## Customer and Installation

|  |  |
| --- | --- |
| Installation | {#rope.facility ? rope.facility : `-`#} |
| Installation ID | {#rope.facilityId ? rope.facilityId : `-`#} |
| Contact | {#customer.contact ? customer.contact : `-`#} |
| Customer | {#customer.name ? customer.name : `-`#} |
| Address | {#customerAddress ? customerAddress : `-`#} |
| Rope | {#rope.name ? rope.name : `-`#} |

## Rope data

|  |  |
| --- | --- |
| Rope construction | {#rope.type ? rope.type : `-`#} |
| Actual diameter | {#measurementInfo.diameter != undefined ? measurementInfo.diameter + ` mm` : `-`#} |
| Nominal diameter | {#rope.nominalDiameter != undefined ? rope.nominalDiameter + ` mm` : `-`#} |
| Nominal laylength | {#rope.nominalLayLength ? rope.nominalLayLength + ` mm` : `-`#} |
| Manufacturer | {#rope.manufacturer != undefined ? rope.manufacturer : `-`#} |
| Nominal strength | {#rope.nominalStrength != undefined ? rope.nominalStrength + ` N/mm²` : `-`#} |
| Minimum breaking load | {#rope.minBreakingLoad != undefined ? rope.minBreakingLoad + ` k/N` : `-`#} |
| Metallic cross section | {#rope.metallicCrossSectionRope != undefined ? rope. MetallicCrossSectionRope + ` mm` : `-`#} |
| Installation date | {#rope.installationDate != undefined ? rope.installationDate : `-`#} |
| Operation hours | {#measurementInfo.operatingHours != undefined ? measurementInfo.operatingHours + ` hrs` : `-`#} |
| Measured rope length | {#measurementMetrics.measuredLength != undefined ? measurementMetrics.measuredLength + ` m` : `-`#} |

## Evaluation methods and limitations

Magneto-inductive rope testing is performed with the ROPESYS Expert Scan test system in accordance with EN 12927. The rope is magnetized axially. Stray fields are detected by sensors arranged around the rope and appear as a basic signal. Defects in the rope structure are superimposed on this stray field and stand out from the basic signal as characteristic signals.

In the case of spliced ropes, the evaluability of the test diagram is restricted in the area of the knots and tucked strand ends.

In the case of ropes that do not run endlessly, the testability is restricted in the areas of the rope end connections and the support areas of suspension ropes, depending on the installation.

The evaluation of this measurement was computer-aided. The inspector named in section 5 is responsible for evaluating the anomalies found and for checking the entire measurement report.

## Device data

|  |  |
| --- | --- |
| Device type - serial number | {#deviceType != undefined  ? deviceType : `-`#} |
| Device software (backend) | {#deviceSoftware != undefined ? deviceSoftware : `-`#} |
| Sensor type - serial number 1 | {#sensorType1 != undefined ? sensorType1 : `-`#} |
| Sensor type - serial number 2 | {#sensorType2 != undefined ? sensorType2 : `-`#} |
| Guiding blocks | {#measurementInfo.guideJaw != undefined ? measurementInfo.guideJaw + ` mm` : `-`#} |

## Report constraints

|  |  |
| --- | --- |
| Report date | {#measurementDate != undefined ? measurementDate : `-`#} |
| Examinant | {#measurementInfo.surveyor != undefined ? measurementInfo.surveyor : `-`#} |
| Participants | {#measurementInfo.participants != undefined ? measurementInfo.participants : `-`#} |
| Weather | {#measurementInfo.weather != undefined ? measurementInfo.weather : `-`#} |
| Temperature | {#measurementInfo.temperature != undefined ? measurementInfo.temperature + ` °C` : `-`#} |
| Starting point & Reference | {#let sp = measurementInfo.startingPoint != undefined; let spr = !measurementInfo.startingPointReference == false; (sp || spr) ? ((sp ? measurementInfo.startingPoint+` m` : ``) + (spr ? ` (`+measurementInfo.startingPointReference+`)` : ``)) : `-`#} |
| Measuring direction | {#measurementInfo.measurementDirection != undefined ? measurementInfo.measurementDirection : `-`#} |
| Mean speed | {#measurementMetrics.meanSpeedInMetersPerSecond? measurementMetrics.meanSpeedInMetersPerSecond+` m/s` : `-`#} |
| Report software (backend / frontend) | {#reportSoftwareVersions != undefined ? reportSoftwareVersions : `-`#} |

## Report results

|  |  |
| --- | --- |
| Rope condition | {#measurementInfo.ropeState != undefined ? measurementInfo.ropeState : `-`#} |
| Lubrication | {#measurementInfo.lubrication != undefined ? measurementInfo.lubrication : `-`#} |
| Strand contact | {#measurementInfo.strandContact != undefined ? measurementInfo.strandContact : `-`#} |

*{# IF measurementInfo.evaluationStandard != undefined #}*

## Evaluation standard report

|  |  |
| --- | --- |
| Evaluation standard | {#measurementInfo.evaluationStandard != undefined ? measurementInfo.evaluationStandard : `-`#} |
| Total number of defects | {#totalNumberOfBreaks != undefined ? totalNumberOfBreaks : `-`#} |
| Total number of defects in splice | {#totalDefectsInSplices != undefined ? totalDefectsInSplices : `-`#} |

*{# IF report.assessments.length === 4 #}*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference length | {#report.assessments[0] != undefined ? report.assessments[0].referenceLength : `-`#} x d | {#report.assessments[1] != undefined ? report.assessments[1].referenceLength : `-`#} x d | {#report.assessments[2] != undefined ? report.assessments[2].referenceLength : `-`#} x d | {#report.assessments[3] != undefined ? report.assessments[3].referenceLength : `-`#} x d |
| Max values | {#maxAllowedLoss[0]#} | {#maxAllowedLoss[1]#} | {#maxAllowedLoss[2]#} | {#maxAllowedLoss[3]#} |
| *{#FOR reportH IN reportHistory#}* |  |  |  |  |
| {#INS $reportH.date#} | {#INS $reportH.assessments[0].peakLoss#} % ({#INS $reportH.assessments[0].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[0].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* | {#INS $reportH.assessments[1].peakLoss#} % ({#INS $reportH.assessments[1].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[1].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* | {#INS $reportH.assessments[2].peakLoss#} % ({#INS $reportH.assessments[2].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[2].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* | {#INS $reportH.assessments[3].peakLoss#} % ({#INS $reportH.assessments[3].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[3].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* |
| *{#END-FOR reportH#}* |  |  |  |  |

*{# END-IF #}*

*{# IF report.assessments.length === 3 #}*

|  |  |  |  |
| --- | --- | --- | --- |
| Reference length | {#report.assessments[0] != undefined ? report.assessments[0].referenceLength : `-`#} x d | {#report.assessments[1] != undefined ? report.assessments[1].referenceLength : `-`#} x d | {#report.assessments[2] != undefined ? report.assessments[2].referenceLength : `-`#} x d |
| Max values | {#maxAllowedLoss[0]#} | {#maxAllowedLoss[1]#} | {#maxAllowedLoss[2]#} |
| *{#FOR reportH IN reportHistory#}* |  |  |  |
| {#INS $reportH.date#} | {#INS $reportH.assessments[0].peakLoss#} % ({#INS $reportH.assessments[0].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[0].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* | {#INS $reportH.assessments[1].peakLoss#} % ({#INS $reportH.assessments[1].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[1].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* | {#INS $reportH.assessments[2].peakLoss#} % ({#INS $reportH.assessments[2].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[2].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* |
| *{#END-FOR reportH#}* |  |  |  |

*{# END-IF #}*

*{# IF report.assessments.length === 2 #}*

|  |  |  |
| --- | --- | --- |
| Reference length | {#report.assessments[0] != undefined ? report.assessments[0].referenceLength : `-`#} x d | {#report.assessments[1] != undefined ? report.assessments[1].referenceLength : `-`#} x d |
| Max values | {#maxAllowedLoss[0]#} | {#maxAllowedLoss[1]#} |
| *{#FOR reportH IN reportHistory#}* |  |  |
| {#INS $reportH.date#} | {#INS $reportH.assessments[0].peakLoss#} % ({#INS $reportH.assessments[0].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[0].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* | {#INS $reportH.assessments[1].peakLoss#} % ({#INS $reportH.assessments[1].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[1].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* |
| *{#END-FOR reportH#}* |  |  |

*{# END-IF #}*

*{# IF report.assessments.length === 1 #}*

|  |  |
| --- | --- |
| Reference length | {#report.assessments[0] != undefined ? report.assessments[0].referenceLength : `-`#} x d |
| Max values | {#maxAllowedLoss[0]#} |
| *{#FOR reportH IN reportHistory#}* |  |
| {#INS $reportH.date#} | {#INS $reportH.assessments[0].peakLoss#} % ({#INS $reportH.assessments[0].maxLossCandidate#})  *{#IF $reportH.date === measurementDate #}*  *{#FOR reploss IN $reportH.assessments[0].representativeLosses#}*  {#INS $reploss.index#} m  *{#END-FOR reploss#}*  *{#END-IF#}* |
| *{#END-FOR reportH#}* |  |

*{# END-IF #}*

*{# END-IF #}*

*{#function map\_annotation\_type(s) {*

*function map(s) {*

*switch(s) {*

*case `wire break`:*

*return `broken wire`;*

*case `wire break cluster`:*

*return `broken wire cluster`;*

*case `insertion`:*

*return `tucked strand`;*

*case `knot`:*

*return `knot`;*

*case `splice begin`:*

*return `splice begin`;*

*case `splice end`:*

*return `splice end`;*

*case `marker`:*

*return `marker`;*

*case `corrosion`:*

*return `corrosion`;*

*case `other`:*

*return `other`;*

*case `automatic detection`:*

*return `automatic detection`;*

*case `measurement`:*

*return `measurement`;*

*case `lma snapshot`:*

*return `LMA Snapshot`;*

*default:*

*return s;*

*}*

*}*

*const lastIndex = s.lastIndexOf(` `);*

*let begin = s;*

*let end = s.slice(lastIndex + 1);*

*if (end.length > 0 && end[0] >= `0` && end[0] <= `9`) {*

*begin = s.slice(0, lastIndex);*

*}*

*else {*

*end = ``;*

*}*

*return map(begin) + (end ? (` ` + end) : ``);*

*}#}*

*{#function check\_annotation\_has\_metadata(a) {*

*switch(a){*

*case `splice begin`:*

*return true;*

*case `splice end`:*

*return true;*

*case `insertion`:*

*return true;*

*default:*

*return false;*

*}*

*}#}*

*{# IF splices[0] != undefined #}*

## Splices

**Splice 1:**

|  |  |  |
| --- | --- | --- |
| **Splice data** | **Position/Length [m]** | **Length [L/d(Rope)]** |
| Splice start | {#splices[0].start != undefined ? splices[0].start : `-`#} | - |
| Splice end | {#splices[0].end != undefined ? splices[0].end : `-`#} | - |
| Splice length | {#splices[0].lengthMeter != undefined ? splices[0].lengthMeter : `-`#} | {#splices[0].lengthLD != undefined ? splices[0].lengthLD : `-`#} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start** | **End** | **Length**  **[m]** | **Length**  **[L/d(Rope)]** | **Ø min\* [mm]** | **Ø max\***  **[mm]** |
| *{#FOR splice IN splices[0].data#}* |  |  |  |  |  |
| {#$splice.startName? *map\_annotation\_type*($splice.startName) : `-`  #} | {#INS $splice.endName? *map\_annotation\_type*($splice.endName) : `-`  #} | {#INS $splice.lengthMeter? $splice.lengthMeter + ` m`: `-`  #} | {#INS $splice.lengthLD? $splice.lengthLD + ` m`  : `-`  #} | {#INS $splice.minDiameter? $splice.minDiameter + ` mm` : `-`  #} | {#INS $splice.maxDiameter? $splice.maxDiameter + ` mm` : `-`  #} |
| *{#END-FOR splice#}* |  |  |  |  |  |

*\* The diameter relates to the splice element from the 'Start' column*

*{# END-IF #}*

*{# IF splices[1] != undefined #}*

**Splice 2:**

|  |  |  |
| --- | --- | --- |
| **Splice data** | **Position/Length [m]** | **Length [L/d(Rope)]** |
| Splice start | {#splices[1].start != undefined ? splices[1].start : `-`#} | - |
| Splice end | {#splices[1].end != undefined ? splices[1].end : `-`#} | - |
| Splice length | {#splices[1].lengthMeter != undefined ? splices[1].lengthMeter : `-`#} | {#splices[1].lengthLD != undefined ? splices[1].lengthLD : `-`#} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start** | **End** | **Length**  **[m]** | **Length**  **[L/d(Rope)]** | **Ø min\* [mm]** | **Ø max\***  **[mm]** |
| *{#FOR splice IN splices[1].data#}* |  |  |  |  |  |
| {#$splice.startName? *map\_annotation\_type*($splice.startName) : `-`  #} | {#INS $splice.endName? *map\_annotation\_type*($splice.endName) : `-`  #} | {#INS $splice.lengthMeter? $splice.lengthMeter + ` m`: `-`  #} | {#INS $splice.lengthLD? $splice.lengthLD + ` m`  : `-`  #} | {#INS $splice.minDiameter? $splice.minDiameter + ` mm` : `-`  #} | {#INS $splice.maxDiameter? $splice.maxDiameter + ` mm` : `-`  #} |
| *{#END-FOR splice#}* |  |  |  |  |  |

*\* The diameter relates to the splice element from the 'Start' column*

*{# END-IF #}*

*{# IF splices[2] != undefined #}*

**Splice 3:**

|  |  |  |
| --- | --- | --- |
| **Splice data** | **Position/Length [m]** | **Length [L/d(Rope)]** |
| Splice start | {#splices[2].start != undefined ? splices[2].start : `-`#} | - |
| Splice end | {#splices[2].end != undefined ? splices[2].end : `-`#} | - |
| Splice length | {#splices[2].lengthMeter != undefined ? splices[2].lengthMeter : `-`#} | {#splices[2].lengthLD != undefined ? splices[2].lengthLD : `-`#} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start** | **End** | **Length**  **[m]** | **Length**  **[L/d(Rope)]** | **Ø min\* [mm]** | **Ø max\***  **[mm]** |
| *{#FOR splice IN splices[2].data#}* |  |  |  |  |  |
| {#$splice.startName? *map\_annotation\_type*($splice.startName) : `-`  #} | {#INS $splice.endName? *map\_annotation\_type*($splice.endName) : `-`  #} | {#INS $splice.lengthMeter? $splice.lengthMeter + ` m`: `-`  #} | {#INS $splice.lengthLD? $splice.lengthLD + ` m`  : `-`  #} | {#INS $splice.minDiameter? $splice.minDiameter + ` mm` : `-`  #} | {#INS $splice.maxDiameter? $splice.maxDiameter + ` mm` : `-`  #} |
| *{#END-FOR splice#}* |  |  |  |  |  |

*\* The diameter relates to the splice element from the 'Start' column*

*{# END-IF #}*

*{# IF splices[3] != undefined #}*

**Splice 4:**

|  |  |  |
| --- | --- | --- |
| **Splice data** | **Position/Length [m]** | **Length [L/d(Rope)]** |
| Splice start | {#splices[3].start != undefined ? splices[3].start : `-`#} | - |
| Splice end | {#splices[3].end != undefined ? splices[3].end : `-`#} | - |
| Splice length | {#splices[3].lengthMeter != undefined ? splices[3].lengthMeter : `-`#} | {#splices[3].lengthLD != undefined ? splices[3].lengthLD : `-`#} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start** | **End** | **Length**  **[m]** | **Length**  **[L/d(Rope)]** | **Ø min\* [mm]** | **Ø max\***  **[mm]** |
| *{#FOR splice IN splices[3].data#}* |  |  |  |  |  |
| {#$splice.startName? *map\_annotation\_type*($splice.startName) : `-`  #} | {#INS $splice.endName? *map\_annotation\_type*($splice.endName) : `-`  #} | {#INS $splice.lengthMeter? $splice.lengthMeter + ` m`: `-`  #} | {#INS $splice.lengthLD? $splice.lengthLD + ` m`  : `-`  #} | {#INS $splice.minDiameter? $splice.minDiameter + ` mm` : `-`  #} | {#INS $splice.maxDiameter? $splice.maxDiameter + ` mm` : `-`  #} |
| *{#END-FOR splice#}* |  |  |  |  |  |

*\* The diameter relates to the splice element from the 'Start' column*

*{# END-IF #}*

*{# IF splices[4] != undefined #}*

**Splice 5:**

|  |  |  |
| --- | --- | --- |
| **Splice data** | **Position/Length [m]** | **Length [L/d(Rope)]** |
| Splice start | {#splices[4].start != undefined ? splices[4].start : `-`#} | - |
| Splice end | {#splices[4].end != undefined ? splices[4].end : `-`#} | - |
| Splice length | {#splices[4].lengthMeter != undefined ? splices[4].lengthMeter : `-`#} | {#splices[4].lengthLD != undefined ? splices[4].lengthLD : `-`#} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start** | **End** | **Length**  **[m]** | **Length**  **[L/d(Rope)]** | **Ø min\* [mm]** | **Ø max\***  **[mm]** |
| *{#FOR splice IN splices[4].data#}* |  |  |  |  |  |
| {#$splice.startName? *map\_annotation\_type*($splice.startName) : `-`  #} | {#INS $splice.endName? *map\_annotation\_type*($splice.endName) : `-`  #} | {#INS $splice.lengthMeter? $splice.lengthMeter + ` m`: `-`  #} | {#INS $splice.lengthLD? $splice.lengthLD + ` m`  : `-`  #} | {#INS $splice.minDiameter? $splice.minDiameter + ` mm` : `-`  #} | {#INS $splice.maxDiameter? $splice.maxDiameter + ` mm` : `-`  #} |
| *{#END-FOR splice#}* |  |  |  |  |  |

*\* The diameter relates to the splice element from the 'Start' column*

*{# END-IF #}*

**Attachment: annotation list**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Position** | **Type** | **Description** |
| *{#FOR annotation IN annotations#}* |  |  |  |
| {#INS $annotation.displayIndex#} | {#INS $annotation.positionMeter#} | {#INS *map\_annotation\_type*($annotation.type)#}  {#IF *check\_annotation\_has\_metadata($annotation.type)#}* Ø Min: {#$annotation.min? $annotation.min + ` mm` :`-`#}  Ø Max: {#$annotation.max? $annotation.max + ` mm` :`-`#}  *{#END-IF#}*  {#IF *$annotation.type === `measurement`#}* Ø Min: {#$annotation.min? $annotation.min + ` mm` :`-`#}  Ø Max: {#$annotation.max? $annotation.max + ` mm` :`-`#}  Ø Lay length: {#$annotation.layLength? $annotation. layLength + ` mm` :`-`#}  *{#END-IF#}*  {#IF *$annotation.type === `lma snapshot`#}* LMA: {#$annotation.lmaOverride?$annotation.lmaOverride+` %`:$annotation.lmaActual+` %`#}  *{#END-IF#}*  *{#IF $annotation.wireBreakCount #}*  broken wire count: {#$annotation.wireBreakCount#}  *{#END-IF#}* | {#INS $annotation.description#} |
| *{#END-FOR annotation#}* |  |  |  |