference to this Method.

No arguments.

Signature

```
GetReady ();
```

Table 21 – GetReady Method AddressSpace Definition

Attribute	Value				
BrowseName	GetReady				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

19 WindingRollsType

19.1 WindingRollsType definition

The WindingRollsType provides information about the wound rolls and is formally defined in Table 22. It is a container for all <WindingRoll_Nr>. The number of <WindingRoll_Nr> instances is fixed by the configuration. Example: A turret rewinder with 2 winding shafts, where the web has been cut into 10 lanes, will have 20 instances of <WindingRoll_Nr> (10 rolls on each shaft).
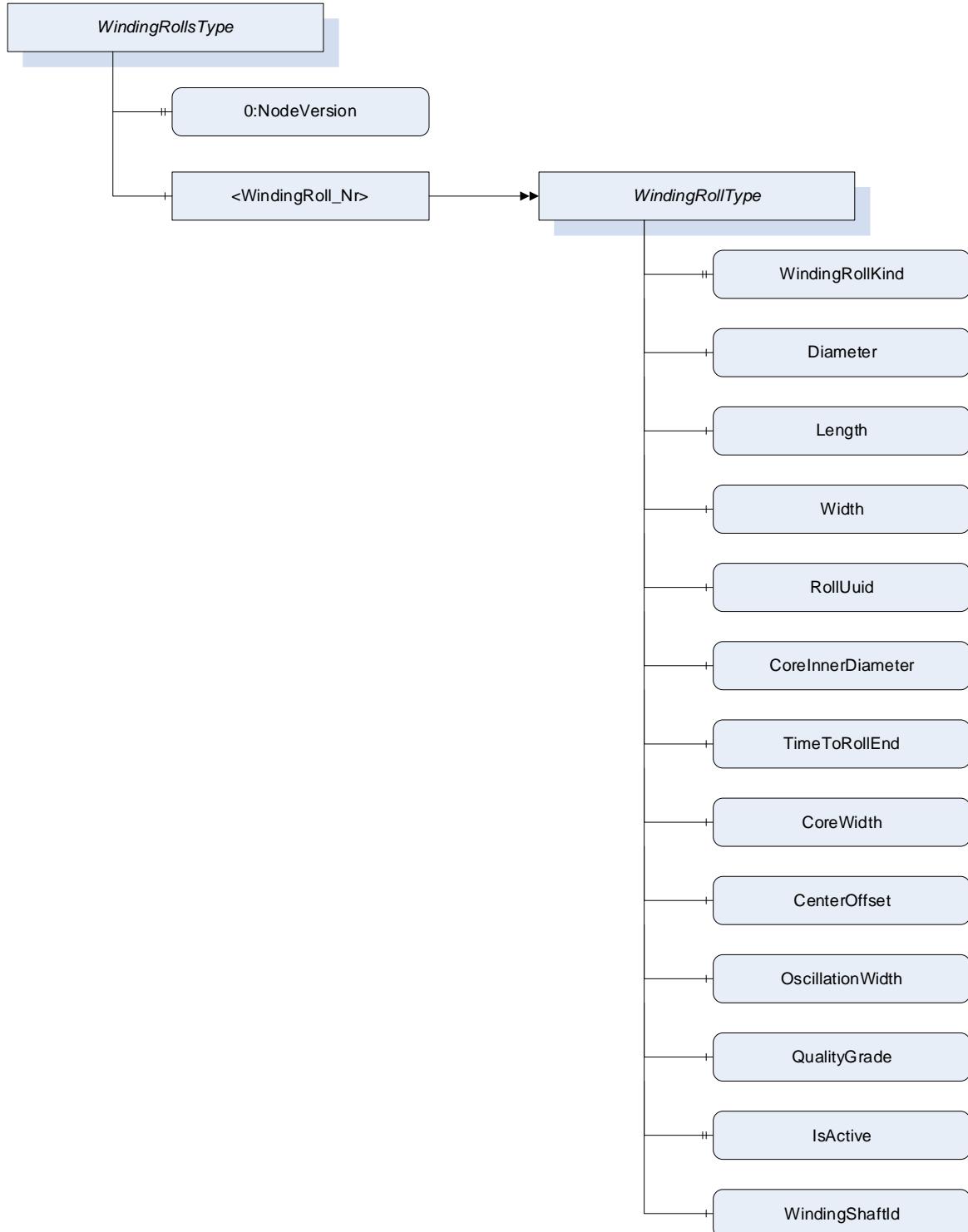

Figure 6 – WindingRollsType Overview

Table 22 – WindingRollsType Definiton

Attribute	Value				
BrowseName	WindingRollsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<WindingRoll_Nr>		WindingRollType	MP
Conformance Units					
OPC 40091 Basic					

The *WindingRollType* is defined in 20.

20 WindingRollType

20.1 WindingRollType definition

The *WindingRollType* provides information about one single wound roll and is defined in Table 23.

Table 23 – WindingRollType Definiton

Attribute	Value				
BrowseName	WindingRollType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	WindingRollKind	WindingRollKindEnumeration	0:PropertyType	M, RO
0:HasComponent	Object	Diameter		4:ProcessValueType	M
0:HasComponent	Object	Length		4:ProcessValueType	M
0:HasComponent	Object	Width		4:ProcessValueType	M
0:HasProperty	Variable	RollUuid	0:String	0:PropertyType	M
0:HasComponent	Variable	CoreInnerDiameter	0:Double	0:AnalogUnitType	M
0:HasComponent	Variable	TimeToRollEnd	0:Duration	0:BaseDataVariableType	M
0:HasComponent	Variable	CoreWidth	0:Double	0:AnalogUnitType	O
0:HasComponent	Object	CenterOffset		4:ProcessValueType	M
0:HasComponent	Object	OscillationWidth		4:ProcessValueType	M
0:HasComponent	Variable	QualityGrade	0:String	0:BaseDataVariableType	O
0:HasComponent	Variable	IsActive	0:Boolean	0:BaseDataVariableType	M
0:HasProperty	Variable	WindingShaftId	0:String	0:PropertyType	M
Conformance Units					
OPC 40091 Basic					
OPC 40091 CoreWidth					
OPC 40091 Quality					

20.2 WindingRollKind

Allowed values for *WindingRollKind* in the *WindingRollType* when instances of <WindingRoll_Nr> are created are listed in the *WindingRollKindEnumeration* in Table 24.

Table 24 – WindingRollKindEnumeration

Name	Value	Description
Unwinder	0	The roll is wound on an unwinder
Rewinder	1	The roll is wound on a rewinder

A server can provide manufacturer specific values with *EnumValues* ≥ 100.

20.3   Diameter

The actual diameter of the wound roll. The *ActualValue* represents the outside diameter of the roll, while the minimum value of the *Range Property* represents the core outer diameter.

Default unit: mm

20.4   Length

Run length of the wound web on the roll.

Default unit: mm

20.5   Width

Actual width of the roll / lane.

Default unit: mm

20.6   RollUuid

Universally unique identifier (UUID) of the roll.

20.7   CoreInnerDiameter

Inner Diameter of the roll core. The outer diameter of the core is given by the *Range Property* in the *Diameter Variable.*

Default unit: mm

20.8   TimeToRollEnd

Remaining time until roll end is reached.

20.9   CoreWidth

Width of the roll core.

Default unit: mm

20.10 CenterOffset

Offset between web center and core center.

Default unit: mm

20.11 OscillationWidth

Maximum oscillation width during the roll winding process.

Default unit: mm

20.12 QualityGrade

Information about the quality if determined during the roll winding process. The quality grade itself is not standardized and is given by the user as *String.*

20.13 IsActive

Indication, if the roll winding is currently active.

20.14 WindingShaftId

Unique (in the scope of the machine) ID of the winding shaft.

21 RollCycleEventType

21.1 RollCycleEventType definition

The *RollCycleEventType* provides cyclic information about wound rolls and is fired after the roll winding is finished.

```
RollCycleEventType
    ├── RollNr
    ├── RollStartTime
    ├── RollEndTime
    ├── Diameter
    ├── Length
    ├── Width
    ├── CoreInnerDiameter
    ├── RollUuid
    ├── CoreWidth
    ├── CenterOffset
    ├── OscillationWidth
    ├── WebThickness
    ├── WindingDirection
    └── WindingShaftId
```
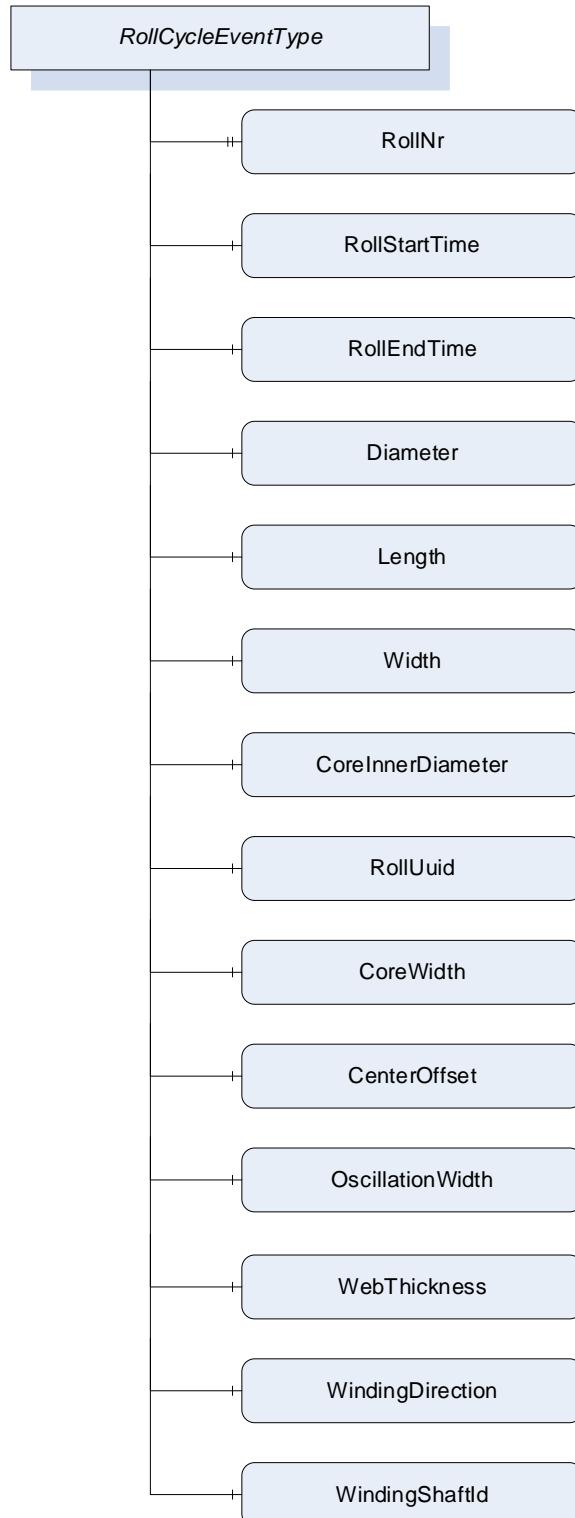
Figure 7 – RollCycleEvent Overview

Table 25 – RollCycleEventType Definition

Attribute	Value				
BrowseName	RollCycleEventType				
IsAbstract	True				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseEventType* defined in OPC UA Part 5					
0:HasProperty	Variable	RollNr	0:String	0:PropertyType	M
0:HasProperty	Variable	RollStartTime	0:UtcTime	0:PropertyType	M
0:HasProperty	Variable	RollEndTime	0:UtcTime	0:PropertyType	M
0:HasComponent	Variable	Diameter	0:Double	0:AnalogUnitRangeType	M
0:HasComponent	Variable	Length	0:Double	0:AnalogUnitType	M
0:HasComponent	Variable	Width	0:Double	0:AnalogUnitType	M
0:HasComponent	Variable	CoreInnerDiameter	0:Double	0:AnalogUnitType	M
0:HasProperty	Variable	RollUuid	0:String	0:PropertyType	M
0:HasComponent	Variable	CoreWidth	0:Double	0:AnalogUnitType	O
0:HasComponent	Variable	CenterOffset	0:Double	0:AnalogUnitType	M
0:HasComponent	Variable	OscillationWidth	0:Double	0:AnalogUnitType	M
0:HasProperty	Variable	QualityGrade	0:String	0:PropertyType	O
0:HasComponent	Variable	WebThickness	0:Double	0:AnalogUnitRangeType	M
0:HasProperty	Variable	WindingDirectionOutside	0:Boolean	0:PropertyType	M
0:HasProperty	Variable	WindingShaftId	0:String	0:PropertyType	M
Conformance Units					
OPC 40091 RollCycleEvent					
OPC 40091 CoreWidth					
OPC 40091 Quality					

21.2   RollNr

Same as <WindingRoll_Nr>

21.3   RollStartTime

Time when the winding process of the roll started.

21.4   RollEndTime

Time when the winding process of the roll ended.

21.5   Diameter

Diameter of the wound roll. The outside diameter is given by the value itself, while the minimum value of the *Range Property* represents the core outer diameter.

Default unit: mm

21.6   Length

Run length wound up on the roll.

Default unit: mm

21.7   Width

Actual width of the roll / lane.

Default unit: mm

21.8   CoreInnerDiameter

Inner Diameter of the roll core. The outer diameter of the core is given by the *Range Property* in the *Diameter Variable.*

Default unit: mm

21.9 RollUuid

Universally unique identifier (UUID) of the roll.

21.10 CoreWidth

Width of the roll core.

Default unit: mm

21.11 CenterOffset

Offset between web center and core center.

Default unit: mm

21.12 OscillationWidth

Maximum oscillation width during the roll winding process.

Default unit: mm

21.13 QualityGrade

Information about the quality if determined during the roll winding process. The quality grade itself is not standardized and is given by the user as *String.*

21.14 WebThickness

Provides the thickness of the web(film), typically measured with a thickness measuring device in the upstream production line. It can be also a value which comes with the roll as an information. The measured thickness gives an information about the thickness profile across the width and about the overall quality of the wound roll.

Default unit: µm

21.15 WindingDirectionOutside

Provides information, which side of the material is outside while winding. More information is provided in 8.8.

21.16 WindingShaftId

Unique (in the scope of the machine) ID of the winding shaft.

22   RollQualityEventType

22.1   RollQualityEventType definition

The *RollQualityEventType* is triggered as soon as a quality-relevant incident occurs during the winding process of a roll. This documents the time and run length together with information about the roll and the incident.

Table 26 – RollQualityEventType Overview
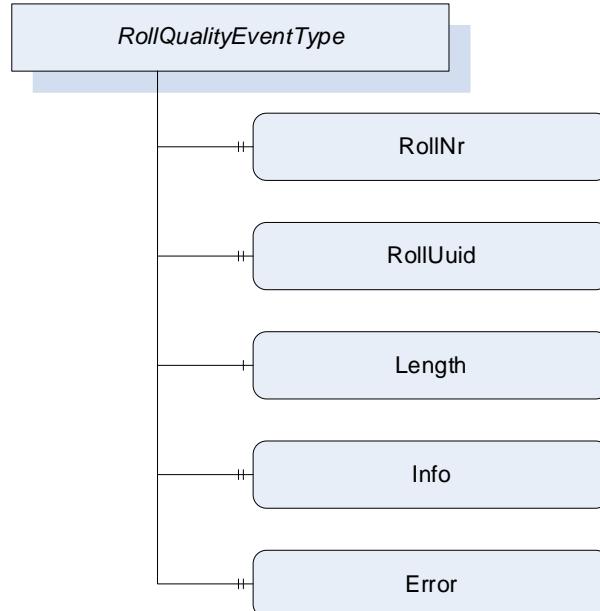
Table 27 – RollQualityEventType Definition

Attribute	Value				
BrowseName	RollQualityEventType				
IsAbstract	True				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseEventType* defined in OPC UA Part 5					
0:HasProperty	Variable	RollNr	0:String	0:PropertyType	M
0:HasProperty	Variable	RollUuid	0:String	0:PropertyType	M
0:HasComponent	Variable	Length	0:Double	0:AnalogUnitType	M
0:HasProperty	Variable	Info	0:String	0:PropertyType	M
0:HasProperty	Variable	Error	0:Boolean	0:PropertyType	M
Conformance Units					
OPC 40091 RollQualityEvent					

22.2   RollNr

Same as <WindingRoll_Nr>

22.3   RollUuid

Universally unique identifier (UUID) of the roll.

22.4   Length

Run length wound up on the roll at the time of the incident.

Default unit: mm

22.5 Info

Information about the incident. As this event can be used for any incident, it is not specified further here.

22.6 Error

Indicates, if the incident is an error.

23 ErrorHandlingType

This Object provides a minimal error handling for devices without alarm support. The *ErrorHandlingType* is defined in Table 28.

Table 28 – ErrorHandingType Definition

Attribute	Value				
BrowseName	ErrorHandlingType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC UA Part 5					
0:HasProperty	Variable	HighestActiveAlarmSeverity	0:UInt16	0:PropertyType	M, RO
0:HasComponent	Variable	ActiveErrors	5:ClassifiedActiveErrorData Type[]	0:BaseDataVariable Type	M, RO
0:HasComponent	Method	ResetAllErrors			O
0:HasComponent	Method	ResetErrorById			O
Conformance Units					
OPC 40091 ErrorHandling					
OPC 40091 ErrorHandlingMethods					

23.1 HighestActiveAlarmSeverity

Description: Indication of the severity of the highest active alarm (0 = no active alarm – 1000 = possible error). Together with *ActiveErrors*, it provides a minimal error handling for devices without alarm support. However, the variable shall be filled even if alarms are supported.

Example: 400

23.2 ActiveErrors

Description: List of the active errors of the device. It provides a minimal error handling for devices without alarm support. However, the variable shall be filled even if alarms are supported. The *ActiveErrorDataType* is defined in OPC 40083. If there is no active error, the array is empty.

23.3 ResetAllErrors

Description: Method to reset all errors of the device.

Signature

```
ResetAllErrors();
```

23.4 ResetErrorById

Description: Method to reset one error of the device.

Signature

```
ResetErrorById(
   [in]    String         Id);
```

Table 29 – ResetErrorById Method Arguments

Argument	Description
Id	Id of the error, listed in *ActiveErrors*, that shall be reset.

Table 30 – ResetErrorById Method AddressSpace Definition

Attribute	Value				
BrowseName	ResetErrorById				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

24 Profiles and Conformance Units

This chapter defines the corresponding profiles and conformance units for the OPC UA Information Model for OPC 40091. *Profiles* are named groupings of conformance units. Facets are profiles that will be combined with other *Profiles* to define the complete functionality of an OPC UA *Server* or *Client.*

24.1 Conformance Units

This chapter defines the corresponding *Conformance Unit* for OPC 40091.

Table 31 – Conformance Units for OPC 40091

Category	Title	Description
Server	OPC 40091 Basic	Support of *Winder_InterfaceType* and all mandatory child elements giving information on the winder and its status. There is at least one instance of the *Winder_InterfaceType* in the *Machines Object.*
Server	OPC 40091 Drives	Support of the *DrivesType* defined in OPC 40083.
Server	OPC 40091 RollCycleEvent	Support of the *RollCycleEventType* and all mandatory child elements to provide cyclic information about wound rolls.
Server	OPC 40091 RollQualityEvent	Support or the RollQualityEventType and all mandatory child elements to provide information about quality relevant incidents during winding.
Server	OPC 40091 CoreWidth	Support of the CoreWidth Variable in WindingRollType and RollCycleEventType to provide information about the core width.
Server	OPC 40091 Quality	Support of the Quality Variable in the WindingRollType and RollCycleEventType to provide information about the quality during winding.
Server	OPC 40091 WebThickness	Support or the *WebThickness* process value to provide information about the web thickness
Server	OPC 40091 GapDistance	Support of the *GapWindingDistance* process value to provide information about the gap winding distance.
Server	OPC 40091 WindingCurve	Support or the *WindingCurveType* to provide characteristic curves for the process values.
Server	OPC 40091 GetReady	Support of the *GetReady Method* to trigger the transition from the *Finished State* to the *Ready State* in the *WindingStateMachineType.*
Server	OPC 40091 ErrorHandling	Support of the *ErrorHandlingType* and all mandatory child elements providing a simplified error handling without alarms.
Server	OPC 40091 ErrorHandlingMethods	Support of the *Methods ResetAllErrors* and *ResetErrorById* inside the *ErrorHandlingType* to reset the errors.
Server	OPC 40091 AlarmSupport	Support of the A & C Alarm Server Facet (defined in OPC 10000-7)

24.2 Profiles

24.2.1 Profile list

The following tables specify the facets available for *Servers* that implement the OPC 40091 Information Model companion specification.

NOTE: The names of the supported profiles are available in the *Server Object* under *ServerCapabilities.ServerProfileArray*

Table 32 lists all Profiles defined in this document and defines their URIs.

Table 32 – Profile URIs for OPC 40091

Profile	URI
OPC 40082-5 Basic Server Profile	http://opcfoundation.org/UA-Profile/PlasticsRubber/Moulds/Server/Basic

24.2.2 Server Facets

24.2.2.1 Overview

The following sections specify the *Facets* available for *Servers* that implement the OPC 40091 companion specification. Each section defines and describes a *Facet* or *Profile.*

24.2.2.2   OPC 40091 Basic Server Profile

Table 33 – OPC 40091 Basic Server Profile

Group	Conformance Unit / Profile Title	Mandatory / Optional
Profile	0:Embedded Server 2017 (defined in OPC 10000-7)	M
Profile	0:ComplexType Server Facet (defined in OPC 10000-7)	M
Profile	0:Standard Event Subscription Server Facet (defined in OPC 10000-7)	M
Profile	0:Method Server Facet (defined in OPC 10000-7)	M
Profile	2:BaseDevice Server Facet (defined in OPC 10000-100)	M
Profile	3:Machinery Machine Identification Server Facet	M
Profile	3:Machinery State Server Facet	M
OPC 40091	OPC 40091 Basic	M
OPC 40091	OPC 40091 Drives	O
OPC 40091	OPC 40091 RollCycleEvent	O
OPC 40091	OPC 40091 RollQualityEvent	O
OPC 40091	OPC 40091 CoreWidth	O
OPC 40091	OPC 40091 Quality	O
OPC 40091	OPC 40091 WebThickness	O
OPC 40091	OPC 40091 GapDistance	O
OPC 40091	OPC 40091 WindingCurve	O
OPC 40091	OPC 40091 GetReady	O
OPC 40091	OPC 40091 ErrorHandling	O
OPC 40091	OPC 40091 ErrorHandlingMethods	O
OPC 40091	OPC 40091 AlarmSupport	O

25 Namespaces

25.1 Namespace Metadata

Table 34 defines the namespace metadata for this document. The *Object* is used to provide version information for the namespace and an indication about static *Nodes*. Static *Nodes* are identical for all *Attributes* in all *Servers*, including the *Value Attribute*. See OPC 10000-5 for more details.

The information is provided as *Object* of type *NamespaceMetadataType*. This *Object* is a component of the *Namespaces Object* that is part of the *Server Object*. The *NamespaceMetadataType ObjectType* and its *Properties* are defined in OPC 10000-5.

The version information is also provided as part of the ModelTableEntry in the UANodeSet XML file. The UANodeSet XML schema is defined in OPC 10000-6.

Table 34 – NamespaceMetadata Object for this Document

Attribute	Value		
BrowseName	http://opcfoundation.org/UA/PlasticsRubber/Winder/		
Property	**DataType**	**Value**	
NamespaceUri	String	http://opcfoundation.org/UA/PlasticsRubber/Winder/	
NamespaceVersion	String	RC 1.0.0	
NamespacePublicationDate	DateTime	2024-10-01	
IsNamespaceSubset	Boolean	False	
StaticNodeIdTypes	IdType []	0	
StaticNumericNodeIdRange	NumericRange []		
StaticStringNodeIdPattern	String		

Note: The *IsNamespaceSubset Property* is set to False as the UaNodeSet XML file contains the complete Namespace. *Servers* only exposing a subset of the Namespace need to change the value to True.

25.2 Handling of OPC UA Namespaces

Namespaces are used by OPC UA to create unique identifiers across different naming authorities. The *Attributes NodeId* and *BrowseName* are identifiers. A *Node* in the UA *AddressSpace* is unambiguously identified using a *NodeId*. Unlike *NodeIds*, the *BrowseName* cannot be used to unambiguously identify a *Node*. Different *Nodes* may have the same *BrowseName*. They are used to build a browse path between two *Nodes* or to define a standard *Property*.

Servers may often choose to use the same namespace for the *NodeId* and the *BrowseName*. However, if they want to provide a standard *Property*, its *BrowseName* shall have the namespace of the standards body although the namespace of the *NodeId* reflects something else, for example the *EngineeringUnits Property*. All *NodeIds* of *Nodes* not defined in this document shall not use the standard namespaces.

Table 35 provides a list of mandatory and optional namespaces used in an OPC 40091 OPC UA *Server*.

Table 35 – Namespaces used in a OPC 40091 Server

NamespaceURI	Description	Use
http://opcfoundation.org/UA/	Namespace for *NodeIds* and *BrowseNames* defined in the OPC UA specification. This namespace shall have namespace index 0.	Mandatory
Local Server URI	Namespace for nodes defined in the local server. This namespace shall have namespace index 1.	Mandatory
http://opcfoundation.org/UA/DI/	Namespace for *NodeIds* and *BrowseNames* defined in OPC 10000-100. The namespace index is *Server* specific.	Mandatory
http://opcfoundation.org/UA/Machinery/	Namespace for *NodeIds* and *BrowseNames* defined in OPC UA for Machinery – Part 1: Basic Building Blocks (OPC 40001-1). The namespace index is *Server* specific.	Mandatory
http://opcfoundation.org/UA/Machinery/ProcessValues/	Namespace for *NodeIds* and *BrowseNames* defined in OPC UA for Machinery – Part 2: Process Values (OPC 40001-2). The namespace index is *Server* specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/ GeneralTypes/	Namespace for *NodeIds* and *BrowseNames* defined in OPC 40083. The namespace index is server specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/Winder/	Namespace for *NodeIds* and *BrowseNames* defined in this document. The namespace index is *Server* specific.	Mandatory
Vendor specific types	A *Server* may provide vendor-specific types like types derived from *ObjectTypes* defined in this document in a vendor-specific namespace.	Optional
Vendor specific instances	A *Server* provides vendor-specific instances of the standard types or vendor-specific instances of vendor-specific types in a vendor-specific namespace. It is recommended to separate vendor specific types and vendor specific instances into two or more namespaces.	Mandatory

Table 36 provides a list of namespaces and their indices used for *BrowseNames* in this document. The default namespace of this document is not listed since all *BrowseNames* without prefix use this default namespace.

Table 36 – Namespaces used in this document

NamespaceURI	Namespace Index	Example
http://opcfoundation.org/UA/	0	0:EngineeringUnits
http://opcfoundation.org/UA/DI/	2	2:DeviceClass
http://opcfoundation.org/UA/Machinery/	3	3:MachineIdentificaionType
http://opcfoundation.org/UA/Machinery/ProcessValues/	4	4:ProcessValueType
http://opcfoundation.org/UA/PlasticsRubber/GeneralTypes/	5	5:MachineInformationType

Annex A
(normative)

OPC 40091 Namespace and mappings

A.1 NodeSet and supplementary files for OPC 40082-5 Information Model

The OPC 40082-5 *Information Model* is identified by the following URI:

http://opcfoundation.org/UA/PlasticsRubber/Winder/

Documentation for the NamespaceUri can be found here.

The *NodeSet* associated with this version of specification can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Winder/&v=1.0.0&i=1

The *NodeSet* associated with the latest version of the specification can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Winder/&i=1

Supplementary files for the OPC 40082-5 *Information Model* can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Winder/&v=1.0.0&i=2

The files associated with the latest version of the specification can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Winder/&i=2

_____