ibaQDR

Quality data recording system

Documentation and project planning guidelines

Issue 2.2

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG

Gebhardtstrasse 10-20

90762 Fuerth

Germany

Contacts

Main office	+49 911 97282-0
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

Version	Date	Revision	Author	Version SW
2.2	04-2026	New options for reversing measuring location; slitting line	dm/rm	v8.13.0

Contents

1 About this documentation

This documentation describes the configuration and engineering of

ibaQDR.

1.1 Target group

This documentation is aimed at qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if they are capable of assessing safety and recognizing possible consequences and risks on the basis of their specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation is aimed at persons who are involved in the evaluation of process and quality data and are required to set up, configure and maintain the *ibaQDR* system. The following prior knowledge is required and/or useful:

- Basic knowledge of *ibaPDA*
- Basic knowledge of *ibaAnalyzer*
- Knowledge of the system / process for which the product data is to be recorded

1.2 Notations

In this manual, the following notations are used:

Action	Notation
Menu command	Menu *Logic diagram*
Calling the menu command	*Step 1 – Step 2 – Step 3 – Step x* Example: Select the menu *Logic diagram – Add – New function block*.
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	`Filename`, `Path` Example: `Test.docx`

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!

The non-observance of this safety information may result in an imminent risk of death or severe injury!

Observe the specified measures.

Warning!

The non-observance of this safety information may result in a potential risk of death or severe injury!

Observe the specified measures.

Caution!

The non-observance of this safety information may result in a potential risk of injury or material damage!

Observe the specified measures.

Note

A note specifies special requirements or actions to be observed.

Tip

Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation

Reference to additional documentation or further reading.

2 Product information

2.1 Introduction

For every manufacturer of high-quality goods, quality data is an indispensable part of the modern production process.

The *ibaQDR* (Quality Data Recording) system is an extension of the *ibaPDA* software and provides transparent, high-resolution quality data acquisition with convenient operation, a wide range of interfaces, and efficient quality data management.

For meaningful quality data analysis, it is essential that the measured values acquired can be correctly assigned to the product. For some products, length-based data mapping within the product is also required. This means that the measured values acquired sequentially in the different sections of the line during production have to be mapped to the length of the finished product at the end.

ibaQDR has been specially developed for the requirements of strip processing lines in the metal production industry, but can also be used in discontinuous systems such as push-pull pickling, hot and cold rolling mills, in coupled systems (“pickling tandem”), as well as in inspection lines and continuous casting plants. Moreover, *ibaQDR* can be applied to all kinds of production lines producing long or web-shaped products, such as rubber calanders, paper machines or extrusion lines, as well as plants for plate glass and wood panels.

The software to be installed is the *ibaPDA* software with an additional license for *ibaQDR* data storage, which is enabled on the dongle.

As with *ibaPDA*, the stored data is displayed, evaluated, and processed using the *ibaAnalyzer* software.

This documentation describes only the *ibaQDR*-specific functions. For all other general information about using the software, refer to the manual for the *ibaPDA* product.

Since *ibaQDR* works closely with the segment mapping in the line automation system, detailed knowledge of the functioning of the automation system is essential for configuration. Therefore, we recommend that new users only implement *ibaQDR* with the assistance of qualified equipment suppliers or in conjunction with support services from iba AG (consultancy, training, support).

2.2 License information

For *ibaQDR* you can choose from several basic products with different numbers of signals and measuring locations, which will cover most of the applications. With the *ibaQDR* basic licenses you will get an *ibaPDA* system with the specified number of signals, two *ibaPDA* data stores and the *ibaQDR* data store for a number of measuring locations.

The *ibaPDA* products are available for the required interface licenses.

Basic licenses

Order no.	Product name	Description
35.702560	ibaQDR-256-6	Quality data acquisition and storage for max. 256 signals and up to 6 measuring locations + 2 ibaPDA stores
35.710240	ibaQDR-1024-32	Quality data acquisition and storage for max. 1024 signals and up to 32 measuring locations + 2 ibaPDA stores
35.720480	ibaQDR-2048-48	Quality data acquisition and storage for max. 2048 signals and up to 48 measuring locations + 2 ibaPDA stores
35.799990	ibaQDR-unlimited-64	Quality data acquisition and storage for an unlimited number of signals and up to 64 measuring locations + 2 ibaPDA stores
35.799992	ibaQDR-unlimited-96	Quality data acquisition and storage for an unlimited number of signals and up to 96 measuring locations + 2 ibaPDA stores
35.799993	ibaQDR-unlimited-128	Quality data acquisition and storage for an unlimited number of signals and up to 128 measuring locations + 2 ibaPDA stores
35.799994	ibaQDR-unlimited-160	Quality data acquisition and storage for an unlimited number of signals and up to 160 measuring locations + 2 ibaPDA stores
35.799995	ibaQDR-unlimited-192	Quality data acquisition and storage for an unlimited number of signals and up to 192 measuring locations + 2 ibaPDA stores

Note

At reversing measuring locations, the maximum number of passes is taken into account in the measuring location count, as each pass is treated as a separate measuring location.

Extended licenses

Order no.	Product name	Description
35.700001	ibaQDR-Measuring-Location	2 additional measuring locations
30.770003	Upgrade PDA-256 to PDA-512	Increase number of signals
30.770004	Upgrade PDA-512 to PDA-1024	Increase number of signals
30.770005	Upgrade PDA-1024 to PDA-2048	Increase number of signals
30.770006	Upgrade PDA-2048 to PDA-4096	Increase number of signals
30.770007	Upgrade PDA-4096 to PDA-8192	Increase number of signals
30.770008	Upgrade PDA-8192 to PDA-unlimited	Increase number of signals

Note

When using multiple, parallel QDR data stores, distribute the measuring locations among the various data stores. If there are not enough measuring locations available for all data stores, you can purchase additional measuring location licenses.

Training and application support for new users

For users who want to plan and install an *ibaQDR* system for the first time, iba AG strongly recommend purchasing the “ibaQDR Requirements” service package.

Unlike with *ibaPDA* the success of *ibaQDR* project planning depends to a large extent on understanding its special functions and the complex interrelationships between *ibaQDR*, the process, and the line control. Training and support are provided by experienced experts.

Order no.	Product name	Description
60.700201	ibaQDR-Requirements	1. ibaQDR standard training (1 day) 2. Project planning support (1 day remote support) 3. Support for commissioning (3 days remote support, including test using playback)

For equipment suppliers who want to use *ibaQDR* for the first time, special sales conditions are applicable. We offer a comprehensive qualification package, culminating in certification. You can find information about this in the Appendix, chapter ↗ *Sales conditions, page 95*.

2.3 System requirements

The same system requirements apply as for *ibaPDA*. You can find these in Part 1 of the *ipaPDA* manual, chapter 3.1 or in the iba Help center under System requirements.

In order to be able to use all functions described herein you'll need the following:

- ibaPDA v8.13.0 or higher
- ibaAnalyzer v8.2.2 or higher for supporting multiple QDR data stores for analysis
- ibaDatCoordinator v4.0 or higher for supporting multiple QDR data stores in automated post-processing

Operating system

Information about the supported operating system versions and other compatibility conditions for other programs and hardware components can be found in the file *versions_pda.htm* on the "iba Software & Manuals" data carrier or in the download Zip file. After installation, you can also open the file by selecting *Help - Version History* in the menu.

Open this file in your Internet browser and click on the *Version compatibility* header. As well as information about the current program version you will also find a history.

ibaPDA

Version history	Version compatibility	
Current version		
Version	Operating system	Requirements
	Windows 8.1 (x86/x64)	.NET Framework 4.8
	Windows 10 (x86/x64)	Version 1607 or later .NET Framework 4.8 (integrated in OS in version 1903 and later)
	Windows 11 (x64)	.NET Framework 4.8 (integrated in OS)
	Windows Server 2012 (x64)	.NET Framework 4.8
	Windows Server 2012 R2 (x64)	.NET Framework 4.8
	Windows Server 2016 (x64)	.NET Framework 4.8
	Windows Server 2019 (x64)	.NET Framework 4.8
	Windows Server 2022 (x64)	.NET Framework 4.8 (integrated in OS)
	Supported hardware	Remarks
	License	WIBU CmStick or CmActLicense MARX dongle
	ibaFOB-D, ibaFOB-Dexp, ibaFOB-io-ExpressCard, ibaFOB-io-USB, ibaFOB-SD, ibaFOB-SDe, ibaFOB-TDC, ibaFOB-TDCe, ibaCom-L2B, VMIC-5565, VMIC-5576, DGM200P, CP1616, CP1626, PC Link SST 5136-RE2-PCI	Supported on both x86 and x64 OS

Hardware

Depending on the design of the system (number of measuring locations, signals, interfaces and scope of calculations), a powerful computer with adaptable hardware should be used. Devices from the *ibaDAQ* family are not suitable for *ibaQDR*.

Since an *ibaQDR* system requires high availability, the computer should be equipped with at least a RAID array and a redundant power supply.

Note

On systems with a very high number of measuring locations (>96) you should choose 64 bit mode when installing the software (x64 operating system required). This means that the *ibaPDA* Server service runs as an x64 component and thus allows better utilization of computer resources (RAM memory, CPU, etc.).

Virtual machines

The use of iba software in virtualized Windows systems is generally possible if no special PCI or PCIe cards are needed and a slight latency can be tolerated. Thus, operation of *ibaPDA* using Ethernet-based interfaces (TCP/IP, UDP, etc.) in a virtual environment is possible.

Operating iba cards (ibaFOB-D, -Dexp, -SD, -TDC, etc.) in virtualized environments is not permitted.

Provision of the licenses stored on a USB dongle may require what is known as a dongle server, which can be purchased from iba AG. If necessary, speak with your iba representative or iba Support.

For new systems, the “WIBU license” licensing method is used. Here, the licenses can be provided on a USB dongle or as a soft license. The soft license is particularly well suited for use in virtualized environments.

Coordination and selection of the virtualization layer (e.g. VMWare, Microsoft- V, Citrix, XEN, Oracle etc.) and/or the use of hardware interfaces via I/O pass through are not the responsibility of iba AG.

2.4 Emergency mode

Functioning of the *ibaQDR* system is generally an essential factor in production, without which quality analysis and documentation for a product or process is extremely restricted, if not impossible. Therefore, the risk factors that can lead to an interruption in acquisition need to be minimized.

As is normal with an *ibaPDA* system, there are regular checks for a valid license. When using USB dongles, the dongle is periodically scanned. If no dongle is installed, or the required license is missing from the dongle, data acquisition is stopped or the corresponding function is terminated.

This is different for *ibaQDR*. If *ibaQDR* storage is in progress and the dongle is damaged or removed, or the soft license is suddenly invalid or not available, the system continues to run in emergency mode. This emergency mode is maintained for four days unless acquisition or storage is stopped in another way.

The period of four days gives users the opportunity to temporarily remove a dongle during operation, e.g. to install a license upgrade. If a dongle is damaged or lost, it should be sufficient time to either obtain a replacement dongle from iba AG Support or to get to the next scheduled production stoppage.

When using soft licenses, a new soft license can be installed during operation.

The license and dongle status can be monitored in various ways:

- Information on SNMP server (...*PDA**General**Licensing**Is valid and … Time limit* objects)
- Information on OPC UA server (*PDA**Licensing**Is valid and … Time limit* tags)
- Virtual signal with DongleInfo function (Info type 0 and/or 3)
  For further information see 8.2.4.2, page 102

In addition, a missing dongle/license is indicated in the display on the *ibaPDA* client:

- Entry in event log
- Conspicuous message on all connected clients daily at approx. 09:30 warning that the license is missing.

3 Basic principles

3.1 Properties

- *ibaQDR* is based on the *ibaPDA* software product.
- *ibaQDR* is an add-on to *ibaPDA* providing a special form of data store.
- In terms of the signal scope and the number of measuring locations, *ibaQDR* is scalable, which means that it can be used for both small and large lines.
- The same system requirements apply as for *ibaPDA*.
- *ibaQDR* can also be easily integrated into existing systems.
- All measured values can be stored as length or time-based.
- Video synchronization signals from *ibaCapture* can also be stored, for example allowing length-based analysis of video data using *ibaAnalyzer*.
- The measured values are mapped to the length of the finished product in standardized form in a final data file.
- In addition to finished product data from the last measuring location, entry product specific data files can also be generated at each measuring location during the process, each containing the data for the process steps completed up to that point.
- At the core of the *ibaQDR* function are the measuring locations, which follow the flow of material along the line and are defined wherever quality-related measured values occur. There are different types of measuring locations to represent different line types and process technologies.
- Changes to the length of the product during the process, e.g. elongation due to rolling, are taken into account, as are special cuts or strip breaks.
- The maximum possible length resolution in the finished product file is determined by the maximum line speed, the maximum material elongation, and the sampling rate.
- The standard length resolution is 1 meter.
  Finer resolutions into the centimeter range or mapping in different units, e.g. “ft”, are possible with a sufficient sampling rate. Coarse resolutions, e.g. 10 m, as common in wire rolling mills, are possible as well.ml
- In addition to length-based data stores, time-based *ibaPDA* data stores can also be created in parallel, e.g. for consumption values or product-independent measurements.
- Virtual signals and special functions enable further adjustments to be made in *ibaQDR*, for example to achieve improved tracking.
- Additional information such as scalars or text data (e.g. product ID, setup values, etc.) can be transferred to *ibaQDR* using text signals and incorporated into the data file.
- The *ibaQDR* data files can be processed and analyzed with no problems using standard tools such as *ibaDatCoordinator* and *ibaAnalyzer*.

- The *ibaDatCoordinator-DB* extension enables the stored data to be automatically provided to other systems such as MES (manufacturing execution system), DataWarehouse or individual applications in a simple, transparent database structure. The most common database systems are supported.
- Using *ibaDatCoordinator*, *ibaAnalyzer* and the report generator, intergrated in *ibaAnalyzer*, highly informative quality reports for each product can be generated automatically.

3.2 Functional principle

The core function of *ibaQDR* is accurately assigning quality data to the product along its length.

ibaQDR carries out this assignment based on the initial time-based data store. At the end of the process, the measured values occurring with a delay are precisely assigned to the product at the point where they were measured.

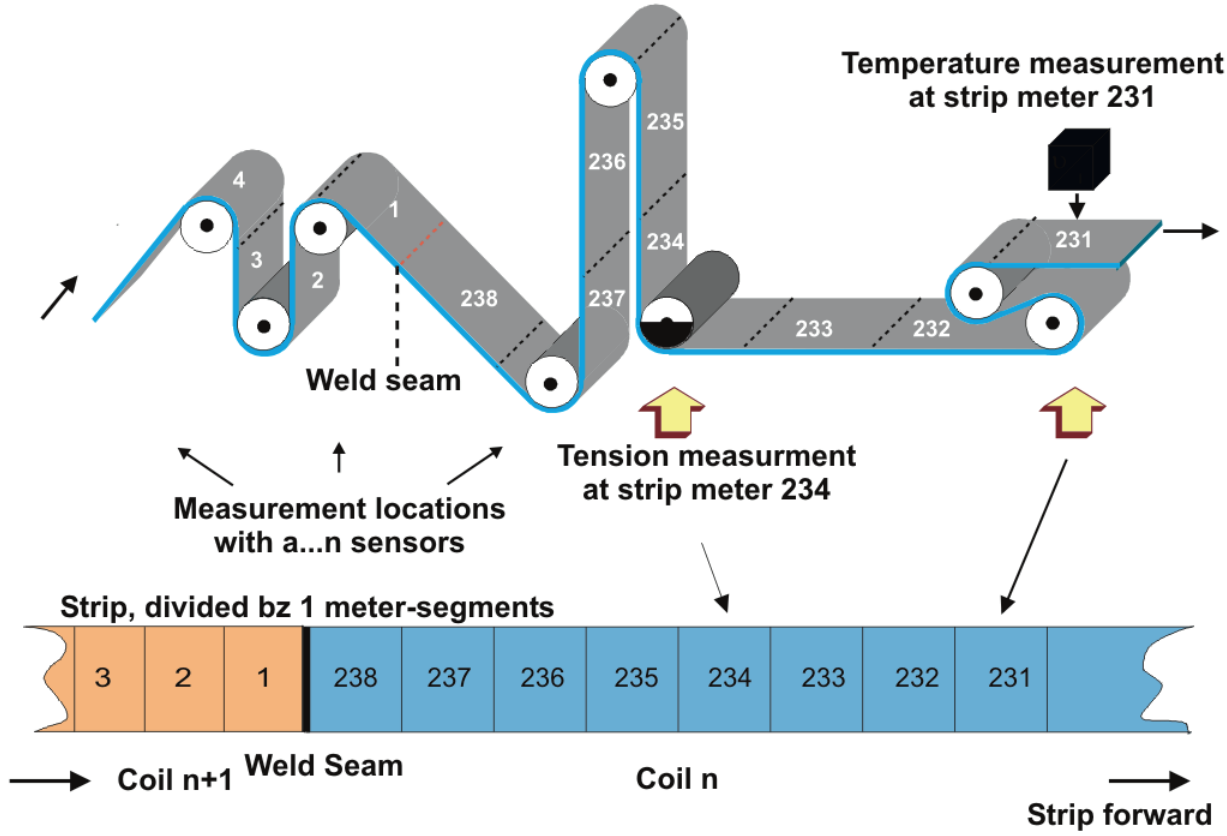
Data relevant for quality includes:

- Product dimensions (width, thickness, etc.)
- Profiles (strip coating, thickness cross-sectional profile, etc.)
- Production parameters (temperature, tension, pressure, etc.)
- Setpoints and measured consumption values, etc.

However, as an additional application for more in-depth process analysis and optimization using *ibaQDR*, it can also be useful to include more extensive signal data such as status bits, control data, etc. in the *ibaQDR* store. These *ibaQDR* applications may then cover several thousand signals.

The production line is split into several sections (measuring locations) in which data relevant to quality is generated. In addition the product, for example a steel strip, is divided into length segments, which are tracked through the line by a material tracking function.

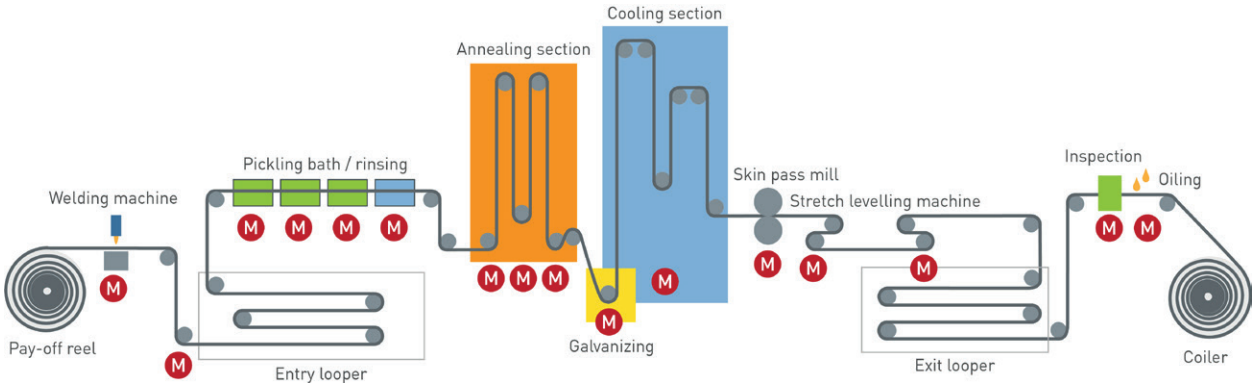
The figure below illustrates this principle using the example of a strip processing line.



The measured data is initially stored separately by measuring locations, and the tracking ID and the position (length) of the product relative to the relevant measuring location are also stored.

These measuring locations can be thought of as small individual *ibaPDA* systems along the overall line, which the product passes through in sequence over time. The number of measurement files produced is the same as the number of measuring locations.

The following graphic shows a simplified example of a hot-dip galvanizing line with measuring locations defined (red circles with “M”).



In this example, the measuring locations are:

- Uncoiler 1 + 2
- Welder
- Measuring roll 1
- Annealing
- Galvanizing
- Cooling
- Skin pass mill
- Stretch leveler
- Thickness gauge
- Shear

The completion signal for a product (e.g. cut at the end of a strip) filters out the measured data from the individual measuring location files by tracking ID and length value and writes it to a finished product file. This file now contains all measured values belonging to the product that has just been produced, standardized to the finished product length.

Material elongation due to production processes is taken into account. In addition to the finished product files, data relating to the entry strip can also be stored, or “intermediate files” can be generated during the process, which include all data up to a particular measuring location.

However, it is important to note that the length standardization here also takes into account strip elongation and thus standardization is to the last measuring location.

3.3 Parallel operation of ibaQDR and ibaPDA

Because the *ibaQDR* function is merely an add-on to the *ibaPDA* software and thus all the *ibaPDA* features can be simultaneously used with the *ibaQDR* function on a system, it would be easy to consider doing without a separate *ibaPDA* system. This is not recommended because of the different objectives of the systems. In actual fact, a separate *ibaPDA* system should always be available when using an *ibaQDR* system.

The *ibaPDA* system is used for diagnostics and analysis of fast-moving process events. It is the preferred solution for maintenance, monitoring, and fault analysis. Therefore, it is often necessary to change the signal configuration, e.g. to apply new signals. However, every change to the I/O configuration in *ibaPDA* requires acquisition to be stopped and restarted.

By contrast, the *ibaQDR* system requires uninterrupted operation when production is in progress. Every time acquisition stops, *ibaQDR* loses synchronization with the production process, which means that the data acquired up to that point is normally lost and the finished product files can no longer be generated. Each time acquisition is restarted, synchronization with the production process must be carried out first. This means that an entire (dummy) product has to pass through the line before all measuring locations are re-synchronized. This prevents spontaneous work, as is sometimes necessary during maintenance.

In addition, the measurement signals for the *ibaQDR* system are usually defined once and are then not, or only very infrequently, changed. These configuration changes should be carried out during scheduled system stoppages.

Some of the process signals to be measured are often required by both systems. Depending on the interfaces used to transfer the measurement signals from the process automation system to *ibaPDA* or *ibaQDR*, different solutions for supplying the data to the two systems are possible.

PROFIBUS DP, Ethernet TCP/IP and other bus systems: Measured values generated by the automation system and to be stored in both systems (*ibaPDA* and *ibaQDR*) are transferred in separate telegrams, once to the *ibaPDA* system and once to the *ibaQDR* system.

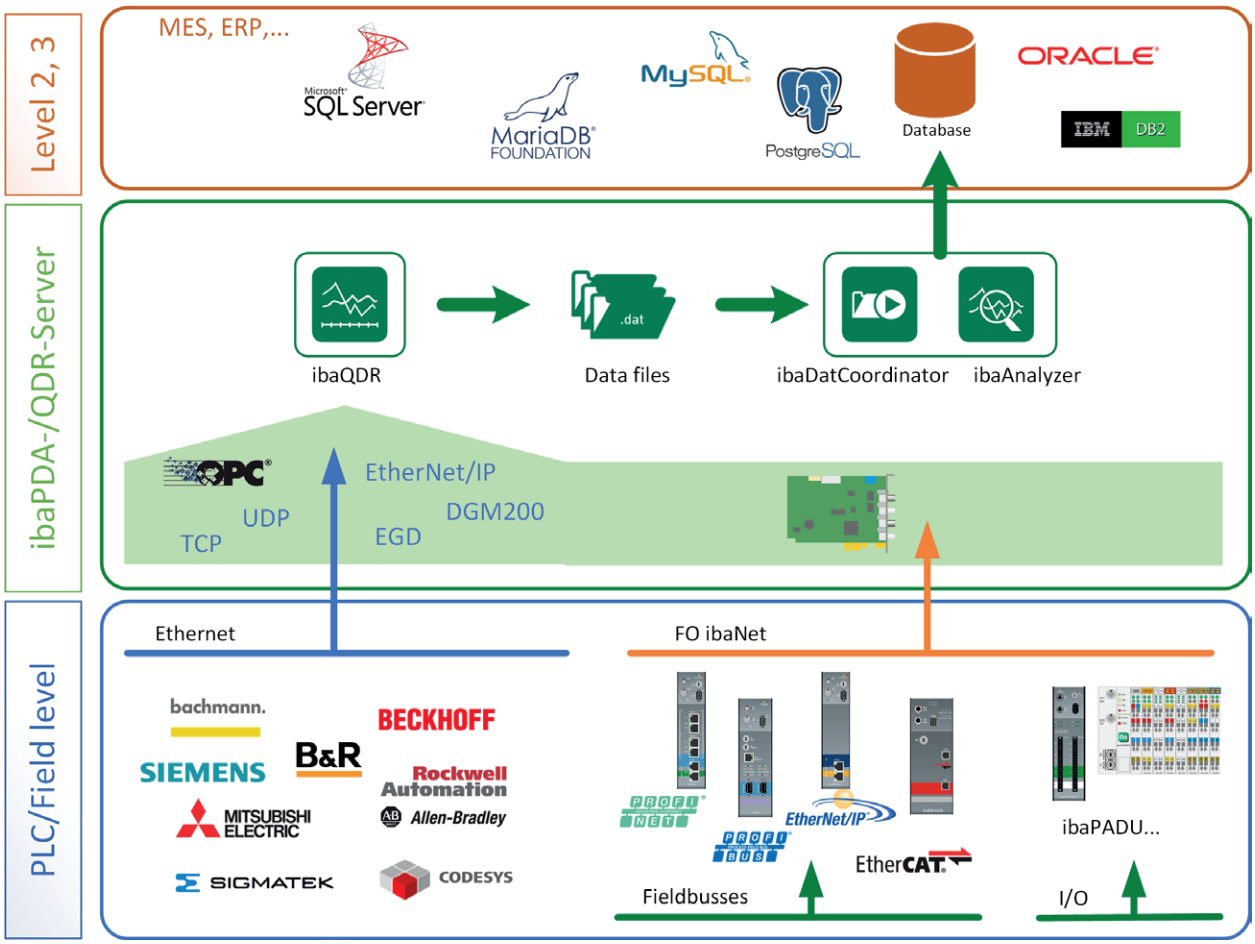
ibaNet: If required, measured values supplied via optical fibers (e.g. from *ibaPADU*, *ibaLink*, *ibaFOB*,...) can be forwarded directly from one system to the other using *ibaFOB-OF-Link* interface connections or *ibaBM-FOX-i-3o* modules (optical fiber multipliers) and fiber optic cables. When using *ibaFOB-OF-Link* interface connections, the signals should first be routed to the *ibaQDR* system and then forwarded from there, as *ibaQDR* is usually operating constantly.

ibaNet-E: The required signals can be received by both *ibaQDR* and *ibaPDA* via this Ethernet-based protocol, provided that ibaNet-E-compatible devices are the data source, e.g. ibaM-COM.

Furthermore, it is useful to use the *ibaPDA* system for storage and analysis of the installed material tracking system in the basic automation (level 1).

In any case, a time-based standard store with all signals should always be configured and used on an *ibaQDR* system alongside the QDR store. Storage can be carried out locally and in a relatively short time. It provides practical assistance in support cases (playback) and with installation and configuration of the system.

3.4 Topology and data flow



The *idaQDR* system is typically located between the basic automation system (level 1) and the production monitoring or production planning level (level 2).

It can be connected to the basic automation system in a variety of ways. The appropriate interfaces for the control type and manufacturer are used. These could be Ethernet-based interfaces, field buses or discrete digital and analog signals.

Storage of quality data can either be carried out locally on the *ibaQDR* computer or on an additional file server in the network.

The material tracking signals and the measured values relevant for quality come from the basic automation system. In addition to the pure tracking and measurement signals, further data corresponding to the relevant product can also be transferred, for example from level 2. This information, which is often alphanumeric, can be saved as text signals in the data file along with the measured data. However, it can also be used to name the data files, so that each data file can be clearly assigned to the product.

Using the SQL interfaces in *ibaPDA*, it may also be possible to incorporate the additional information directly from higher-level systems.

The product-specific data must be transferred from the basic automation system and from level 2 to *ibaQDR* with the correct time and location as a “live stream”. This means that *ibaQDR*, level 2, material tracking, and the basic automation system must be synchronized with respect to the product (“tracked part”).

The data files stored on the *ibaQDR* computer or a file server can be automatically analyzed, processed and extracted into databases using a combination of *ibaDatCoordinator* and *ibaAnalyzer*.

3.5 Material tracking

This section describes the signal processing and the tasks involved in material tracking.

Example illustrations are based on a strip processing line.

3.5.1 Material tracking function

As the name suggests, the material tracking function involves tracking the material that is processed to a create a finished product as it passes through a line or a process.

The primary purpose of this is to provide the components of the line or the processing machines with information about where the material is located and when particular processing operations are to be started or new setpoints adopted. The start and end of a piece of material are particularly important, for example to adjust the lateral guides in good time when changing material or to adopt new setpoints for controlling the thickness of a mill stand.

For *ibaQDR*, the length of the material is also important, as the ultimate aim is to map the measured values to the length of the finished product.

In terms of the tracking data, each piece of material is assigned a tracking ID, which is normally linked to the material or product ID from the production planning system.

Depending on the line type, the material tracking system must be capable of tracking multiple products (tracked parts) that are simultaneously located in the line.

For processes such as a strip processing line, the material tracking system also has to be able to deal with different production scenarios (1:n, n:1 and n:m production).

3.5.2 Signals from material tracking system for ibaQDR

Ideally, the material tracking system supplies the following signals that can be used for *ibaQDR* (example of strip processing line):

- Measured strip length values or length counter for the individual line sections, where available
- Internal tracking IDs for the tracked parts in these sections
- Trigger signal for completion of tracked parts (e.g. cut signal from exit shear)
- Operating statuses of uncoilers and coilers and / or active entry paths and the speeds in these sections

From text signals (e.g. from Level 2):

- Product ID of next product coil
- Setpoints
- Further information such as customer ID, material class, setpoints, tolerances, etc.

What if there are not sufficient signals?
If there is no material tracking available, or the material tracking system installed only supplies very rudimentary or imprecise data, it is possible to compensate for this deficit to a limited extent in *ibaQDR* itself. Special functions and virtual signals can be used to determine or calculate the required information.

You can find information about this under ↗ *Adjustments with virtual signals, page 87*

If a good supply of data to *ibaQDR* is not feasible using this method, either more effective material tracking should be planned or expansion of a PLC program should be considered.

Reducing the number of tracking signals required
On very large lines with a high number of measuring locations, the signal volume at the interface between the material tracking and basic automation systems or with *ibaQDR* can be considerable.

ibaQDR provides effective functions to enable this signal volume to be reduced.

For example, it is possible to use the same speed signal for multiple measuring locations if they are located within a line section with the same material speed. This is the purpose of the *measuring locations with offset*, where only the relevant distance (offset) to a *standard measuring location* has to be known.

Tip

For measuring locations with offset, neither a measured length value nor a tracking ID is required. To reduce the number of signals between the material tracking system and *ibaQDR*, it is worth investigating where measuring locations with offset can be used. The measured length value and tracking ID are then only required for the standard measuring location to which the offsets refer.

3.6 Important definitions

To gain a better basic understanding, some of the *ibaQDR*-specific terms are explained below.

3.6.1 Measuring location

A measuring location is a point or a tightly delineated section of the line (e.g. brush cleaning, furnace pre-heating zone or skin pass mill) at which one or more measurement signals are acquired.

The following types of measuring locations can be defined in the *ibaQDR* system:

- Standard measuring location
- Measuring location with offset (after a standard measuring location)
- Measuring location in front of the tracking start
- Reversing measuring location
- Dividing measuring location

A further sub-type “Tracking start” is provided once by default (generally a standard measuring location).

For measuring locations in front of the tracking start, the distance to the tracking start measuring location behind it must be specified, and for measuring locations with offset, the distance to the previous standard measuring location.

At a standard measuring location, product elongation can be taken into account.

At a reversing measuring location, product elongation and the number of reversing operations can be taken into account.

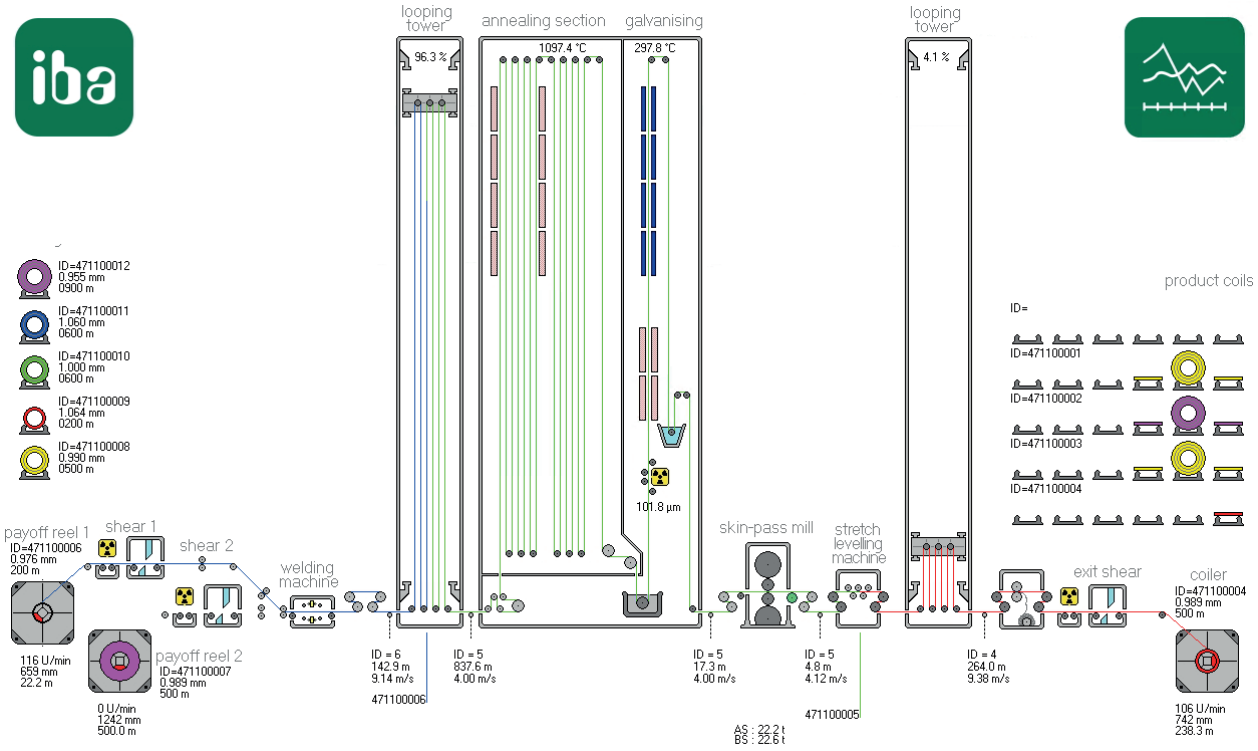
For correct product management (tracking) and mapping of measured values to the correct length, each measuring location also requires the information from the tracking ID and one or two length measurement signals.

For further information about the measuring locations and their configuration, see ↗ *Creating and configuring the measuring locations, page 45*.

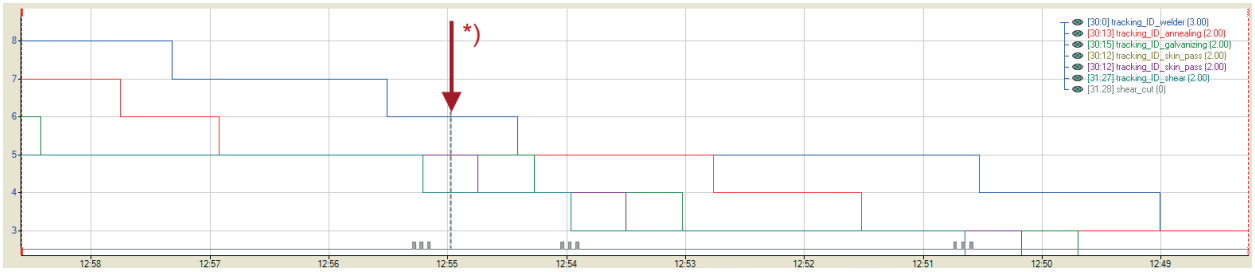
3.6.2 Tracking signals - Tracking ID

Tracking IDs are special input signals that are required for accurate material or product tracking in the line. *ibaQDR* uses these signals for internal management of the measured data, identification of the start of the product, the end of the product, and a change of product, and for subsequent precise assignment of the measurement values to the length.

The following HMI screenshot contains three tracking IDs (ID = 4 red, ID = 5 green and ID = 6 blue) in the line.



In *ibaPDA* or *ibaQDR* the tracking IDs are displayed as follows in the graph:



**)The snapshot in the HMI screen above is currently suitable. Welder ID = 6, Annealing, Galvanizing and Skin Pass ID = 5, Shear ID = 4*

Tracking IDs must be unique in the line, i.e. a tracking ID may only represent a single complete section (tracked part). When a tracked part has completely left the line, the tracking ID can be assigned again.

In large lines, the tracking IDs are usually generated by a special control unit for material or weld seam tracking and transferred to *ibaQDR*. In smaller lines, this function can also be performed by a PLC.

In certain cases, it is also possible to use the virtual functions in *ibaPDA* to calculate corresponding signals, if the line control unit not fully provide them.

3.6.3 Tracking signals - Length counter

For correct mapping of the measured values over the product length, *ibaQDR* requires length signals, which are normally transferred from the material tracking system.

Some lines, mainly newer ones, have actual value sensors that can output measured length values. *ibaQDR* can process these measured length values directly for the data store.

If no length signals are available, the length value can also be calculated as a virtual signal using the calculation functions in the ibaPDA expression builder. This can be done easily by integration of a speed signal. Calculations based on a motor speed, taking into account the gear factor and roll diameter, can also be easily implemented.

Older lines often still contain actual value sensors that provide meter pulses. A meter pulse indicates that the product has advanced by 1 m in the relevant section of the line. The material tracking system pre-processes the meter pulses and provides them to *ibaQDR* as binary signals using a suitable interface.

As only analog values can be entered for the length signals at a measuring location, in this case the meter pulse must first be converted into an analog length value using the virtual signals (expression builder).

More information about the individual functions and application examples can be found in chapter ↗ *Adjustments with virtual signals, page 87*

3.6.4 Tracking signals - Reference to finished product

At least a reference to the naming scheme in the higher-level systems and thus meaningful naming of the *ibaQDR* DAT file is essential. This text signal must exist before the *ibaQDR* DAT file is generated and is generally provided by Level 2/3 or the MES via a TCP/IP-telegram. As mentioned already, it is also possible to export this data directly from a database using the DB interface.

If the product name (e.g. the finished coil number) is not yet defined immediately before completion, it is possible to specify the final name later using *ibaDatCoordinator* (data update task). If temporary autonomous operation of the line without Level 2/3 (MES) is possible, the reference information should be provided directly by Level 1.

3.6.5 Time base

Measured value acquisition in the *ibaQDR* system is based on the acquisition time base. The measured values are acquired using the timing from this time base. The choice of time base influences the computer load and should be adjusted for the features of the relevant line. The maximum line speed is the most important design criterion. The frequency of the division cuts and the rate of the finished products should also be taken into account.

Particularly for the tracking signals it is essential to make sure that the required length resolution is assured in the *ibaQDR* system, i.e. there is at least one time-based raw measured value for each measured value in the *ibaQDR* DAT file.

Example

Maximum line speed

300 m/min or 5 m/sec

Required length resolution in ibaQDR

0.5 m

Selected time base

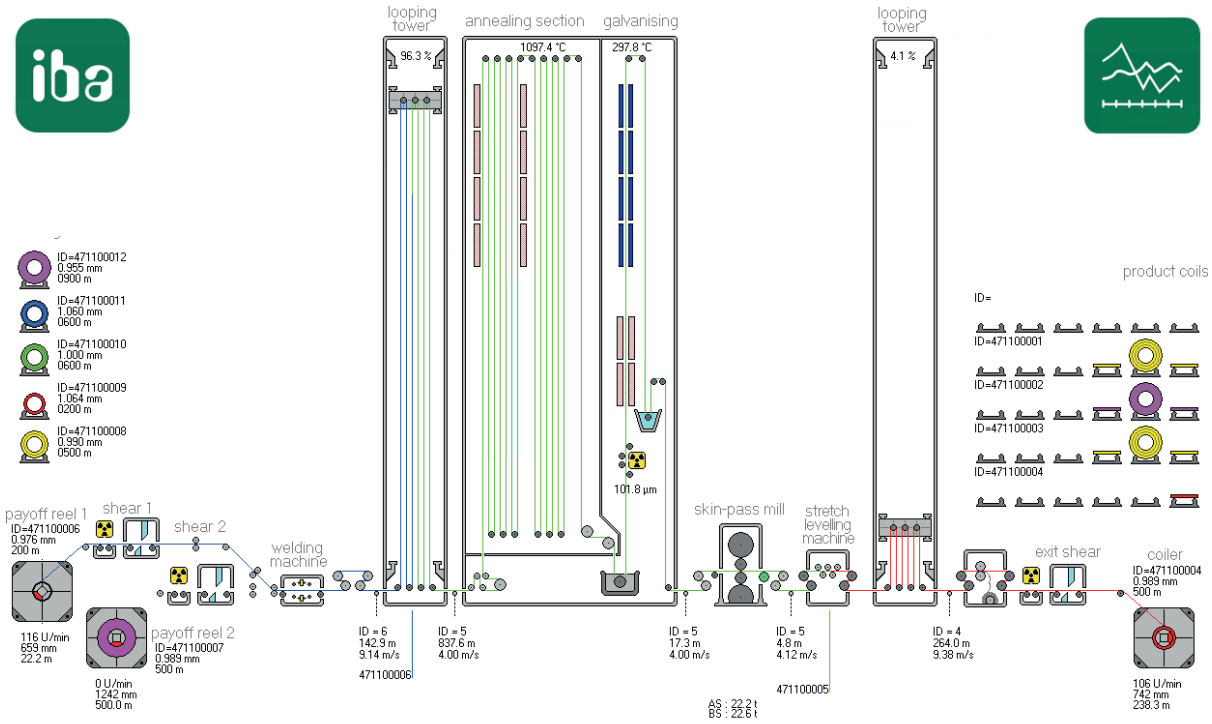
To reliably acquire every 0.5 meter of the strip, a time base of < 100 ms should be set. However, setting a time base of ≤ 50 ms (corresponding to ≥ 20 samples per second) is strongly recommended.

4 Project planning

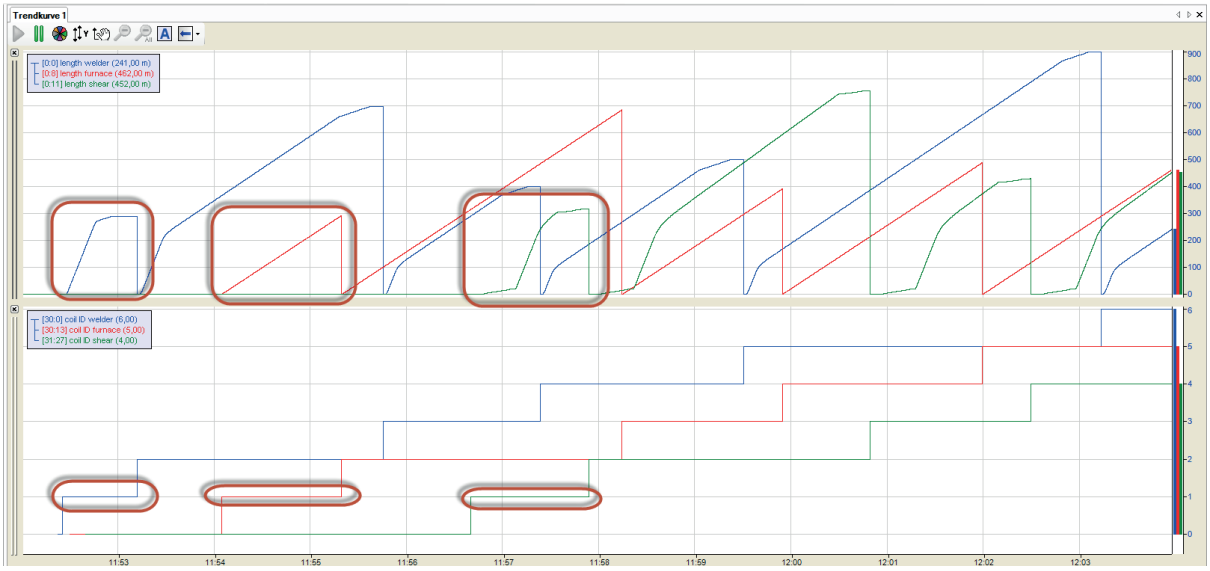
This chapter describes the essential project planning steps in preparation for configuration of the system.

4.1 Project planning guidelines

1. Identify the line type and the associated line properties (e.g. maximum speeds, operating modes and functions, such as reversing mode, etc.). A line diagram (Visio or AutoCAD drawing, HMI overview diagram or similar) should always be available.
   Example: HMI overview of QDR simulator

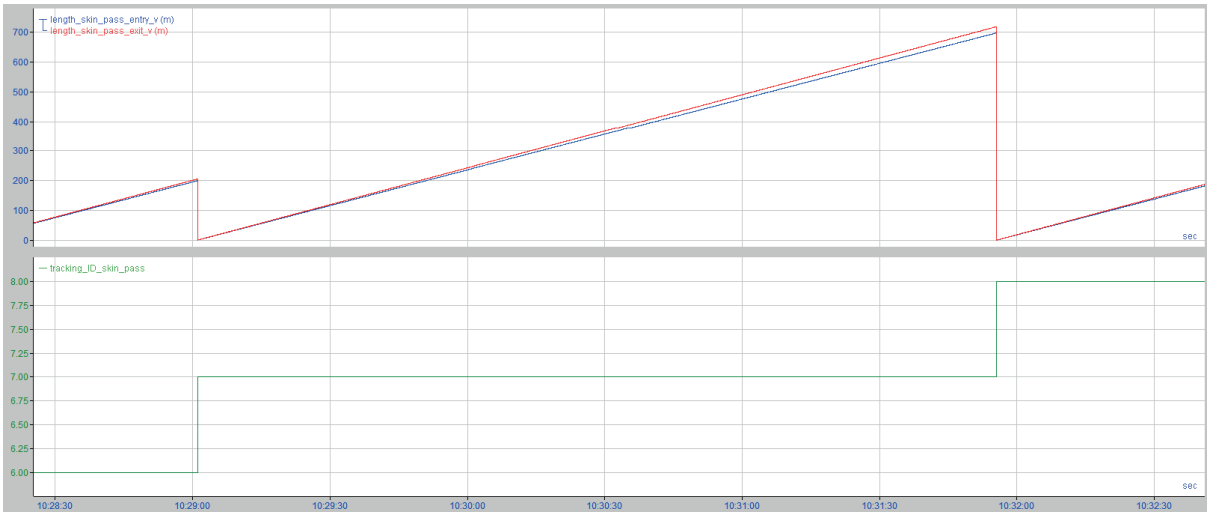


2. Clarify which data is relevant for quality in your application and therefore needs to be stored. It may be useful to involve the quality department at this stage.
3. Clarify the data sources and thus also the data interfaces that are required for *ibaQDR* to acquire the measurement signals (including any options requiring a license).
4. Identify the line sections with different speeds than one another, for example, the number and positions of the installed bridle rolls for speed measurement in a strip processing line.



At the top, the figure shows a simplified view of three different speed ranges using the example of length measurement in a strip processing line: Welder in the entry section (blue) with acceleration and deceleration, furnace (red) with constant speed and shear in the exit section (green) with acceleration and deceleration. The graph below shows the tracking IDs.

5. Identify locations in the line where a change of length (elongation) of the product occurs, e.g. roll stands, stretcher leveler, etc.



At the top, the figure shows an example of measured length values for the entry side (blue) and exit side (red) on a skin pass mill (approx. 3% skin pass level). The graph below shows the tracking ID.

6. Identify which sensors or which signals are available for precise material tracking, e.g. hole detectors for weld seam tracking.
7. Define the text telegrams and text signals that *ibaQDR* receives to write the product-specific information to the data file and name the finished product files.
8. Specify the measuring locations along the line (type, number, position).
9. Specify the signals that *ibaQDR* requires from the material tracking system for each section of the line (measured length or speed values, tracking ID, offsets, etc.). This step is best car-

ried out using a spreadsheet, e.g. MS Excel.

No	ML Name	Type	Tracking-ID	Max. Elongation %	Entry Length	Exit Length	Active	Offset	Speed
0	payoff reel 1	BTS					[3.2]	21	[1:0]
10	payoff reel 2	BTS					[3.3]	13	[1:1]
20	welder	STD	[30:0]	0	[0:0]				
30	gauging roll 1	STD	[30:5]	0	[3:0]				
40	annealing	STD	[30:13]	0	[0:8]				
50	galvanizing	OFF						360	
60	cooling	OFF						402	
70	skin pass	STD	[30:12]	10	[1:2]	[1:3]			
80	stretch leveler	OFF						4	
90	thickness exit	STD	[30:10]	0	[1:12]				
100	shear	STD	[31:27]	0	[1:11]				

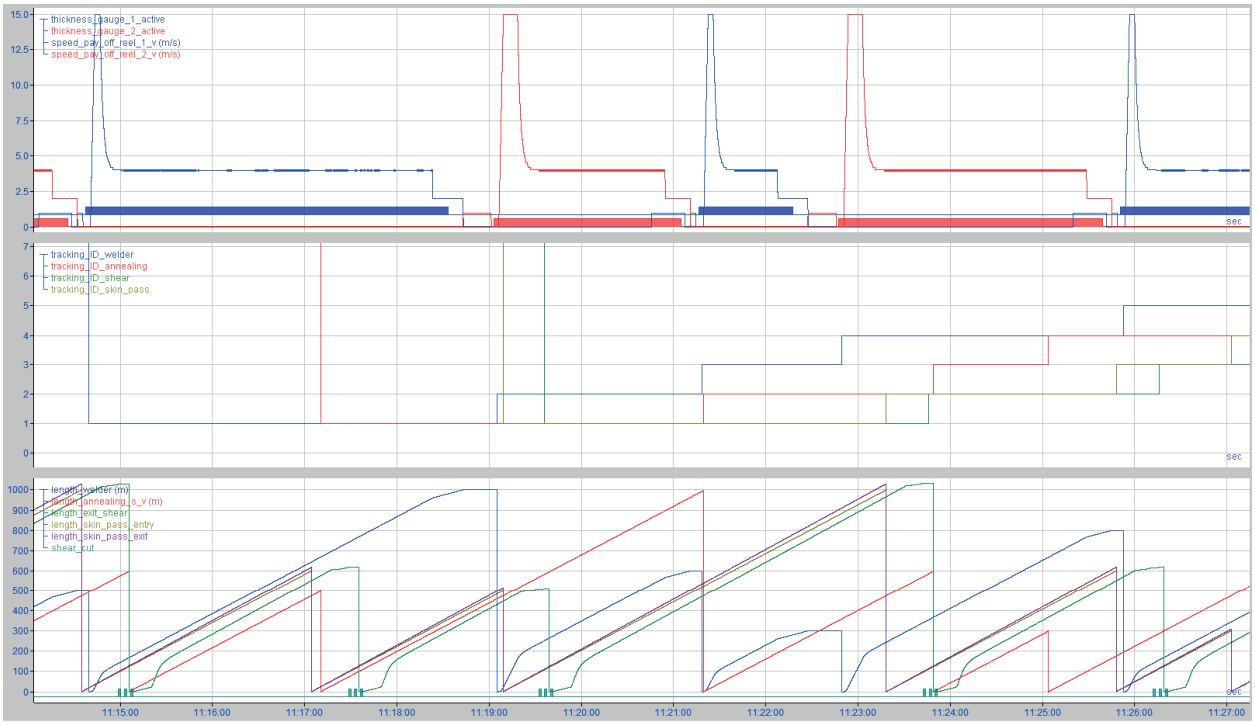
10. Specify the measurement signals to be measured during operation and assign them to the relevant measuring location.
11. Establish the connections between the *ibaQDR* system and the data sources (PLC, level 2).
12. Configure the input signals in the I/O manager (signals from material tracking system, process signals, text signals).
13. Configure the data store.
14. Check the DAT files generated by *ibaQDR*.

4.2 Preparatory actions in ibaAnalyzer

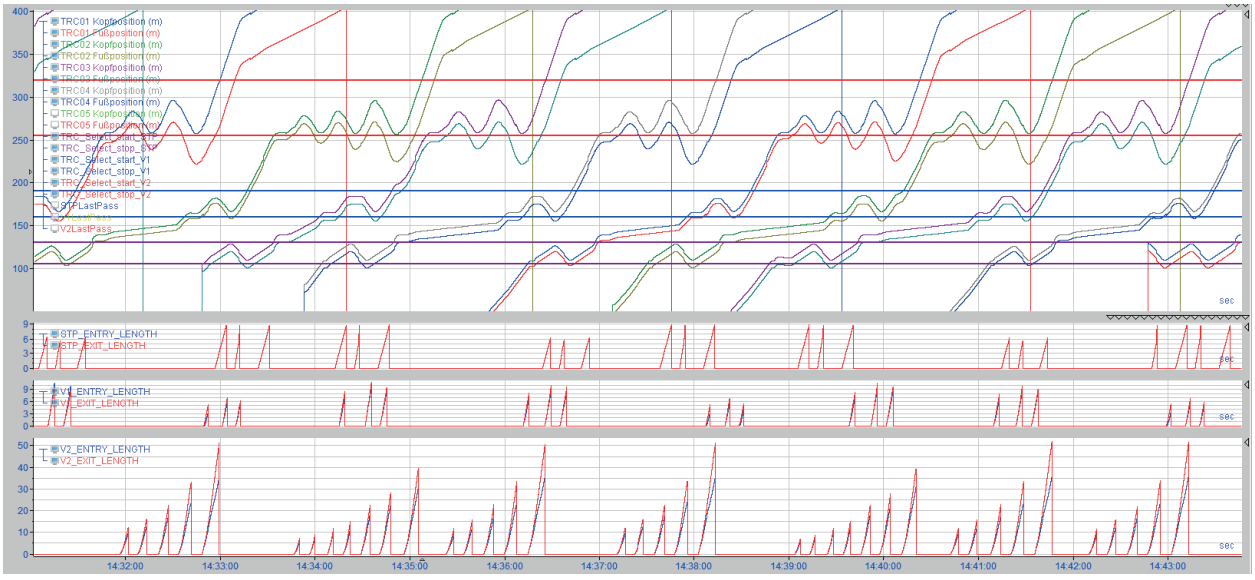
If an *ibaPDA* system is already installed,the available tracking signals should be examined in *ibaAnalyzer*. Here, you can also prepare virtual signals so that they can subsequently be transferred to the *ibaPDA* I/O configuration.

Checking the availability and plausibility of tracking signals

The following figure shows the tracking signals from our CGL-QDR simulation (speeds, tracking IDs, measured length values and digital active/inactive signals for measuring devices).



The following figure shows an example from a hot rolling mill, where the edger and two roughing stands are run in reversing mode. The top graph shows the raw signals from the material tracking system for the head and tail positions of the slabs. The lower three graphs show the length values for each pass calculated in *ibaAnalyzer* based on these raw values.



Checking the required time base and the required length resolution

The time base must be selected so that at least 2 samples are acquired for each required length segment (in this case 1 m) at maximum speed. The more values per length segment, the better.

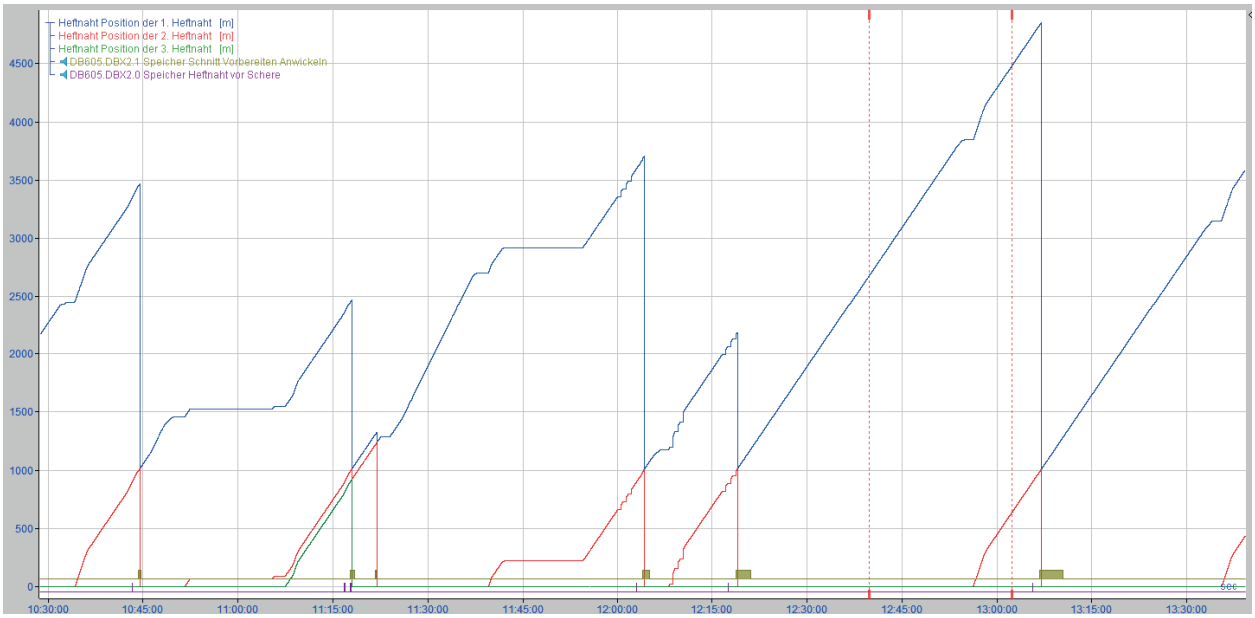
In the following example, we can see that the signals are available with a sampling rate of 10 ms, but the actual resolution differs for the different length signals. However, a 1 m resolution is always possible overall.

The top two graphs show the length at the welder and annealing. There a full meter is traveled over the entire length of the graph (approx. 240 ms). An acquisition time base of 10 ms results in approx. 24 samples per meter. At the shear in the exit section (lower graph) 5 m is traveled in the same period due to the higher speed. The resolution is still sufficient at 4 to 5 samples per meter. The jumps in the length value are always exactly 1 m and no more.

Check for signal quality and accuracy

Examine the usability of the tracking signals in terms of sufficient accuracy for material tracking (resolution of length measurement signals, meter pulses, etc.). Are there any outliers or other anomalies?

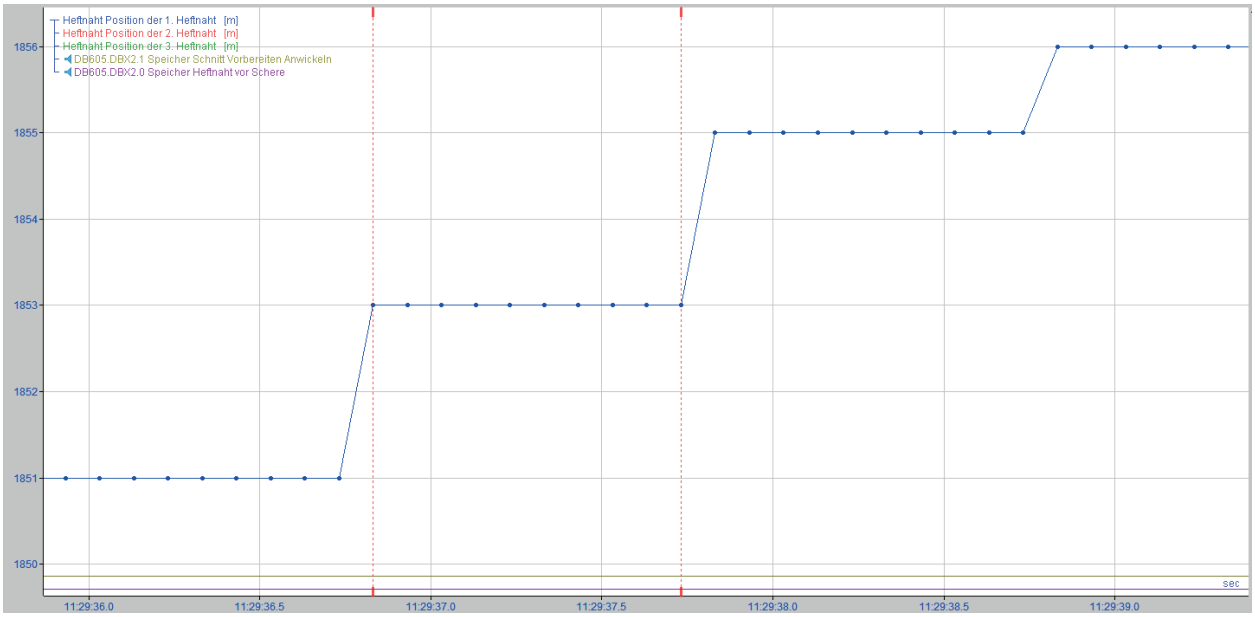
In the following example, anomalies can be identified in the signal form for the length signals.



Check for deterministics in the length values or data loss

If a length signal comes from a PLC, the output cycle must be sufficiently fast that there are still at least two output cycles per length segment (in this case 1m) at maximum speed.

In the following example, we can see that the output cycles of 100 ms would be sufficient, but the length signal actually only has a resolution of 1000 ms and thus the 1 m QDR resolution can no longer be deterministically mapped.



4.3 Planning measuring locations

The best way to create an overview of measuring locations is in a table (Excel sheet). Set up a sufficient number of columns to allow the information to be added gradually.

Based on your technological requirements, identify the necessary measuring locations and specify the type and a name for each of them. Then add the corresponding parameters, e.g. offsets or max. elongation. Assign the appropriate tracking signals to the measuring locations, for example tracking ID, entry and exit length, or speed.

No.	ML name	Type	Offset	Tracking-ID	Entry length	Exit length	Max. elongation	Speed
0	Uncoiler1	BTS	21					[1:0]
10	Uncoiler 2	BTS	13					[1:1]
20	Welder	STD		[30:0]	[0:0]		0	
30	Measuring roll 1	STD		[30:5]	[3:0]		0	
40	Annealing	STD		[30:13]	[0:8]		0	
50	Galvanizing bath	OFF	360					
60	Strip cooling	OFF	402					
70	Skin pass mill	STD		[30:12]	[1:2]	[1:3]	10	
80	Stretch leveler	OFF	4					
90	Thickness gauge	STD		[30:10]	[1:12]		0	
100	Shear	STD		[31:27]	[1:11]		0	

Table 1: Measuring location table for our example

BTS = Before tracking start, STD = Standard, OFF = With offset

Tip

If you have access to *ibaPDA* or *ibaQDR* you can also enter the measuring locations there in the data store configuration.

The advantage is that you get a constant overview of the measuring locations in order (tree structure) and also an overview table showing the assignment of signals, measuring locations, and profiles.

4.4 Planning tracking signals

Specify which signals you require for full control of the *ibaQDR* system.

To do this, it is useful if you have specified the required measuring locations first. For example, measuring locations that are located within the same speed zone can obtain their length signals based on the same speed signal.

Are there measured length values or meter pulses, or are only speed signals available? Depending on whether or not the required signals are available in sufficient quality, you either have to program suitable signals in the PLC or create them in *ibaQDR*. Where you have clear length signals, you should use them.

In addition, for each measuring location (except measuring locations with offset) you need a signal that transmits the tracking ID.

Some measuring locations require additional signals:

- A *measuring location in front of the tracking start* requires an “Active” signal.
- A *dividing measuring location* requires two tracking ID signals.
- A *reversing measuring location* requires signals for the passes.

To ensure a clear configuration, it is advisable to list the signals in a table.

Measuring location	Signal	Signal ID	Signal source
Uncoiler 1	Speed	[1:0]	PLC
Uncoiler 1	Tracking ID		MTS
Uncoiler 1	Active	[2:0]	PLC
Uncoiler 2	Speed	[1:1]	PLC
Uncoiler 2	Tracking ID		MTS
Uncoiler 2	Active	[2:1]	PLC
Welder	Length	Welder	QDR virtual
Welder	Tracking ID	Welder	MTS
...	Length	Measuring roll 1	...
...	Length	Annealing	...
...	...	...	...

Table 2: Example table of tracking signals

4.5 Planning the signal to measuring location assignment

For a comprehensible configuration, it is advisable to define the data relevant to quality for the *ibaQDR* store and list it in a table.

You can then add the signal name, unit and, if applicable, the measuring channel number, and also note the assigned measuring locations and profiles.

Tip

In addition to the measured quality data, you should also store the tracking signals. This helps to verify easily the length mapping of *ibaQDR* in *ibaAnalyzer*.

ID	Signal name	Unit	Measuring location	Profile
[0:0]	Speed	m/s	Uncoiler 1	1m + As is
...	Strip thickness	mm	Thickness gauge 1	1m + As is
...	Speed	m/s	Uncoiler 2	1m + As is
...	Strip thickness	mm	Thickness gauge 2	1m + As is
...	Entry looper strip tension	kN	Bridle roll 1	1m + As is
...	Furnace temperature	°C	Heating zone	1m + As is
...	Galvanizing bath temperature	°C	Galvanizing bath	1m + As is
...	Galvanizing layer thickness	g/m²	Air knife	1m + As is
...	Rolling force	kN	Skin pass mill	1m + As is
...	...	...	...	...

Table 3: Example signal table

5 Configuring a QDR data store

The core function of *ibaQDR*, namely mapping measured values to the length of the product, is configured in the form of the *ibaQDR* data store.

This is where you map the line with the measuring points, define the length and time resolution for the data store and specify what the finished product files will be called.

There is only one single *ibaQDR* data store per *ibaQDR* system.

Note

It is strongly recommended to configure not more than one plant in an *ibaQDR* system in order to avoid interdependencies concerning configuration changes or maintenance intervals.

5.1 Profile

General information about profiles can be found in the manual for *ibaPDA*, Part 5.

A special “Time/Length” profile type is available for an *ibaQDR* data store.

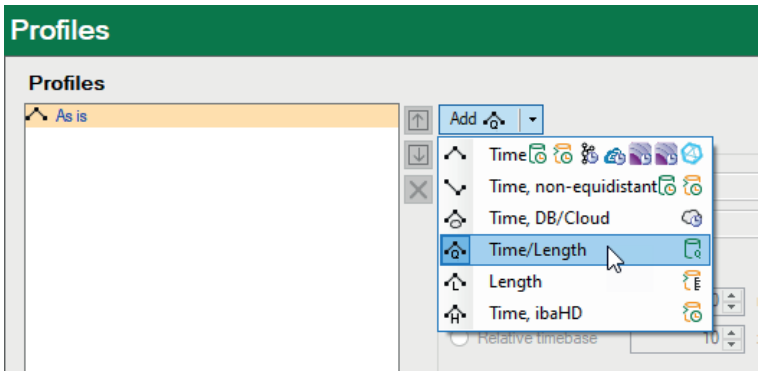
Before you configure the actual data store, you should create at least one suitable profile for the *ibaQDR* data store, so that you can assign the measurement signals later.

You assign the signals and profile later when configuring the measuring locations.

Adding a profile

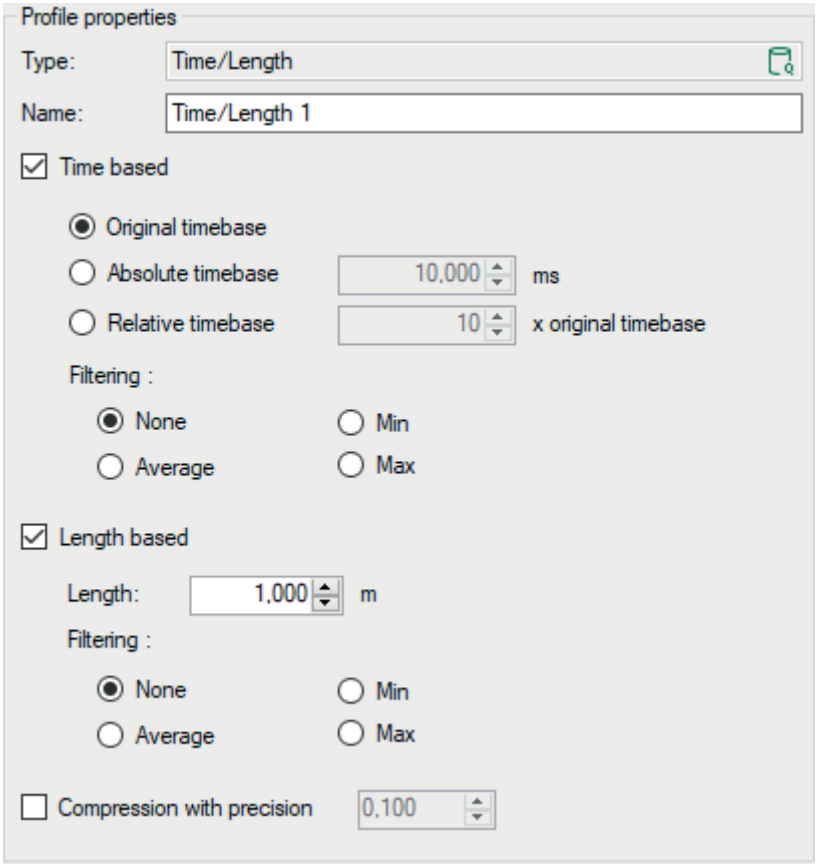
Open the data store configuration and select the Profiles node in the top left.

Expand the <Add> button on the right of the dialog and select Time/Length.



You can now configure the profile properties.

Profile properties



Type
Shows the type with the appropriate store types (symbols)

Name
Enter an unambiguous and understandable name here.

Time-based, length-based
By enabling one of these options or both of them, you determine how the data is written to the data file.

For a time-based store, the settings are the same as for the “Time” profile.

For a length-based store, the settings are made accordingly. A length interval can be specified here instead of a time interval. The default setting is 1 m. You can enter both higher and lower values.

Tip

If you enable both store types, you will have advantages later when examining the data in *ibaAnalyzer*. You can then easily switch between a length-based and time-based representation. Particularly during commissioning and troubleshooting, comparing the time and length representations can be very useful.

Filter
For both time-based and length-based stores, you can use filters.

- None: The current value is stored (first sample).
- Min, Max, Average: The corresponding calculated value is saved. The acquired values within a storage interval are used in the calculation. For length-based *ibaQDR* data, as there are normally a large number of “raw values” for a length interval (e.g. 1 m), the best approach should be considered carefully. It is also possible to define multiple profiles, each with different length intervals (see note below).

Note

If you want to store Average, Max. and Min. values, you must first have made copies of the original measurement signals using the virtual signals. You can then assign each of these signals to a different profile, one for the average values, one for the maximums, and one for the minimums.

Example:

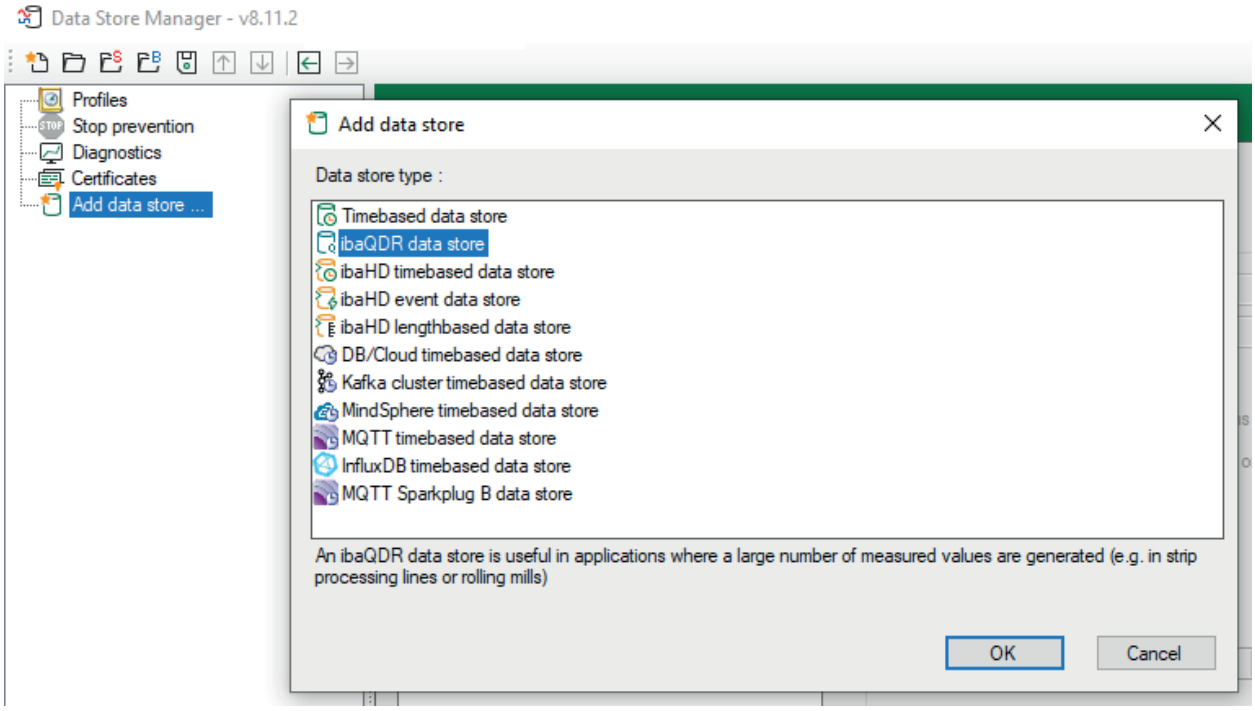
Compression with precision
Here you can enter a value to achieve further compression of the stored data volume. If this option is enabled, a new measured value is only saved if it differs from the preceding measured value by more than the amount entered. This allows you to define a tolerance band for changes of values, e.g. to prevent “noise” from a measuring device unnecessary inflating the data file as long as the changes in value are within a reasonable range.

Note

The value to be entered is an absolute value, which always relates to the scaling of the measurement signal to which the relevant profile is linked. If you want to use this option and assign different signals to this profile, make sure that the value entered is appropriate for all linked signals. For example, the value 1.0 may be appropriate for a temperature measurement (in °C) but not for a thickness measurement (in mm).

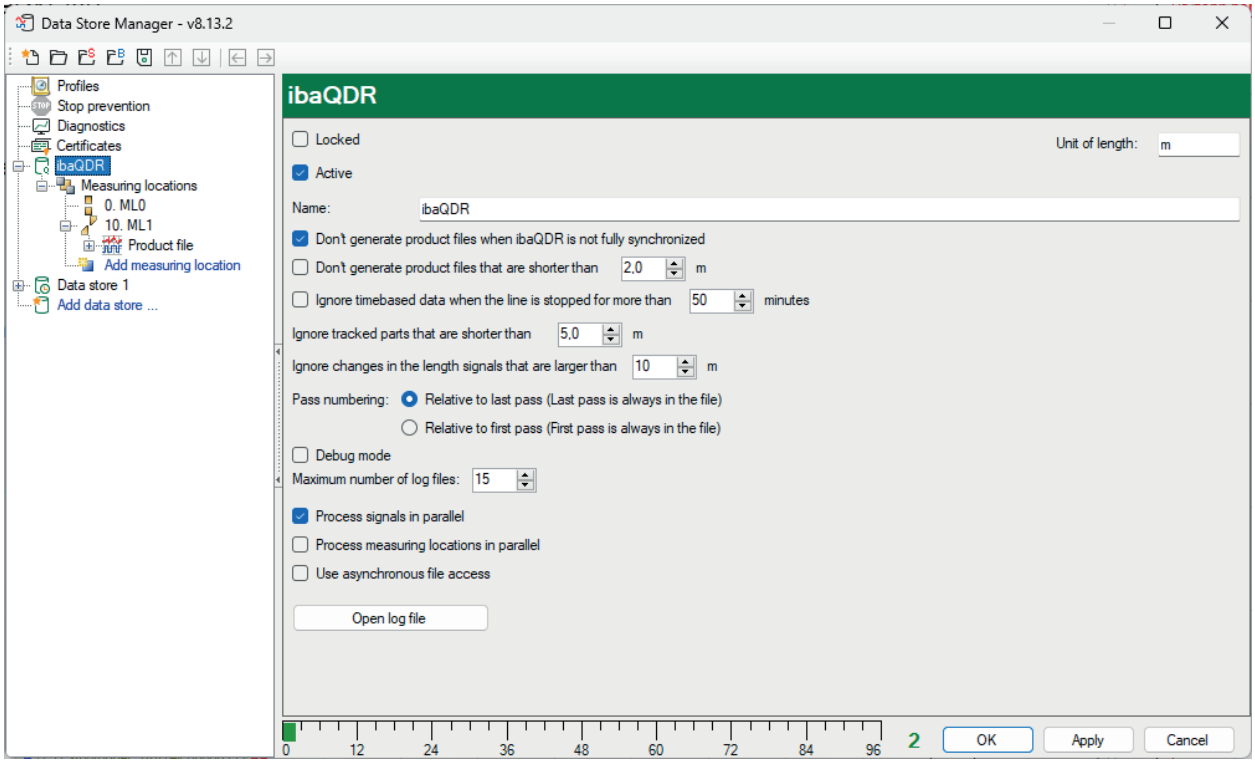
5.2 Data store - General settings

Add an *ibaQDR* data store by selecting *Add data store...* in the *Data Store Manager*.



Select “ibaQDR data store” and confirm with <OK>.

5.2.1 Basic settings for the data store



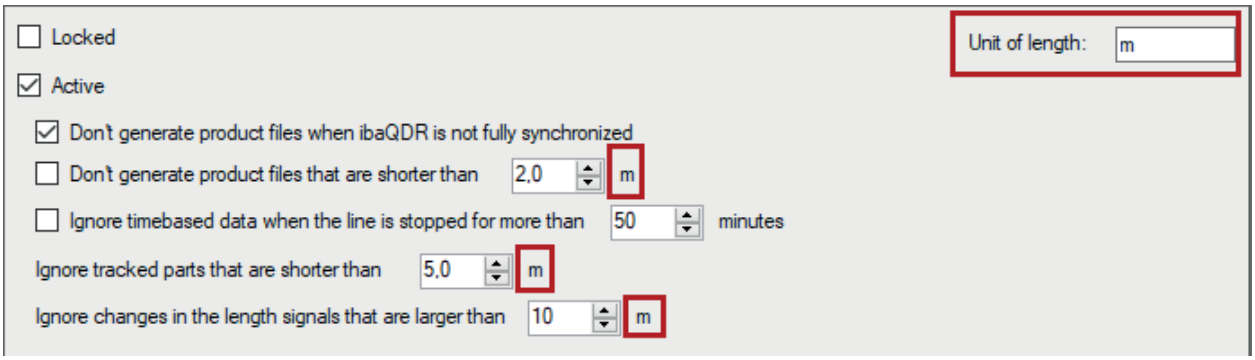
Locked
If you enable this option, a data store configuration can only be changed or disabled if the current user has the corresponding rights with the user management enabled.

Active
Data store must be enabled in order to work. The advantage of configuration and activation being separated is that you can define and configure several data stores without being obliged to use them all at the same time when you disable the store you do not require. Simply disable the data stores you do not want to use.

To enable a data store, click on the major branch of the relevant data store and check the *Active* option.

Name
Enter a comprehensible name for the data store here. When using multiple QDR data stores, take advantage of the naming in order to distinguish the data stores.

Length unit
Set the base unit of length (default m). This string is then reused in all subsequent dialogs (up to *ibaAnalyzer*). You can enter another unit, e.g. “ft” (feet) here instead of “m”.



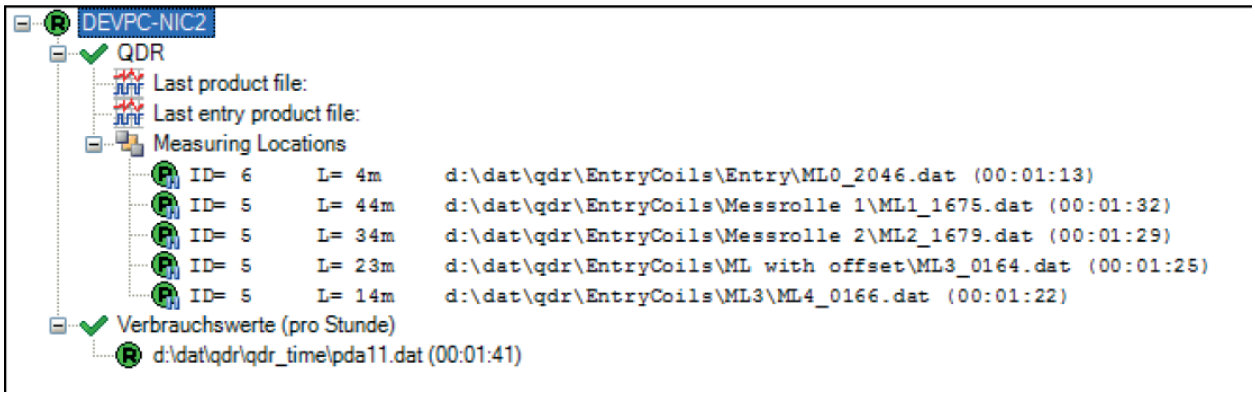
Don't generate product files when ibaQDR is not fully synchronized
If you enable this option, only complete *ibaQDR* files are generated. However, when starting data storage it means that no *ibaQDR* file is generated for all entry products currently in the line. Particularly for “long lines” such as pickling tandem lines (PLTCMs) this option is frequently disabled as a compromise, in order to at least obtain files with data for the last measuring locations for the entry products already in the line.

Don't generate product files that are shorter than ...
This option allows scrap cuts to be hidden and thus prevents product files being generated for incomplete strip sections. The option is definitely advisable if a large number of sheet cuts are made using flying shears in the exit section of a line. It is also possible that with short scrap cuts, for example, (e.g. 0.2 m) there is no guarantee that data will be available for all measuring locations.

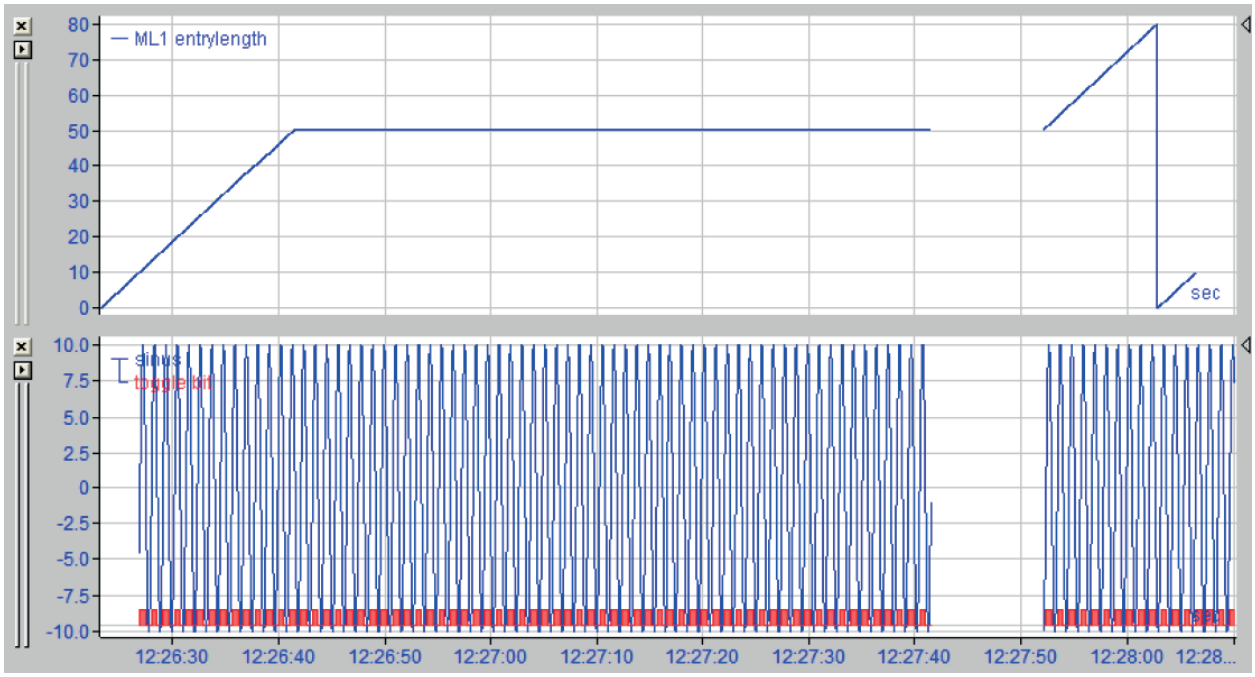
Ignore timebased data when the line is stopped for more than ... minutes
If the entire line is stopped, i.e. all length signals are constant for more than ... minutes, it does not generally make sense to continue storing data for length mapping. Particularly for longer stoppages, very large data volumes would accumulate for the entry products, which could lead to problems when restarting and generating the *ibaQDR* files (memory overflow, etc.). Setting

an appropriate value here is strongly recommended. The time-based data from the stopped sections can also be taken from parallel configured *ibaPDA* stores if required.

If a line stop is detected, *ibaQDR* reports this in the event log and the measuring locations are flagged with a pause icon in the data store status window:



This would produce the following in the *ibaQDR* data file:



Ignore tracked parts that are shorter than ...
In the entry section of strip lines, it is possible that very short tracking IDs are generated at the first measuring location (e.g. welder) due to movement back and forth, and these do not need to be tracked further. This setting can be used to filter them out.

Ignore changes in the length signals that are larger than
Large “jumps” in the length counters (with the exception of the reset at the transition from one tracked part to the next) generally indicate an error and can be filtered out.

Pass numbering
This setting is only relevant if there is a reversing measuring location in the plant. It helps to identify correctly the last pass, e.g. in a reversing mill. Depending on whether the numbering of

the last pass should always be the maximum pass number or the real number of passes choose one of the two options.

If you select *Relative to last pass*, the number of the last pass is always the maximum number as configured under *Maximum number of passes* in the measuring location configuration, regardless of the actual number of passes done. Thus, the number of the last pass is always the same and further analyses of the data files can always refer to the same number if the last pass is relevant.

Example for "Relative to last pass"
Setting for maximum number of passes at reversing measuring location: 7

If all seven passes are done, the numbering is as follows:

First pass = 1, ..., last path = 7

If only five passes are done, numbering is as follows:

First pass = 3, ..., last pass = 7

If you select *Relative to first pass*, the pass number of the last pass depends on the number of passes actually done. Take this into account, when identifying the last pass in further data file analyses.

Example for "Relative to first pass"
Setting for maximum number of passes at reversing measuring location: 7

If all seven passes are done, the numbering is as follows:

First pass = 1, ..., last path = 7

If only five passes are done, numbering is as follows:

First pass = 1, ..., last pass = 5

Process signals in parallel
With more than 32 signals per measuring location, this enables you to enable or disable parallel processing, which results in a performance improvement particularly on modern multi-core computers.

Debug mode
If you enable this option, detailed messages are displayed in the *ibaPDA* event log window. Enabling the option is recommended.

Maximum number of log files
Here, you can set how many log files are stored (cleanup). If the system is only checked very sporadically, it may be useful to increase the value. One file is normally created each day. The value specifies how many days the log files are available for.

<Open log file> button
You can use this button to open the current log file (qdrLog.txt) in the default text editor.

Process measuring locations in parallel
If you enable this option, then more resources (processor cores) of the computer will be used in favor of processing the measuring locations. Enabling this option is recommended if you have a multi-core computer and many measuring locations or if you use multiple QDR data stores, e.g. as for multi-strand caster lines or hot rolling mills.

Use asynchronous file access
Enabling this option decouples data acquisition and data storage, i.e., data acquisition and data storage are then processed in different threads. This prevents data acquisition from being negatively affected by slow file access times during data storage. When using multiple physical drives for different data stores, if one drive is ever unavailable, only the corresponding store is affected. The other stores can continue writing. The disadvantage of this operating mode is a higher memory requirement. The buffer memory for write commands to be executed is 500 MB per data store. If this limit is reached, then the recording stops and restarts after 5 s.

Enable this option if you use multiple QDR data stores in parallel.

Note

If asynchronous file access is enabled, please note in case of producing partial coils from a parent coil (1-to-n production), that product files of the partial coils are not generated until the entry file is fully completed. This can result in delays in the completion of product files.

5.2.2 Basic settings for measuring locations

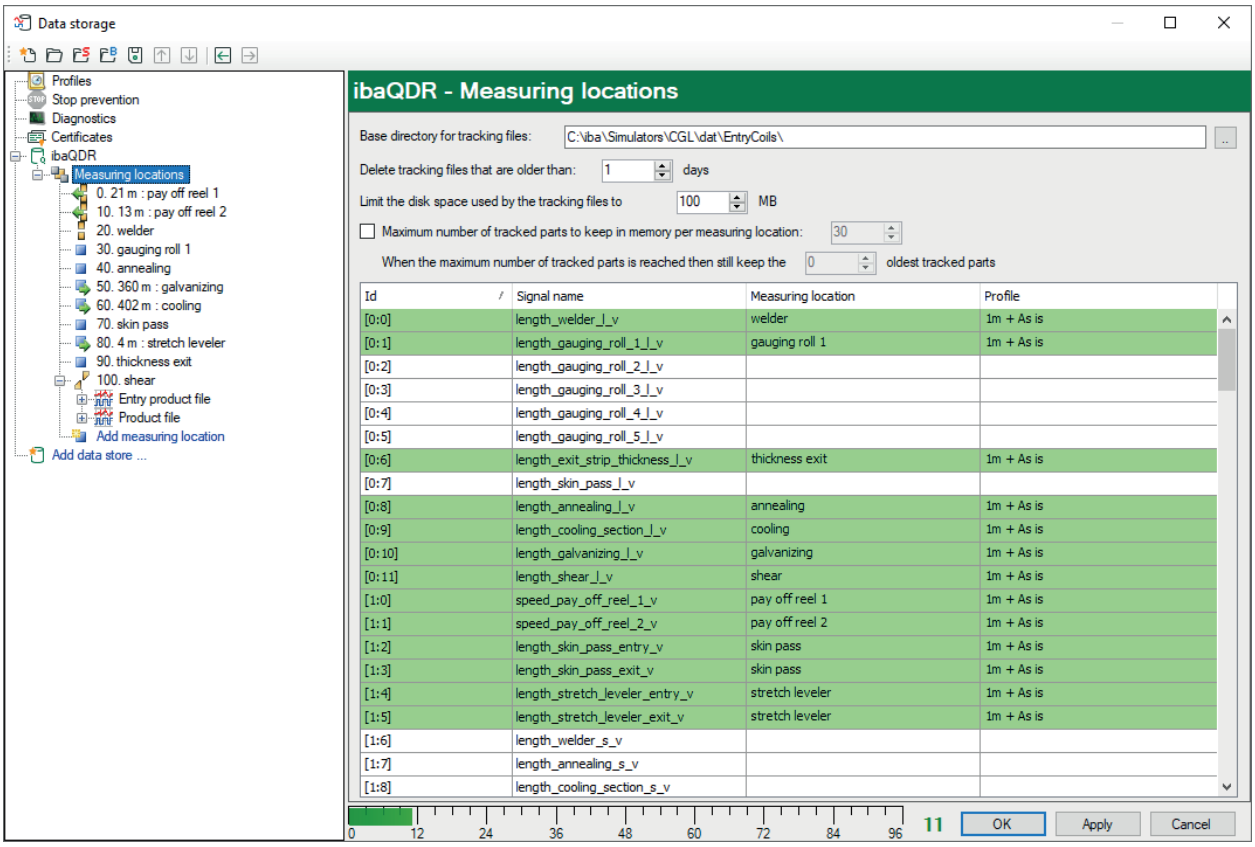
In the navigation tree for the *ibaQDR* data store, under “Measuring locations” you can make settings for saving the temporary measuring location data stores.

There is a default path preconfigured as base directory for these temporary files: `c:\iba\QDR\EntryCoils`. For each configured measurement location a corresponding sub-folder will be created automatically. If needed, you may modify the base directory path and make settings for the cleanup strategy.

When one of the set time or memory limits is reached, the oldest files are discarded or overwritten.

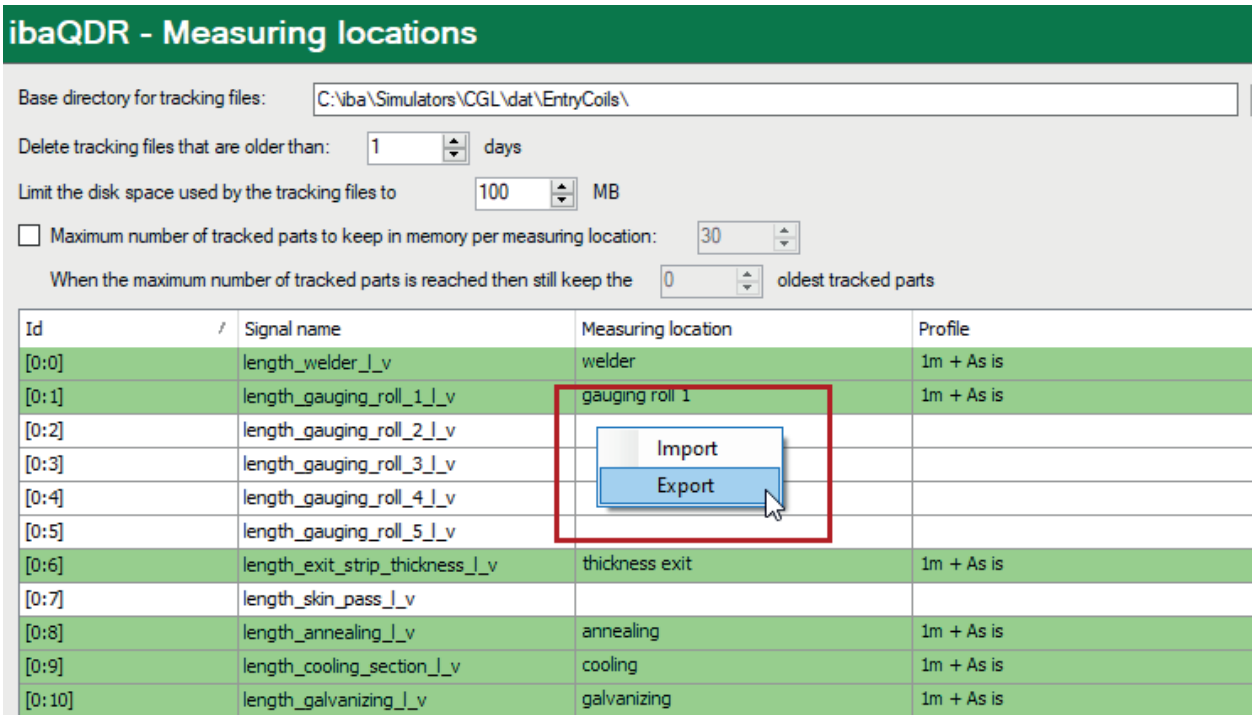
Maximum number of tracked parts to keep in memory per measuring location
For very special requirements, it may be necessary to make additional settings to ensure that on lines with a changing strip travel distance all relevant temporary measuring location files are still provided after a changeover.

Use this option to control the handling of “old” tracked part files.



In the lower section of the dialog is a signal table, which shows all available signals with those already assigned to a measuring location highlighted in green. The assigned measuring location and the profile used are also displayed.

The export and import functions in this table are very useful and can be performed simply by right-clicking in the table.

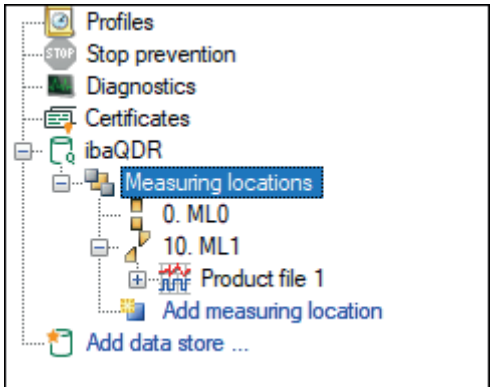


The exported signal and measuring location list is a text file, which you can easily edit in a text editor or spreadsheet program (e.g. MS Excel). Particularly in large systems with a high number of signals and measuring locations, this enables you to make editing of the signal assignments more efficient. You can then re-import the list.

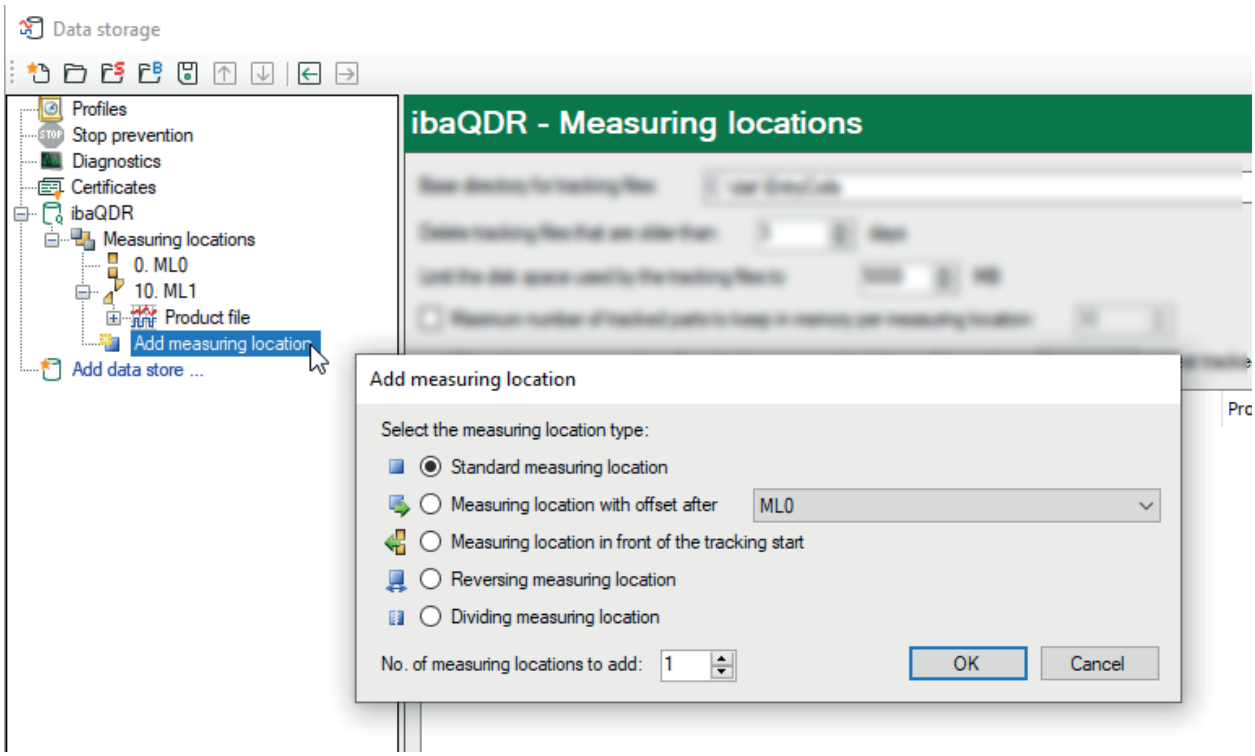
The document is also very well suited for documentation of the *ibaQDR* system and could be provided to the quality department if required.

5.3 Creating and configuring the measuring locations

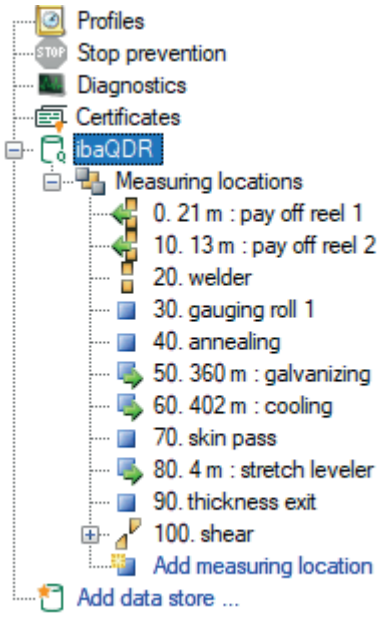
A new *ibaQDR* configuration contains two initial measuring locations corresponding to the beginning of material tracking (tracking start) and the end of tracking. On a continuous strip processing line, for example, these would be the welder and the shear.



Further measuring locations can be added using *Add measuring location*. There are five types of measuring locations, which are described in detail below.



The measuring locations in the tree structure can be moved or rearranged using drag and drop. It is important that they have the exact same sequence as in the production line.

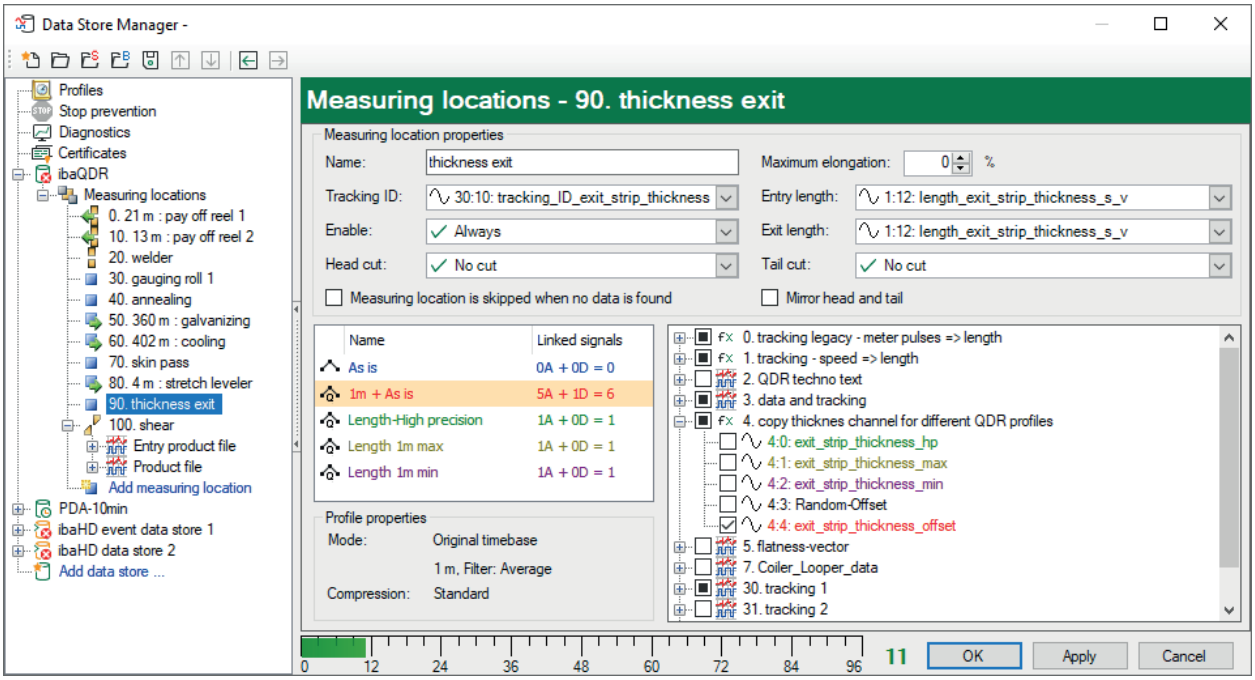


The numbering of the measuring locations is used for sorting. The numbering of measuring locations is specified in increments of 10.

For *measuring locations in front of the tracking start* and *measuring locations with offset*, the relevant offset values are displayed in the tree structure and used for sorting in relation to the reference location.

For all measuring location types, the tracking information is configured in the upper section, and in the lower section the signals for the measuring location are selected with the relevant profile. Note that a signal can only be assigned to one measuring location with only one profile.

In the figure below, different profiles have been created for the thickness signals, and the raw thickness signal has been appropriately duplicated first in the virtual Module 4.



Tip

For tracking and fault analysis, storage of the tracking signals as both length and time-based at each measuring location is strongly recommended.

During signal assignment, you can also assign *ibaCapture* signals. This allows length-based video data analysis later in *ibaAnalyzer*.

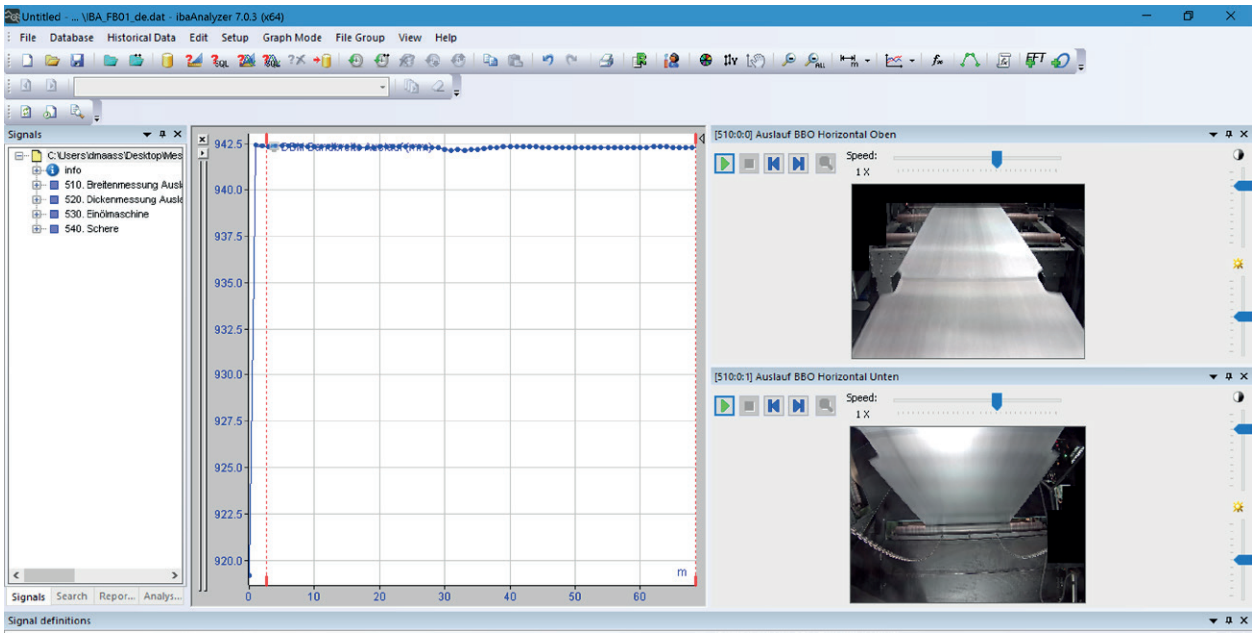


Image source: ThyssenKrupp Steel Europe AG

5.3.1 Standard measuring location

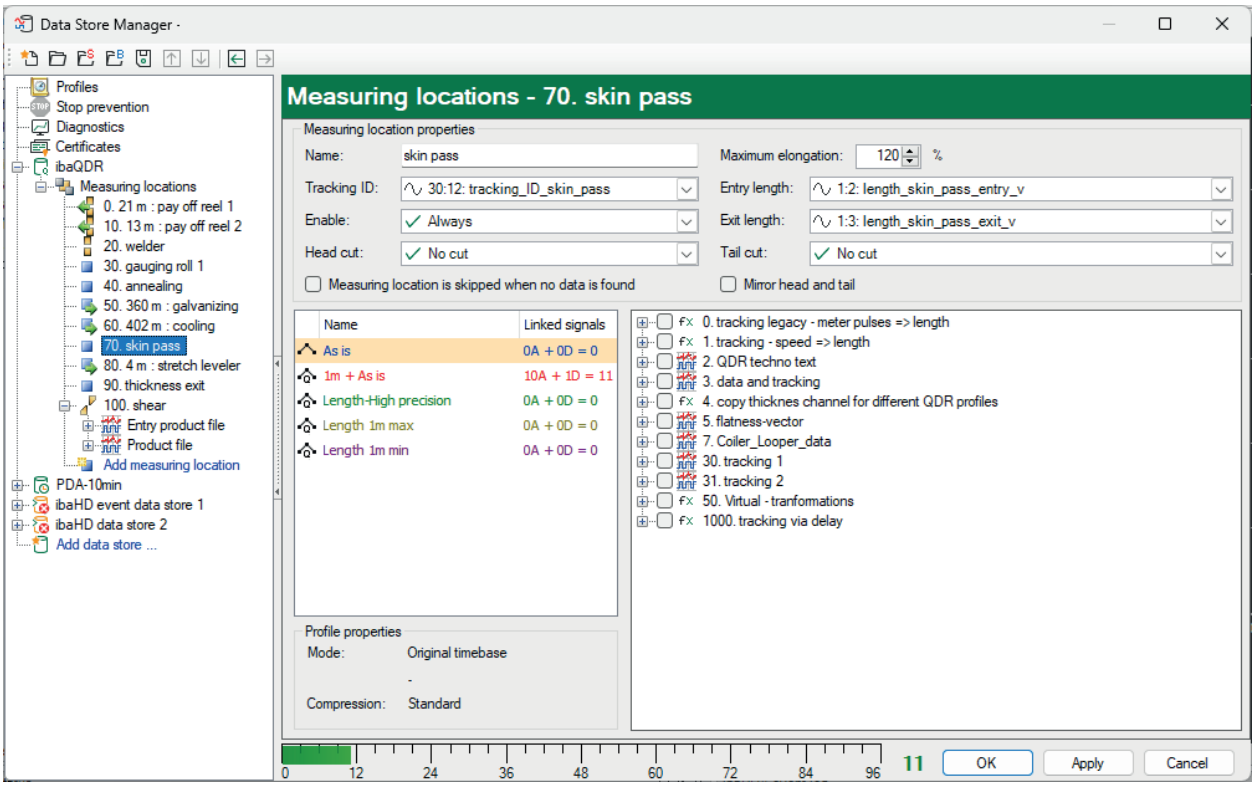
Description

A standard measuring location is a measuring location for universal use, with the properties described below. Standard measuring locations are used in all line types, particularly for the purpose of tracking synchronization and at locations with material elongation.

For each standard measuring location, a matching signal for the tracking ID and the entry and exit length must be provided in the material tracking system.

Typical applications

- as measuring location for tracking synchronization (pinch hole detectors)
- all line types
- thickness and width measuring devices
- roll stands or skin pass mills (not reversing)



Measuring location properties

Name

Enter a unique name for the measuring location here.

Tracking ID

The tracking ID signal is generated by the material tracking system and a change in its value controls the beginning and end of a “tracked part” (e.g. coil) at a measuring location. This means that each time the tracking ID signal value changes, a data file is closed and a new one is started.

It is an internal ID from material tracking, which is represented using an analog signal. Therefore, only analog signals are available in the drop-down window for selecting the tracking ID signal. Text signals are not supported here.

Enable

The *Enable* option was originally introduced to represent different line layouts or operating modes. However, there are now almost always easier solutions for this, so the default value “Always” is advisable for a new configuration.

Signals are only stored if the measuring location is enabled. If the measuring location is not enabled, the signals assigned to it are not stored and the measuring location is not included in the product file.

A measuring location is enabled by default (“Always”). The measuring locations ML0 (tracking start) and the final measuring location (exit shear) are enabled as a fixed setting. For all other *Standard measuring locations* and *Measuring locations in front of the tracking start* (welder), the setting is variable.

The enable setting can be “Always” or “Never”, or can be controlled by a (digital) signal (Enable = log. 1). All digital input signals and virtual signals can be selected. Signal-controlled enable can be used to enable or disable measuring locations depending on operating requirements, e.g. if parts of the line are bypassed or machines are not engaged. The enable signals are constantly checked for changes. If a change of state lasting longer than 2 s is detected, *ibaQDR* data storage is stopped and then automatically started with the new configuration.

The measuring location is skipped if no data is found
Enable this option, if measuring data may be missing temporarily while the plant and the ibaQDR system should run on. This can be the case, e.g. if machinery is temporarily taken out of operation for operational or maintenance purposes.

By enabling this option, the configuration of an "Enable" signal becomes obsolete in many cases

Maximum elongation
This value can be used to specify the estimated or known maximum elongation in % that the material undergoes when passing the measuring location. This value is then used by *ibaQDR* to calculate the required length resolution at this measuring location to guarantee basic accuracy in the product file.

Typical application: Roll stands and skin pass mills, stretch levelers

Note

The value should be as realistic as possible and an excessively high value should not be selected for safety, as internal resampling is initiated. In a hot finishing section with seven stands, 100% elongation per stand would therefore mean a maximum elongation factor of 2^7 = 128, which would mean that with a target resolution of 1 m the data before the finishing mill would have to be analyzed with approx. 1 cm resolution.

Entry and exit length
For most measuring locations the entry and exit length of the material will be the same as the material does not undergo any elongation. However, on a roll stand or skin pass mill, and sometimes also on furnaces and stretch levelers, the thickness of the material is reduced, leading to an expansion in the length.

The entry and exit lengths therefore differ and so two length measurement signals are required to enable the measured value mapping to be carried out correctly. *ibaQDR* uses the different entry and exit lengths to stretch the entry material to the final length when the finished product is created by the cut at the exit shear.

When generating the *ibaQDR* measurement file, *ibaQDR* checks whether the exit length of each measuring location matches the entry length of the subsequent measuring location.

See also chapter ↗ *Length transformations in the data file, page 69*.

Head cut/Tail cut
In case, there is a crosscut shear to be used for crop cuts at strip head and strip tail in front of this measuring location, the value of the cut length can be taken into account. This ensures the correct length-related mapping of the measured values.

Select here the signals which transmit the total cut length of head and tail cuts. If there is no shear in front of this measuring location, select "No cut".

For more information about head and tail cuts, see chapter ↗ *Reversing measuring location, page 56*.

Mirror head and tail
Enable this option for a measuring location if the production direction has reversed compared to the previous measuring location. For example, this is the case in beam rolling mills after lateral transportation of the material to the next roller table, where production runs in the opposite direction.

5.3.2 Measuring location with offset after

Description

A measuring location with offset is used for measurement signals that are acquired at a distance x (offset) in the material flow direction after a standard measuring location. Therefore, a standard measuring location must be defined first, before a measuring location with offset can be configured.

The offset is the distance, specified in the base unit of length (m, cm, ft, etc.), that the material has to travel between the standard measuring location and the measuring location with offset based on the line geometry. This value can be determined from engineering drawings or by measurement.

A measuring location with offset does not require tracking signals such as *Tracking ID* or *Entry/Exit length*. Measuring locations with offset therefore have the advantage that they allow the number of standard measuring locations and thus of the material tracking signals *Tracking ID* and *Entry/Exit length* to be minimized. Provided the line geometry does not change and the speed is identical after a standard measuring location, downstream measurement signals or sensors can be assigned to one or more measuring locations with offset.

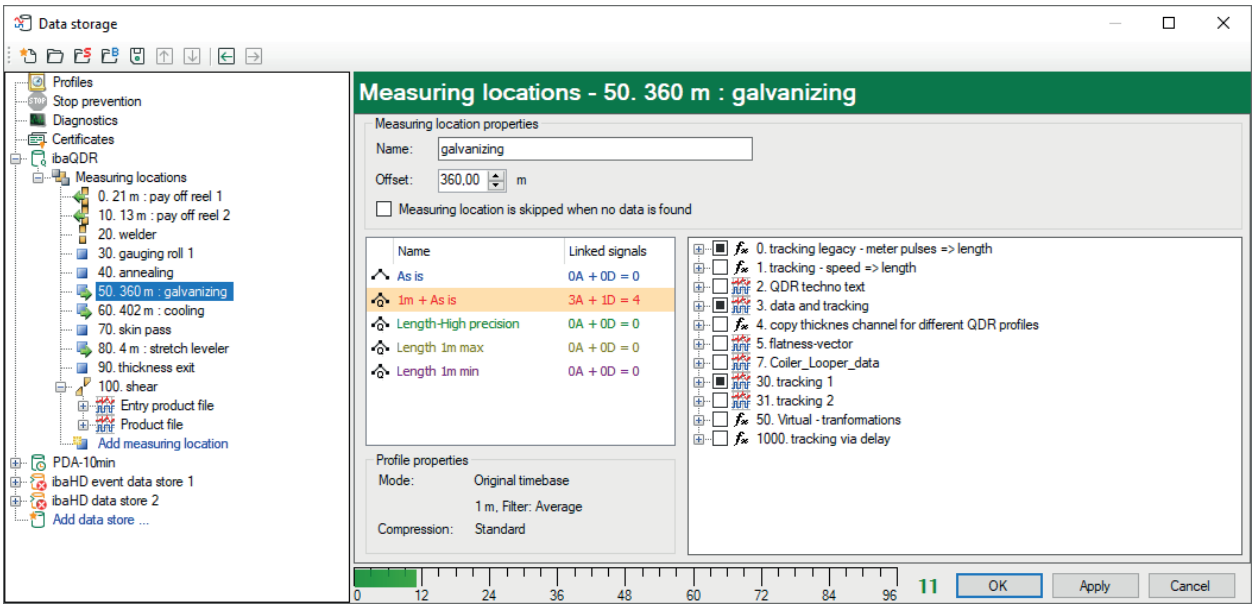
Multiple measuring locations with offset can refer to the same standard measuring location. In the data store configuration (tree) they are positioned under the node for the standard measuring location, sorted by the size of the offset.

These measuring locations do not support strip elongation. The *Tracking ID* and *Length* signals are created internally in *ibaQDR* by applying a length delay to the tracking signals from the reference standard measuring location. If the reference standard measuring location is disabled, the associated offset measuring locations are also disabled.

Measuring locations with offset are only possible in continuous strip lines, i.e. not in hot rolling mills, push-pull pickling lines, discontinuous tandem lines or inspection lines.

Typical applications

- Coating lines
- Annealing and pickling lines



Measuring location properties

Name as for a standard measuring location

Offset

Enter the distance to the standard measuring location in the base unit of length.

Maximum permitted value: 5000

Note that a point must be entered as the decimal separator.

The measuring location is skipped if no data is found

Enable this option, if measuring data may be missing temporarily while the plant and the ibaQDR system should run on. This can be the case, e.g. if machinery is temporarily taken out of operation for operational or maintenance purposes.

By enabling this option, the configuration of an "Enable" signal becomes obsolete in many cases

5.3.3 Measuring location in front of the tracking start

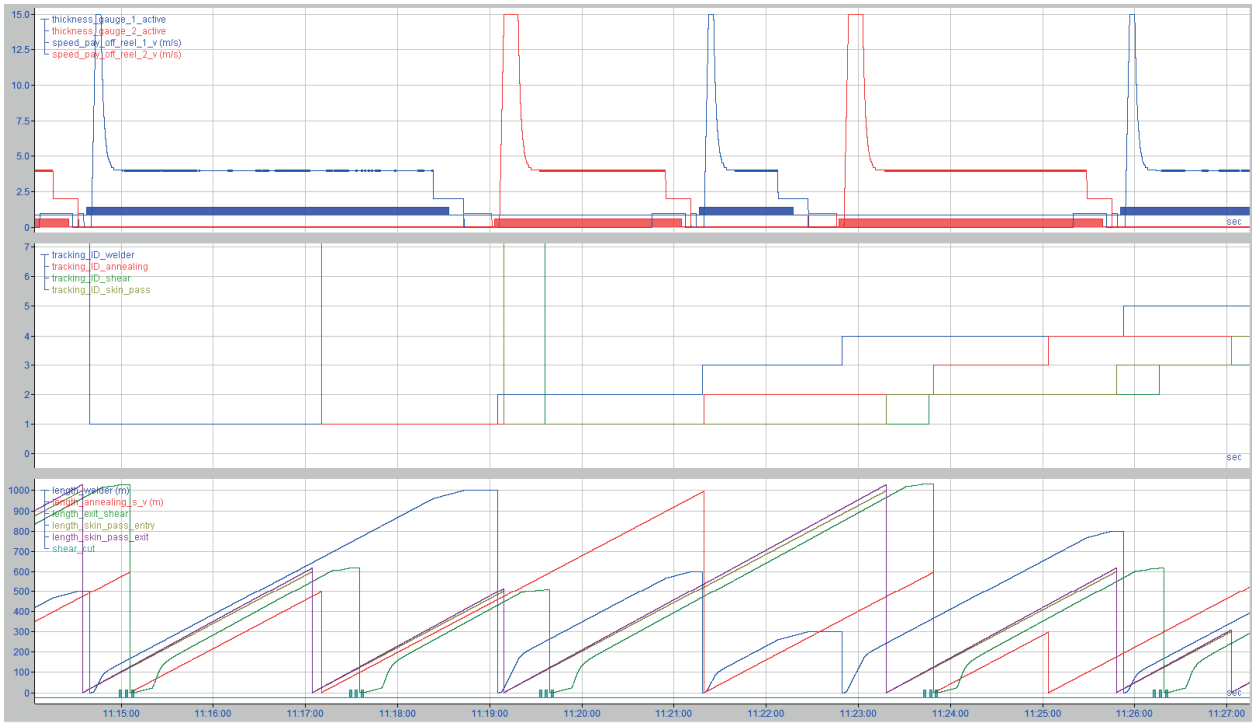
Description

A measuring location in front of the tracking start is required for measuring devices that are physically located before the first standard measuring location, at which material tracking begins.

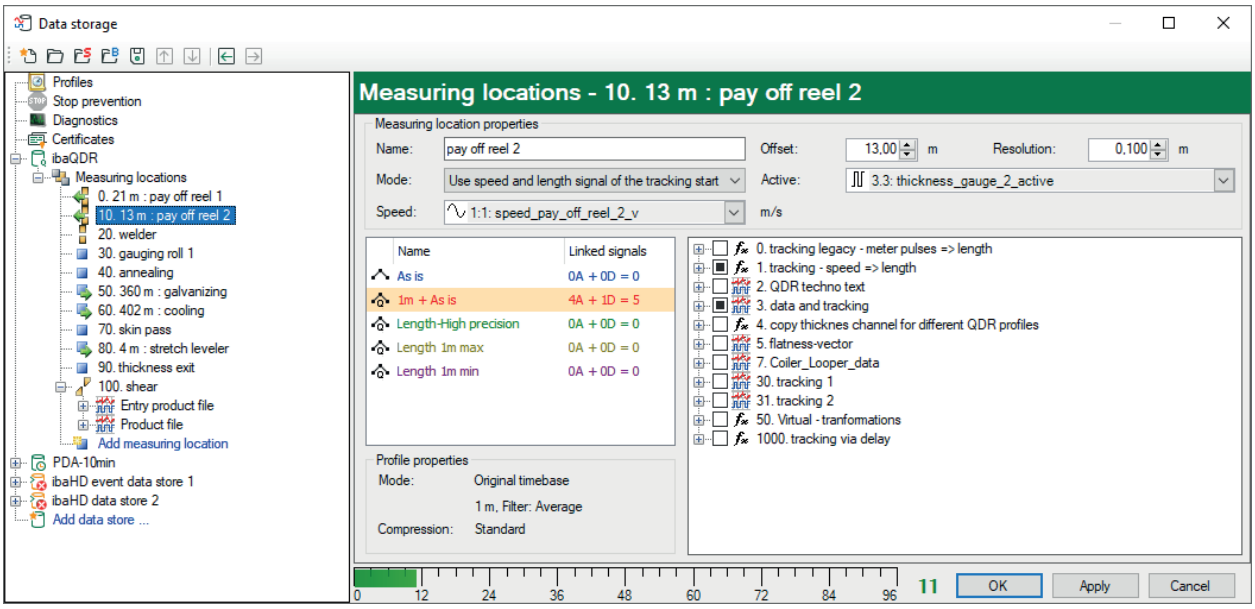
Typical application

- Uncoiler or thickness gauge before welder
- Measuring devices in an inspection line that are located before the last measuring location

Based on experience, a huge range of modes and operator interventions have to be mapped in the line section before the tracking start. As a result, the following procedure is adopted in *ibaQDR*. The measured data for the measuring location along with the speed are initially stored in an internal buffer. As soon as the measuring location is activated (e.g. strip from the relevant uncoiler path is welded or tack welded) the speed signal is used to convert the last data from the buffer for the offset section into a length and it is then treated as the first part of the measurement signal. The length tracking at the tracking start measuring location can then be used to carry out the subsequent length mapping. In the following example, the first graph shows the speeds and active signals for the two thickness measurements on the relevant uncoiler paths:



Associated settings in *ibaQDR*:



Measuring location properties

Name, Active as for a standard measuring location

Offset

The offset (material travel in base unit of length) here is the distance between the relevant measuring device and the first (standard) measuring location, at which tracking of the material begins.

Resolution

Length resolution of the data at the measuring location. This should be lower than the base length resolution for *ibaQDR*.

Active

The signal must be active (true) at precisely the time when the section (uncoiler path) is connected and thus the speed is the same as at the tracking start measuring location. This is normally the case from the precise moment of welding (tack welding) until the end of the strip has reached the welder or tacking machine.

Mode

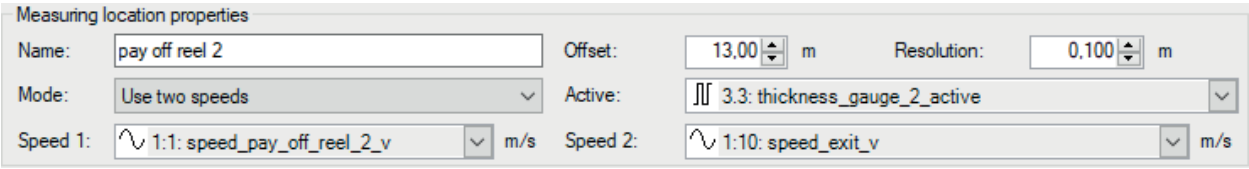
You can select one of the following modes:

<u>Use speed and length signal from tracking start</u>

This mode (default) is intended for processing lines with multiple uncoilers. It enables measuring locations before the welder or tacking machine to be correctly mapped.

Use two speeds

If you select this mode, on inspection lines with multiple shears, you have the option of switching between the uncoiler and coiler speed, for example, thus ensuring full acquisition of the data.



Speed
Select a speed signal here for the material speed before the tracking section. For example, this can be calculated from the uncoiler speed and the coil diameter.

Use two speeds and length signal from tracking start

Please contact iba AG for further information about this mode.

> **Example for mode with one speed**
> In continuous strip processing lines, tracking normally begins at the welder. The start of the strip, and thus length 0 for the new “tracked part”, is not clearly defined until the cleanly cut head of the strip is welded to the tail of the preceding strip and the weld seam has been marked by hole punching or notching.
>
> However, there is usually a thickness gauge in the infeed section of the line several meters before the welder to measure the material thickness before the strip is welded. Strip sections with excess or insufficient thickness can then be removed by scrap cuts.

5.3.4 Reversing measuring location

Description

A reversing measuring location is a measuring location that the material passes through completely multiple times, each time in the opposite direction. This means that reversing partial sections of a product cannot currently be mapped with *ibaQDR*.

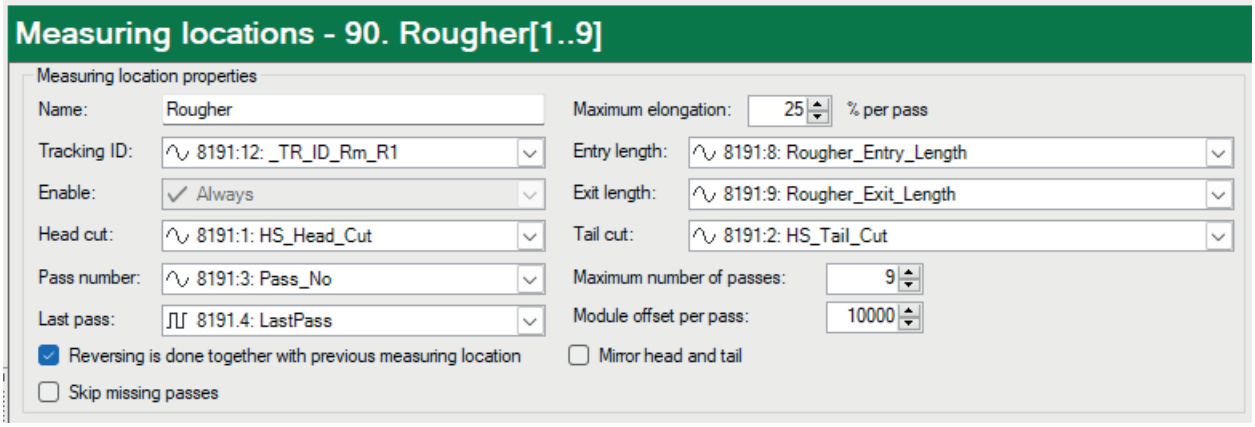
Internally, each pass is stored separately as though there were multiple measuring locations passed through in turn. Therefore for licensing purposes, the configured maximum number of passes is taken into account in the number of measuring locations.

To correctly map the data, the head and tail of the length-based measuring series are reversed at every second pass (even pass numbers).

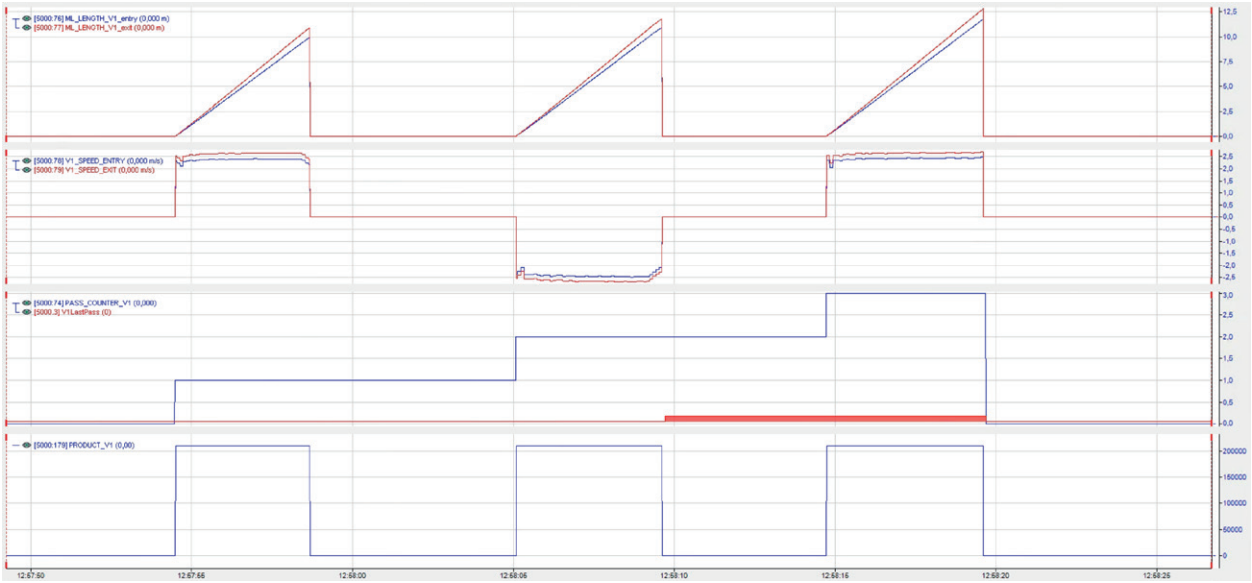
Typical applications

- Reversing stand (with material elongation)
- Edger-rougher combination (with material eleongation)
- Coil box (without material elongation)

Example of a roughing stand in a hot rolling mill (actual data):



The following figure shows the associated signal progression for the three passes:



Measuring location properties

Name, *Tracking ID*, *Enable*, *Maximum elongation, Entry and Exit length* like for standard measuring location.

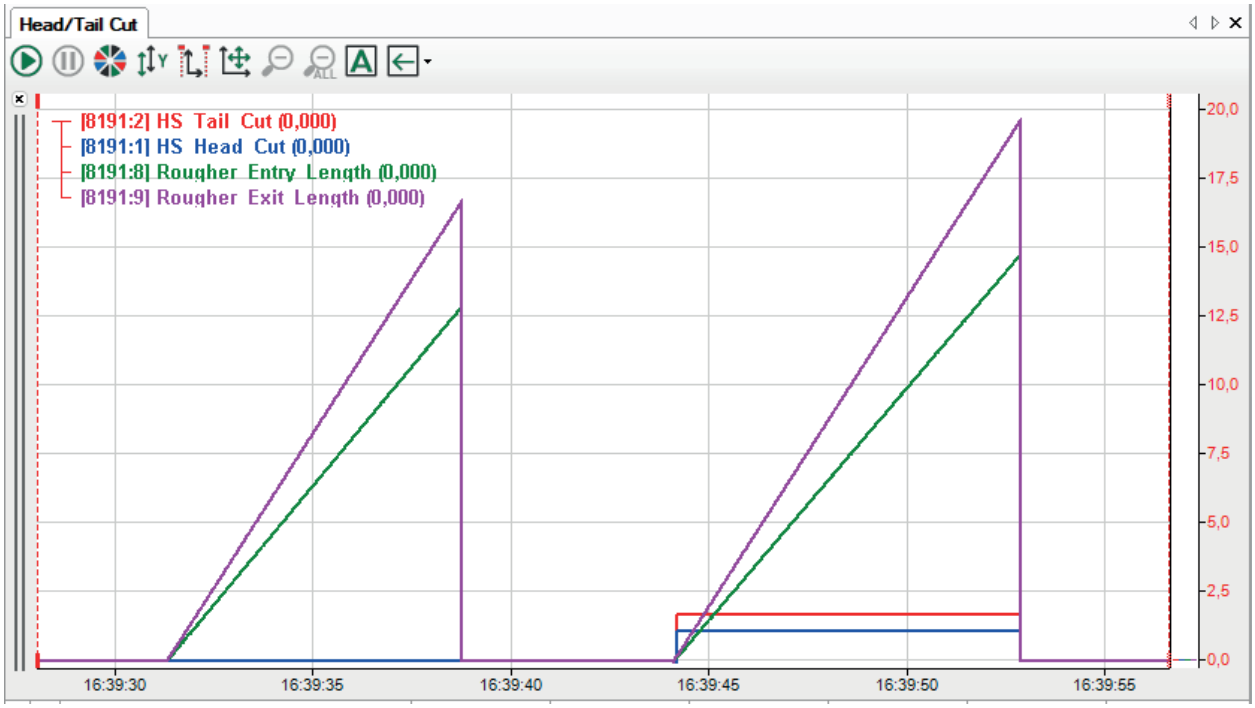
The value for *Maximum elongation* applies to each pass and is not a total over all passes.

Head cut/Tail cut

In case of one or more crop cuts at strip head (head cut) or strip end (tail cut) after this measuring location, the length value of the strip needs to be corrected by the cut length. This ensures the correct length-related mapping of the measured values for this and the following measuring locations.

Select here the signals which transmit the total cut length of head and tail cuts. If there is no shear behind this measuring location, select "No cut".

The following figure shows an example of a rougher mill with a head cut and a tail cut between the first and second pass. The red and the blue curve show the cut lengths. Consequently, the entry length (green curve) of the second pass equals the exit length (magenta curve) of the first pass reduced by the cut lengths.



Pass number

Select an analog signal that counts the passes. The signal should be provided by the line control or material tracking system.

If no material is present, the value should be 0. For the first pass the value should be 1, for second pass 2, etc.

Last pass

Here, you can select a digital signal that indicates when the last pass is running. This setting is optional. You only need the signal if a product file for the tracked part (e.g. entry coil) is to be generated at the reversing measuring location.

If the signal is to be used, it must be log. 1 (TRUE) for the entire duration of the last pass.

Maximum number of passes
Set the maximum number of passes *n* to be completed on the line. This value *n* specifies how many instances of the measuring location are created in the product file. At each pass, the signal values are saved to an instance. Each of these instances is automatically given a designation containing the pass number. The instance *n* always corresponds to the last pass, instance *n-1* the penultimate pass, etc.

If more passes are made than the maximum number *n* set here, only the data for the last *n* passes is saved in the product file.

If fewer passes than the set maximum number *n* are made, only the last passes or the last instances are populated and the first passes contain no data.

Note

Please note in this context the setting for *Pass numbering* in the basic QDR data store settings.

↗ *Basic settings for the data store, page 39*

Module offset per pass
As the signals assigned to the reversing measuring location are added to the product file at each pass, they cannot retain their original signal ID. Otherwise there would be multiple signals with the same signal ID in the product file, which is not permitted.

Therefore we have this *Module offset per pass* setting, which calculates a new module number for each pass instead of the original module number. The offset is calculated by multiplying the value set here by the pass number.

The signal IDs of the signals in the first instance (Pass 1) would have a module number equal to the Module offset per pass setting. In the second instance (Pass 2), the signals would have a module number 2 x the Module offset per pass etc. up to n x Module offset per pass.

Using example of default value (10000):

Module offset per pass	Pass number	Module number in signal ID
10000	1	10000
10000	2	20000
10000	3	30000
10000	4	40000
10000	5	50000

Select the offset value to ensure that there is no clash with other module numbers in the system!

Reversing is done together with previous measuring location
This option should be enabled for successive reversing measuring locations starting from the second measuring location. Normally, *ibaQDR* expects the entry length to correspond to the exit length from the previous pass. However, if there are multiple reversing measuring locations in succession, the entry length should be compared with the exit length of the previous measuring location.

Note

If the entry length does not match the associated preceding exit length, the previous data is “stretched” linearly and a message is posted in the event log.

A typical case for enabling this option is a combination of edger and roughing mill. These mills are usually located next to each other and the material passes through both mills in reversing operation. In this case you should enable this option for the measuring location of the roughing mill (which comes after the edger in terms of material flow).

Mirror head and tail

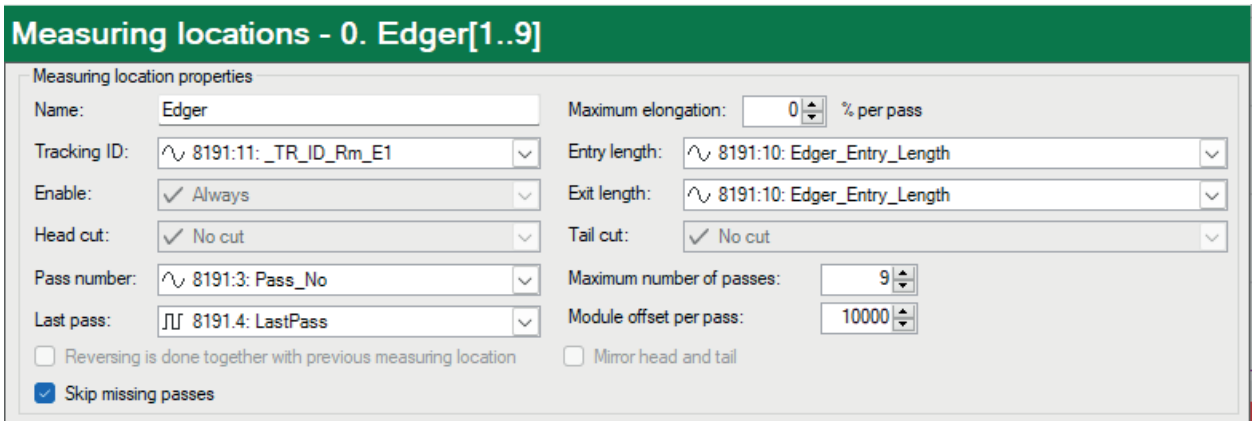
Enable this option for a measuring location if the production direction has reversed compared to the previous measuring location. For example, this is the case in beam rolling mills after lateral transportation of the material to the next roller table, where production runs in the opposite direction.

Skip missing passes

This option was created for coupled reversing measuring locations, e.g. as for combinations of edger and roughing mill. In reversing operation it may happen that the first part (edger) is not used every time. This means that the roughing mill can do more passes than the edger.

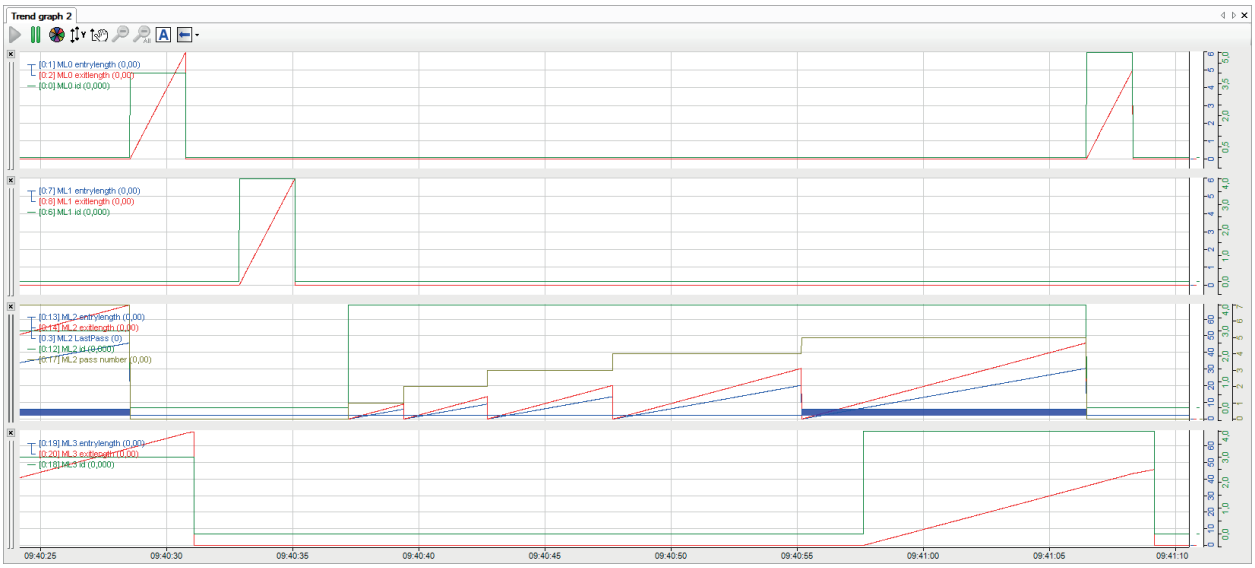
If you enable this option for the measuring location of the edger, the "missing" passes compared to the roughing mill will be taken into account.

The following figure shows the setting for the measuring location of the edger.



Example (theoretical)

The following figure shows the trend graphs for four measuring locations. The first two measuring locations are standard measuring locations, the third is a reversing measuring location and the fourth is a standard measuring location again.



We can see the length values (red and blue), the ID (green), the pass number (brown) and the digital “Last pass” signal (blue).

After each pass, the length signals are reset to zero on the entry and exit side. The tracking ID and pass number values can remain the same after a pass (as in the example) or be set to zero. As long as the tracking ID and pass number are not zero, time-based measured data is stored for the relevant pass.

Please note that the pass numbering depends on the setting in the QDR data store configuration, see also ↗ *Basic settings for the data store, page 39*.

5.3.5 Dividing measuring location

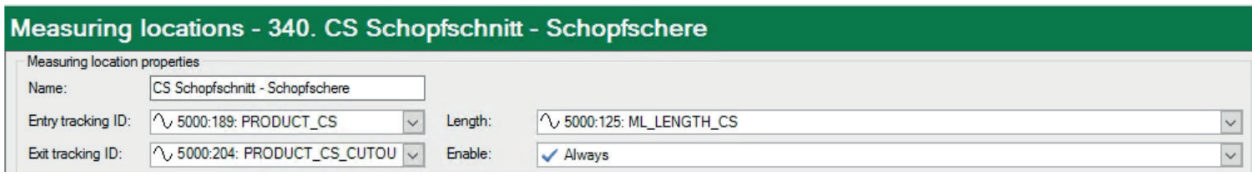
Description

A dividing measuring location is used to map division cuts, scrap or sample sheet cuts.

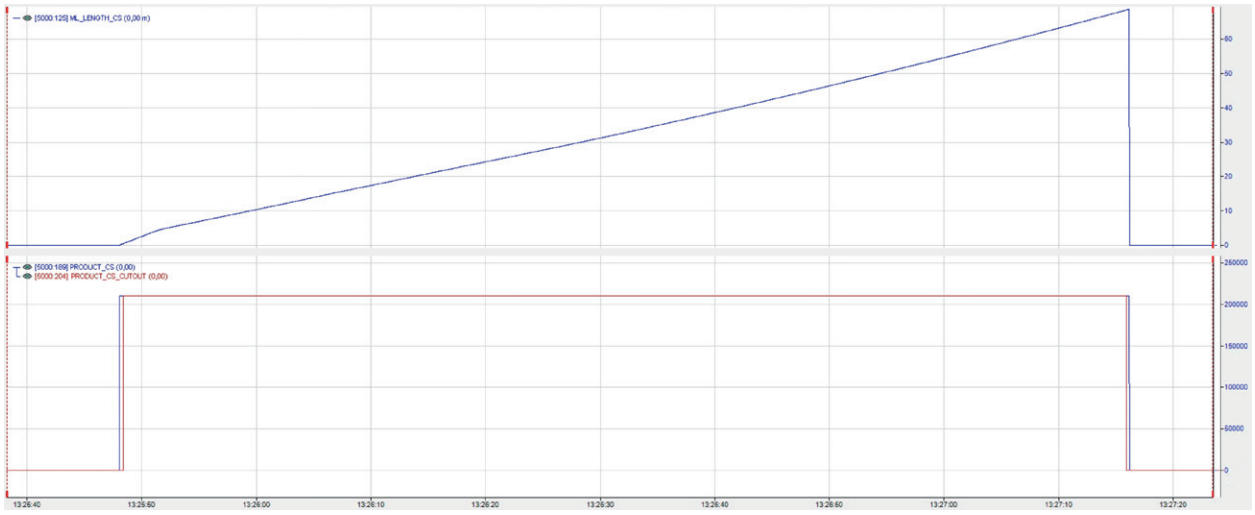
Typical applications

- Hot rolling mills
- Inspection lines

Example for cropping shear in hot rolling mill (actual data):



The following figure shows the associated signal progression for length (top) and tracking IDs (bottom) for two scrap cuts at the beginning and end of the product.



Lower trend view: Tracking ID of the original entry product (blue) and tracking ID of the cut product (red)

Measuring location properties

Name and *Enable* as for standard measuring location

Entry/Exit tracking ID

For the *Entry tracking ID* and *Exit tracking ID*, the material tracking system has to supply two different analog signals, which are then selected in the measuring location properties. Controlling the *Entry tracking ID* and *Exit tracking ID* enables two modes to be differentiated.

Scrap cut mode (material exits the line with no further processing)

If the *Exit tracking ID* is equal to zero and the *Entry tracking ID* is not equal to zero, this corresponds to a scrap cut. This means that the cut material at this measuring location is removed from production and undergoes no further processing.

Division cut mode (material remains in the line)

A cut should change the *Exit tracking ID*, so that the cut material receives a new ID. The *Entry tracking ID* remains unchanged.

The “tracked parts” resulting from the cut then pass through the remaining measuring locations after the dividing measuring location. When the product file for each of these new products is generated at the end, *ibaQDR* takes each corresponding section of the original “tracked part” at the dividing measuring location and adds the data from all preceding measuring locations.

When a product file is generated at the end, *ibaQDR* identifies the scrap sections and only takes into account the section of the “tracked part” remaining in the line.

Length

There is only one length signal as no elongation of the material takes place at a dividing measuring location. In contrast to a standard measuring location, the length signal corresponds to the entry length, which means that it has to be reset to zero with every cut.

Example (theoretical)

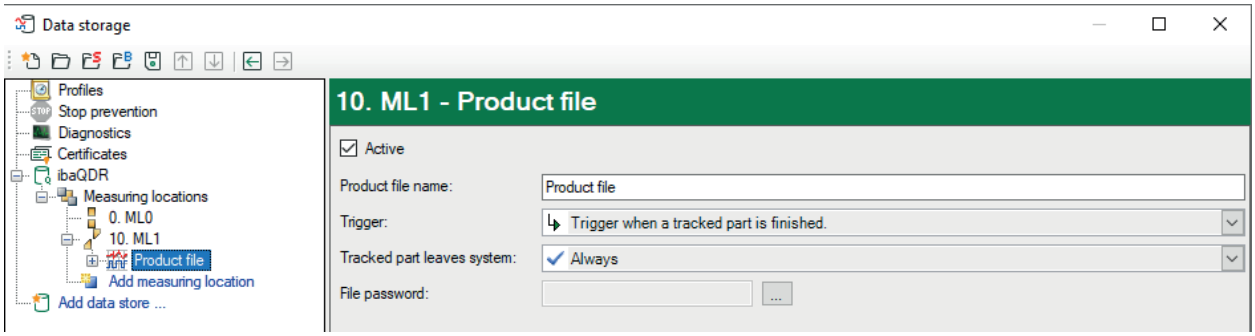
The following figure shows the time-based graphs for four measuring locations. The first two measuring locations are standard measuring locations, the third is a dividing measuring location and the fourth is a standard measuring location again.

Before the dividing measuring location, the tracked part has a length of 60 m (measured length value = red curve).

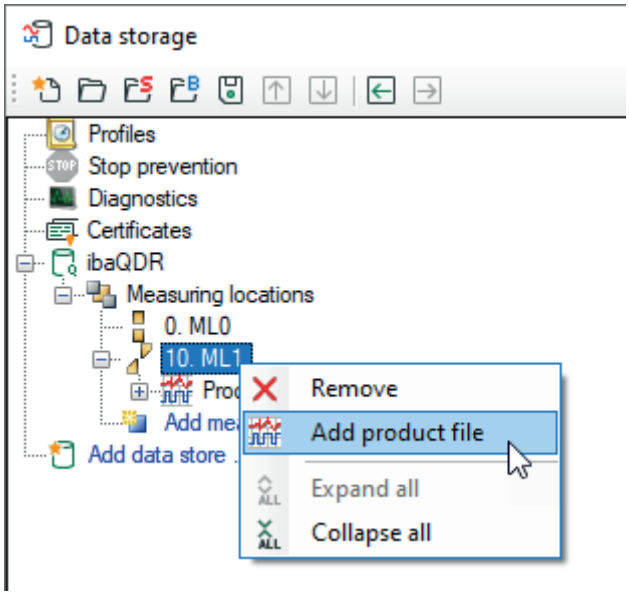
At the dividing measuring location, a division cut is made every 12 m. Each cut increments the *Exit tracking ID*. This produces five new tracked parts, each with a length of 12 m. These can be seen at the last measuring location, which they pass through in turn after the dividing measuring location.

5.4 Generating the data file

ibaQDR generates a product file at the last measuring location. A node in the tree is already prepared for it in the *ibaQDR* data store configuration.



However, you can also right-click on a measuring location and select *Add product file* to generate a product file for that measuring location.

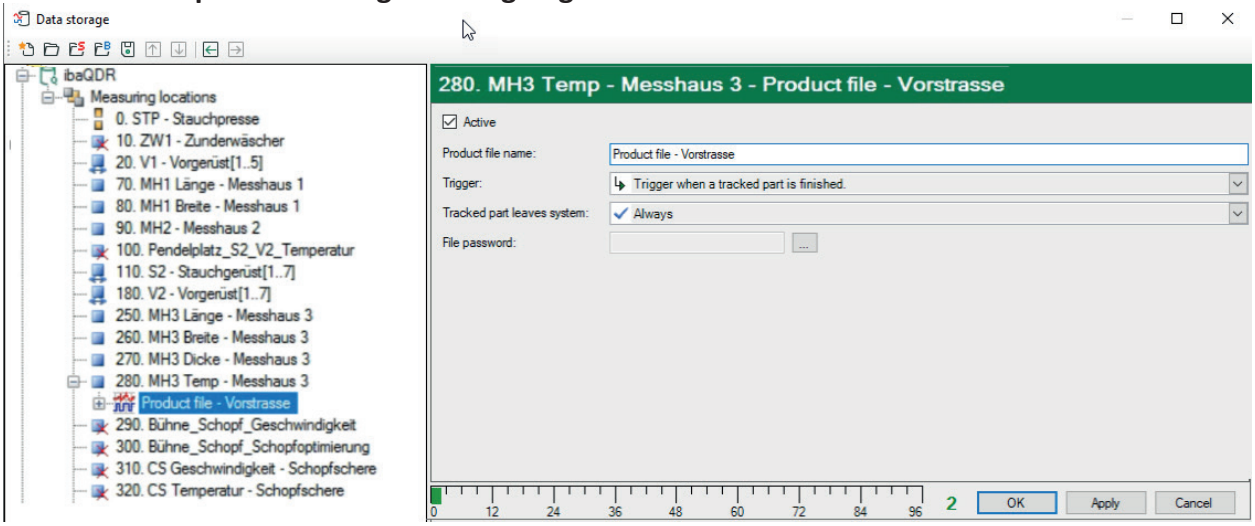


ibaQDR supports multiple product files. One or more product files can be added to a measuring location. This can be any measuring location, it does not have to be the last one. However, the last measuring location should contain at least one product file. Only measuring locations in front of the tracking start cannot contain product files.

There is a preconfigured default path for the base directory of the product files: `c:\iba\QDR\Product`.

If you use multiple QDR data stores then a new sub-folder will be created automatically for each additional QDR data store: `c:\iba\QDR2\Product, c:\iba\QDR3\Product` etc.

Practical example: Hot rolling mill roughing line

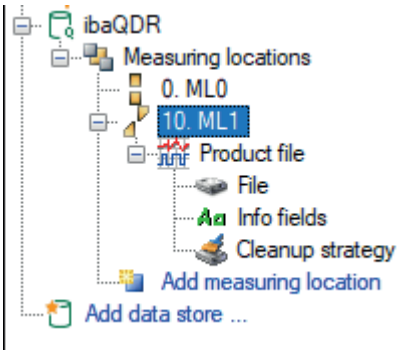


You can set a product file to active or inactive and you can give it a name. This name is used in the data store status window, in validation, and in the log file. This product file name is not the actual name that it will subsequently have in the file system.

You then have to configure the trigger signal that is used to create a product file. The trigger must be a digital signal (digital input signal, digital virtual signal or digital result of an expression). In this case, the product file contains the data between two rising edges of the trigger signal.

The trigger can also be the special signal “Trigger when a tracked part is finished”. In this case, the product file contains the data from the beginning of the entry product to its end.

You can protect the product file by entering a password in the *File password* field.



The product file node has the same sub-nodes as the normal, time-based *ibaPDA* data store.

You can use the *File* node to configure the file name and storage location.

You can use the *Info fields* node to configure the info fields.

The Cleanup strategy node allows you to configure cleaning up of the product files created.

The last product files are displayed in a separate node in the data store status window. The files can be opened in *ibaAnalyzer* by double-clicking or using the pop-up menu. The pop-up menu can also be used to open the file storage location in Windows Explorer.

Note on the trigger

Depending on the *ibaQDR* synchronization status, the trigger results in the following behaviors.

If a trigger occurs while *ibaQDR* is not fully synchronized, data will be missing at some measuring locations (at the start of the line). If data is missing, these measuring locations have no data in the product file.

The product file then contains the info field *$QDR_DataMissing* with the value 1.

If *ibaQDR* is not synchronized, the info field *$QDR_NotSynchronized* will be written to the product file with the value 1.

In the basic settings for the *ibaQDR* data store, you can set an option that *ibaQDR* will only generate product files when it is completely synchronized. See ↗ *Basic settings for the data store, page 39* for details.

In this case, all data at all measuring locations should be available (if the length signals are correct). *ibaQDR* is synchronized when a change of tracking ID (e.g. from ID 1 to ID 2) has been tracked from the tracking start measuring location to the last measuring location (e.g. exit shear).

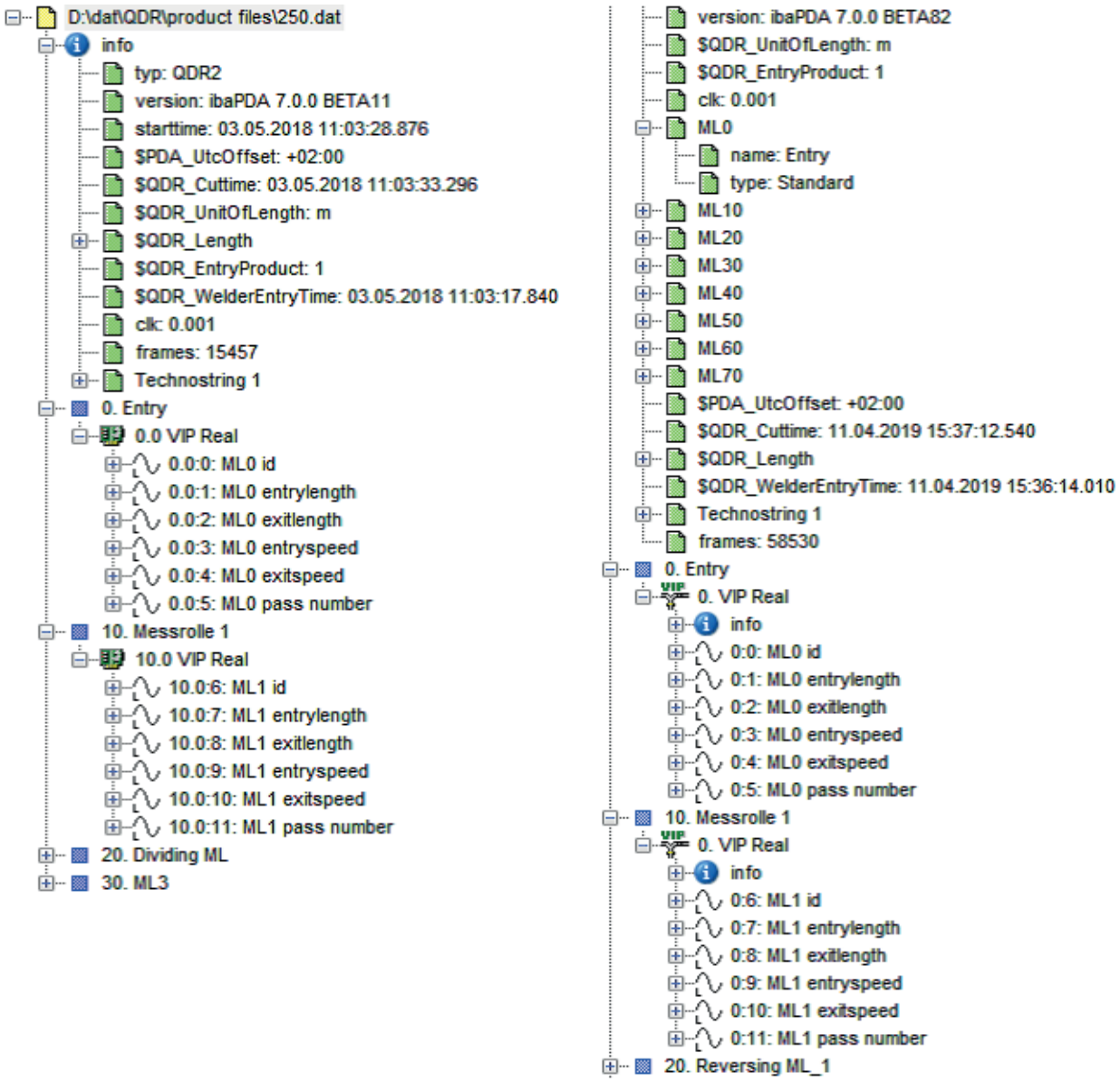
5.5 Comments on the data file format

In *ibaQDR* v7 a new file format has been introduced and there have also been some changes to the structure of the signals in the *ibaQDR* data files.

Signal ID

The most important change is that the measuring location number is no longer part of the signal ID. In *ibaQDR* v6 files a signal with the ID [0:0], belonging to the first measuring location (no. 0), was saved in the product file with the signal ID [0.0:0]. A signal with the ID [0:6], belonging to the second measuring location (no. 10), was saved in the product file with the signal ID [10.0:6].

From *ibaQDR* v7 the signals are only saved with their original ID. Thus, a signal with the ID [0:0] will always have the ID [0:0] regardless of which measuring location it belongs to. The advantage is that there are no longer any additional restrictions for the signal IDs in *ibaQDR* files. The following figures show a data file from *ibaQDR* v6 on the left and a data file from *ibaQDR* v7 on the right.



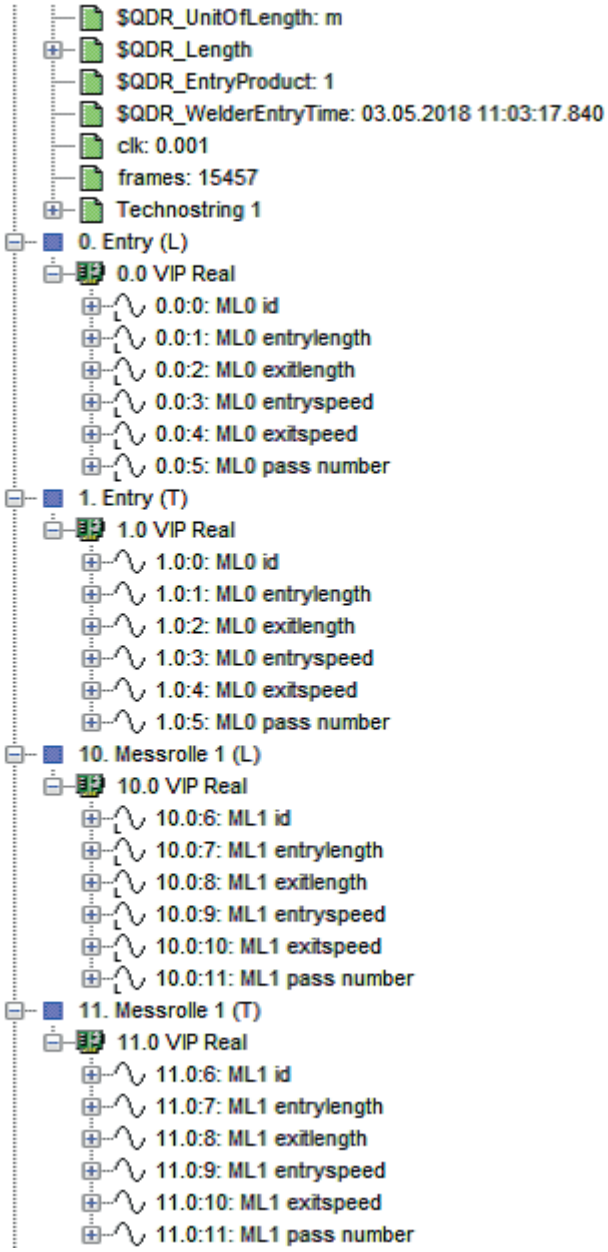
Saving time and length-based data

A further change is the way in which the time and length-based data for a signal is saved in the data file. In *ibaQDR* v6 the length and time data was saved in two separate signals.

The length signal had the measuring location number times 10 as the first number in the ID.

The time signal had the measuring location number times 10 plus 1 as the first number in the ID.

In *ibaAnalyzer* the length and time signal can be represented together, as in the previous figure, or they can be represented separately, as in the figure below. For example, for the signal [0:6] the length data was saved as the signal [10.0:6] and the time data as the signal [11.0:6].



From *ibaQDR* v7 the time and length data in a signal is saved as a single signal. The signal includes both time and length data. In *ibaAnalyzer* it behaves in exactly the same way as the *ibaQDR* v6 file with combined time and length signals. *ibaAnalyzer* has a compatibility function that allows the *ibaQDR* signals to be addressed using their old *ibaQDR* v6 ID. This function is useful when you are updating an *ibaQDR* v6 system to the latest version and want to continue using your old analyses (pdo files).

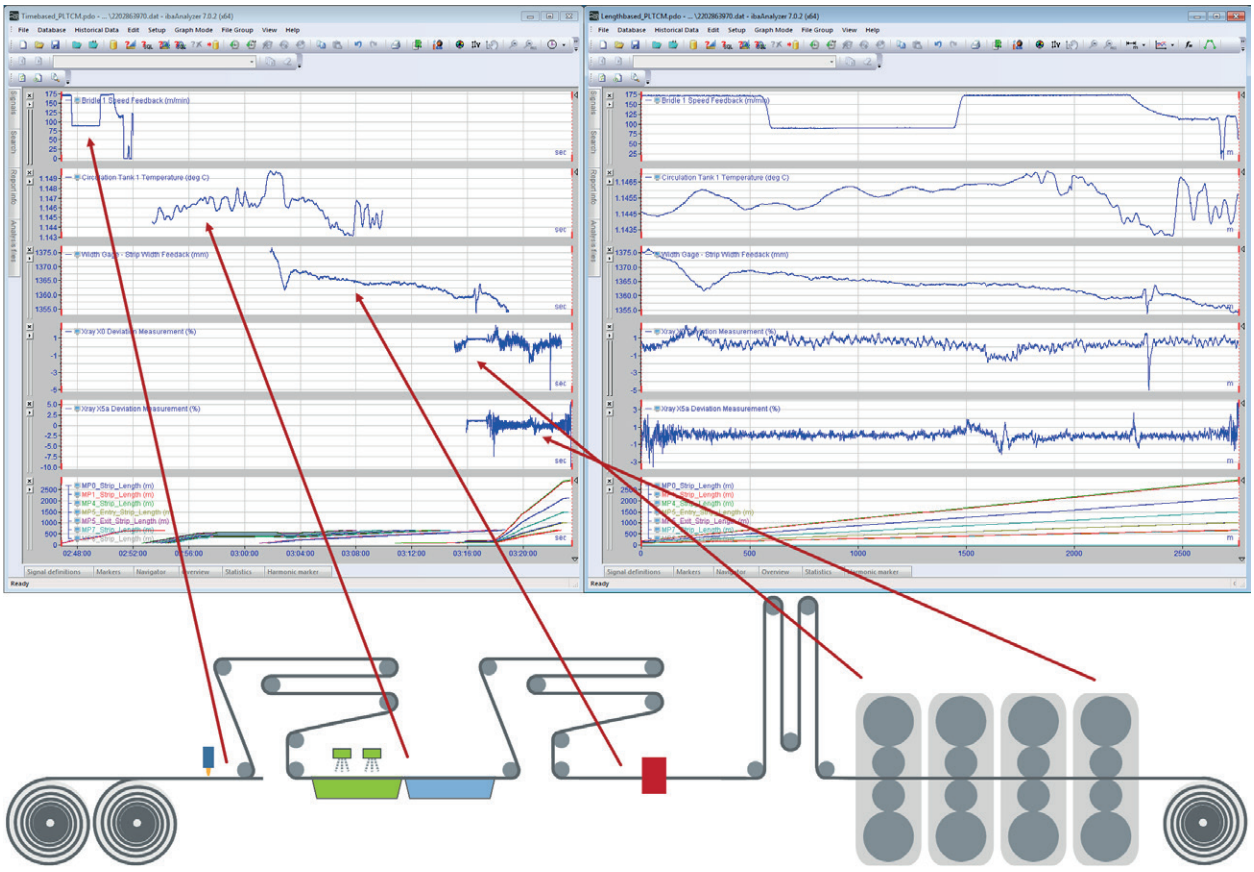
In *ibaAnalyzer*, you can address the signal [0:6] from the mapping as [0:6], [10.0:6], and [11.0:6]. This means that you can use the signal, e.g. in an expression, in all three familiar forms, with [11.0:6] supplying the time-based signal and [10.0:6] the length-based signal.

5.6 Length transformations in the data file

The length-based data in the product file is mapped to the length at the measuring location where the file was generated. Stretching is carried out dynamically and iteratively using the configured entry and exit lengths for the measuring locations, i.e. starting with the measuring station where the file was generated, *ibaQDR* successively looks at each preceding measuring location, stretches the previous data accordingly if necessary and continues to the first measuring location in the line.

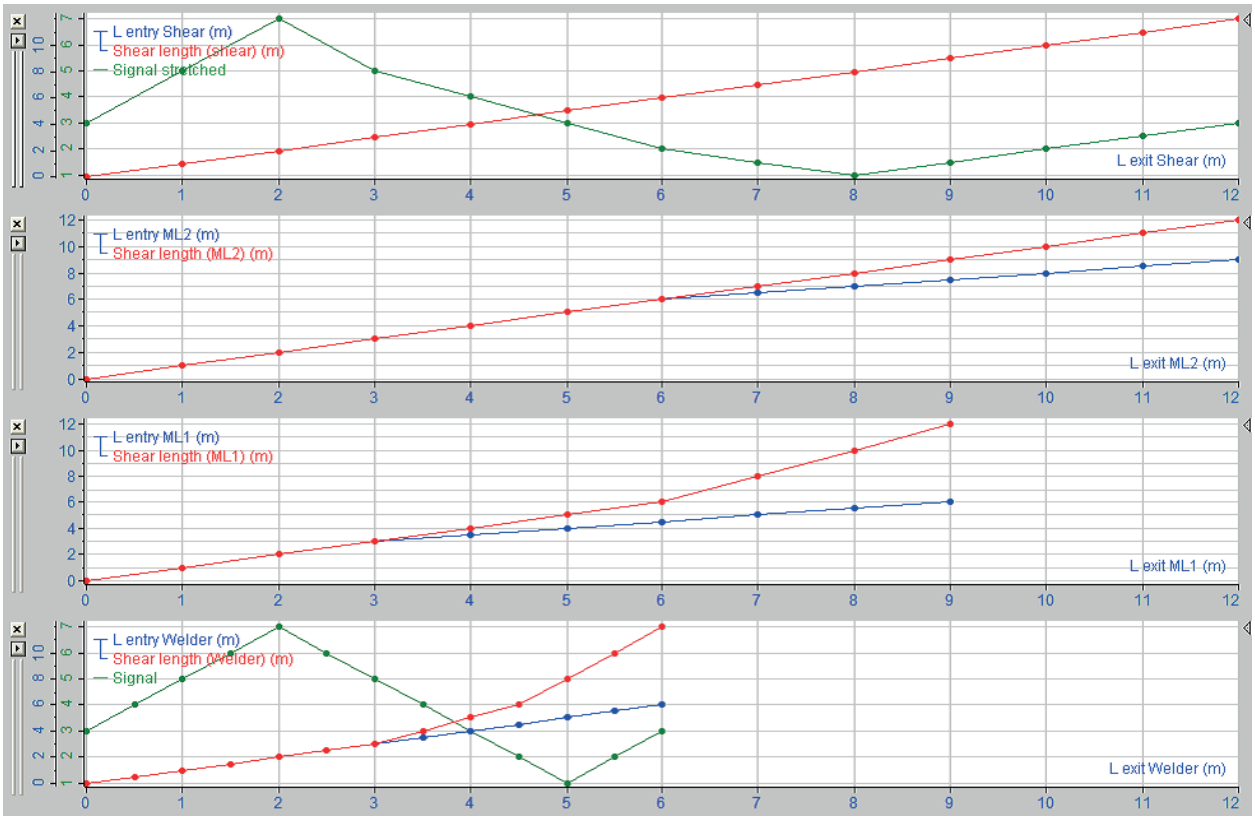
In the following real-world example of a pickling tandem line, the graphs show time-based signals for a product coil (*ibaQDR* DAT file) on the left-hand side and the length-based signals on the right-hand side. The first five graphs show signals from different measuring locations in the line (see reference arrows to line diagram), while the last graph shows the length signals from material tracking.

The progression of the length signals indicates that the line is run at different speeds and that a considerable elongation of the strip occurs in the rear tandem section. An entry product with a length of approx. 600 m is turned into a finished product 3000 m in length at the end, and all measured values are precisely mapped to the finished product length.



This length-based representation for the finished product is exactly what is required for a final quality assessment. It is important to determine which values were measured at which position on the finished product, for example to specify which strip sections have to be cut out due to tolerance violations during rework or to identify how many meters of the strip are outside the product specification and at which points.

The following graphs show a (theoretical) example of strip elongation.



The product coil is 12 m long. The first graph shows the length signals at the exit shear, plotted over the exit length (x axis).

The entry length (blue) and the length at the exit shear (red) are congruent. The red curve therefore covers the blue one. The green signal is any measurement signal acquired at the welder and shown stretched to the final length.

The second graph shows the situation at measuring location 2 (ML2). For the first 6 m there was no stretching at ML2. For the last 6 m, however, 100% stretching was identified. Therefore, at the entry to ML2 the coil was 9 m long and 12 m at the exit. The graph shows the entry length signal plotted against the exit length. The length signal is the same as at the shear as *ibaQDR* always uses the exit length to acquire the data at a measuring location.

The third graph shows the situation at measuring location ML1. The exit length here is 9 m. Creating an X/Y diagram of the shear length signal at ML2 and the entry length at ML2 gives the length at ML1. The length has a gradient of 1 for the first 6 m and then a gradient of 2 for the last 3 m. There is additional stretching at ML1. There is no stretching on the first 3 m, followed by stretching of 100 % on the last 6 m. You can see this from the ML1 entry length signal. The coil was 6 m long at the entry to ML1 and 9 m at the exit.

The fourth graph shows the situation at the welder. The exit length here is 6 m. The length signal at the shear is the result of an X/Y plot of the length signal at ML1 and the entry length at ML1. There was no stretching on the first 3 m, resulting in a gradient of 1. On the next 1.5 m there was 100% stretching at ML1, resulting in a gradient of 2. On the last 1.5 m there was 100% stretching at ML1 and 100% stretching at ML2, resulting in total stretching of 200% and a gradient of 4.

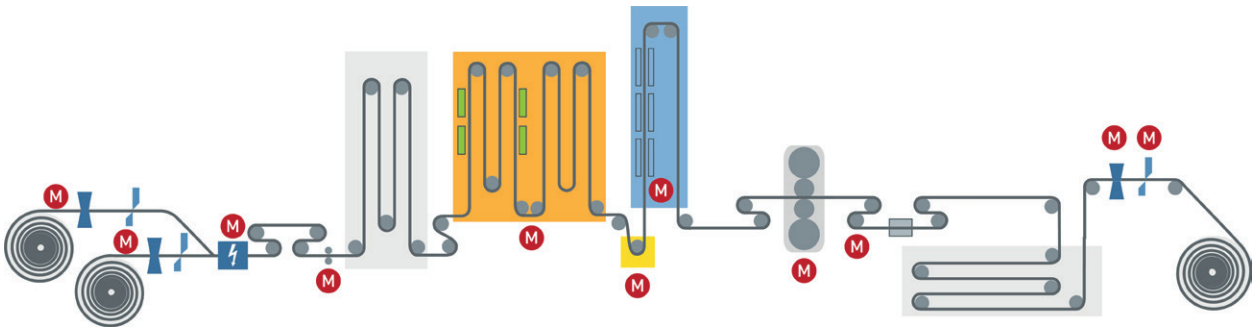
The green signal is a signal scanned at the welder. *ibaQDR* then uses the shear length signal at the welder to stretch the green signal to the length of the product coil. The result can be seen in the first graphic. *ibaQDR* expects the exit length from one measuring location to be equal to the entry length at the next measuring location. If it is not, *ibaQDR* stretches the signals uniformly. If the length difference is more than 0.5 %, an info field is written to the data file. The info field is called *$QDR_MismatchBetweenExitAndEntryLengthAtMlx* where x is the measuring location index. The value of the info field is the length difference in percent. If the difference is more than 10 %, an error is displayed in the event log.

6 Configuration for different line types

This section describes the most important configuration steps and settings for different line types. For the continuous strip processing line type, these are explained in more detail using the example of a simplified hot-dip galvanizing line. The key features and differences for the other line types are then explained.

6.1 Continuous processing lines

This section describes project planning using the example of a simplified hot-dip galvanizing line. The information also applies to other continuous processing lines, such as tin plating, chromium plating, plastic coating, annealing and pickling line, etc.



Key features

- Multiple line sections with different speeds (entry, processing, and exit sections)
- Multiple uncoilers and coilers possible
- A skin pass mill or stretch leveler are often included, where elongation of the material and thus a speed difference occurs.
- 1:n, n:1 and n:m production possible.
- Multiple products in the line simultaneously
- Scrap cuts, sample cuts

Typical measured values

Thickness, width, layer thickness, temperatures, strip tension, skin pass level, oil quantity

ibaQDR

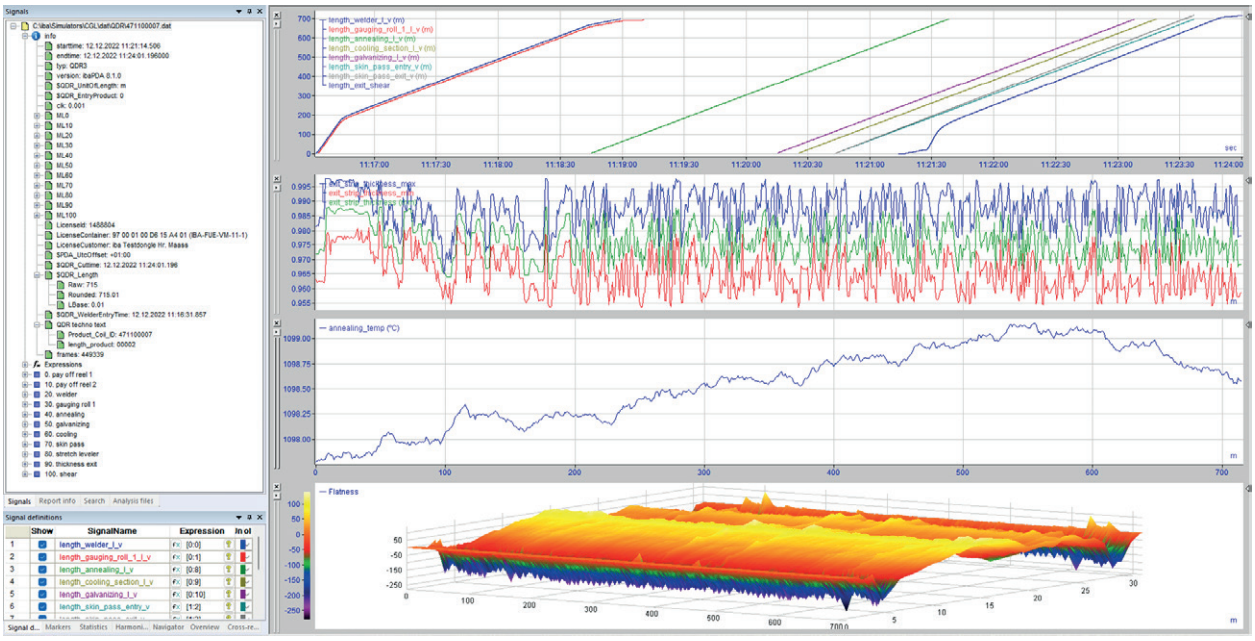
Property	Remark
Tracking ID	With each welding operation
Tracking start	Length counting begins at completion of welding operation
First measuring location	Uncoiler
“Finished” signal	Division cut at the shear before coiler
Length resolution	1 m

Typical measuring locations

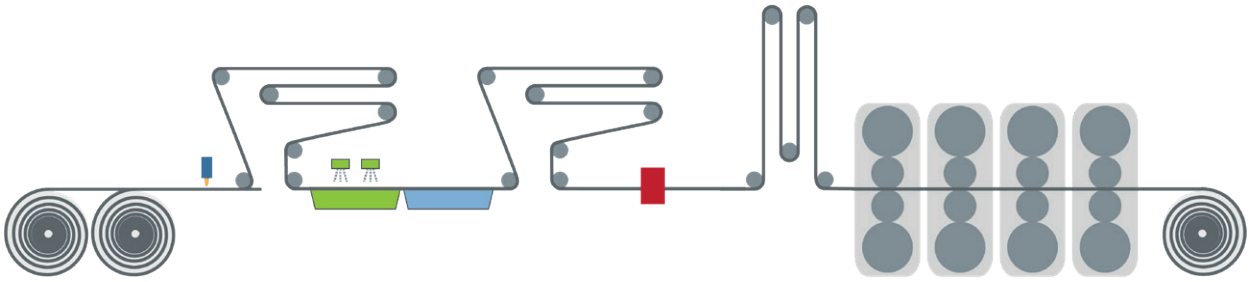
Measuring location type	Application / Special features
Tracking start (default)	Standard measuring location, which is provided and enabled by default. Normally corresponds to the welder.

Measuring location type	Application / Special features
Tracking end (default)	Standard measuring location, which is provided and enabled by default. Example: Exit shear
Standard measuring location	Distributed through the entire line, where material tracking delivers precise signals or length values. Use is essential at locations where material elongation occurs (entry length ≠ exit length) Example: Skin pass mill
Measuring location with offset after MLx	Distributed through the line, always with reference to an upstream standard measuring location. To be used wherever material tracking provides no signal or no length value. The exact distance (strip travel) to the standard measuring location must be known. Example: Shapemeter roll 3 m after skin pass mill
Measuring location in front of the tracking start	The exact distance (strip travel) to the tracking start measuring location must be known. Example: Thickness gauge before welder

The following figure shows the data from the CGL simulator in *ibaAnalyzer*. The top graph shows the length signals from material tracking against time. In the following two graphs, signals from different parts of the line are shown standardized to the finished strip length, with the last graph showing the signal vector for a flatness measurement. These visualizations are particularly useful for signals from traversing measurement devices, e.g. layer thickness or oil coverage measurements.



6.2 Coupled lines (pickling tandem)



Key features

- Combination of two kinds of lines: continuous processing line and tandem cold rolling mill, picking section operates continuously, tandem discontinuously
- The lines are connected to one another by a material looper (coupling looper).
- Increased requirements for tracking and ibaQDR synchronization if “uncoupled operation” is also possible.
- 1:n, n:1 and n:m production possible.
- Multiple products in the line simultaneously
- Scrap cuts, sample cuts
- Significant material elongation in tandem section

Typical measured values

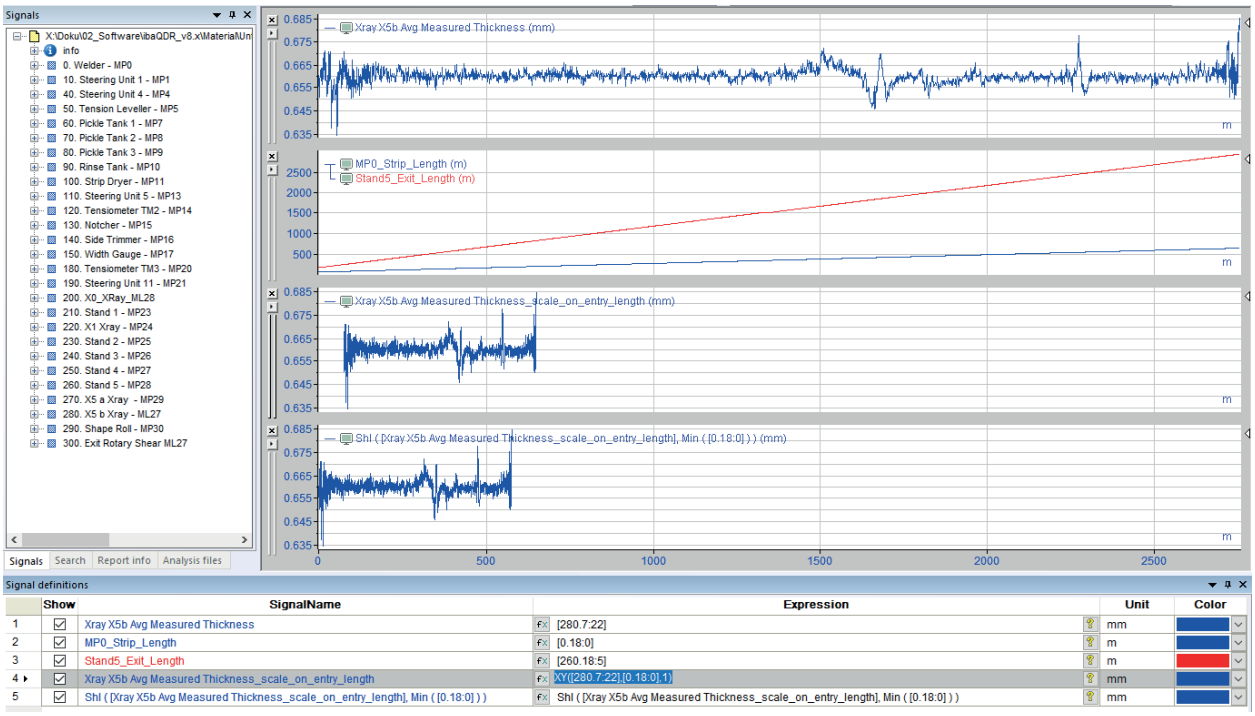
Thickness, width, bath temperature, concentration/pH value, strip tension, rolling forces

ibaQDR

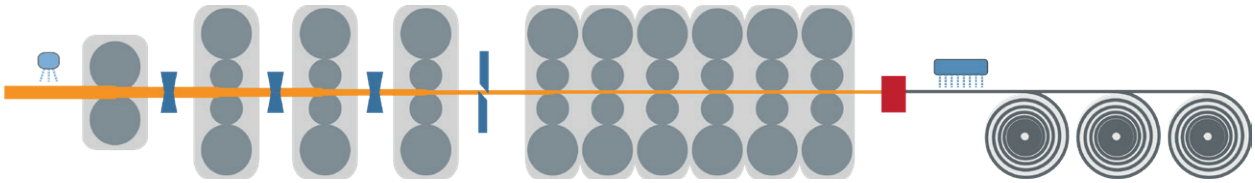
Property	Remark
Tracking ID	With each welding operation (entry to pickling)
Tracking start	Length counting begins at completion of welding operation
First measuring location	Uncoiler
“Finished” signal	Strip end detection at tandem coiler
Length resolution	1 m

Example:

In the following figure, the first graph shows the last thickness measurement signal from a pickling tandem line transformed to the finished strip length (standard behavior of *ibaQDR*). As the length signal from the measuring location at the tracking start is available (second graph), the XY function from *ibaAnalyzer* can be used to perform a transformation to the entry length (third graph). For the sake of completeness, in the last graph the signal has then been shifted to the left to the origin.



6.3 Hot rolling mill



Key features

- Very significant elongation of the material (up to 170 times the entry length)
- Very high number of signals
- Speed/length measurement of red-hot material not always very accurate
- Roughing stands with reversing mode
- When using a coil box, transposition of start and end of strip between roughing and finishing mill
- Scrap and, where necessary, division cuts in the line (cropping shear)
- Discharge of material from the line before the last measuring location is reached
- Different final measuring locations (coilers)
- Often non-homogeneous automation topology, i.e. an *ibaPDA/ibaQDR* system for the entire line can be very complex in terms of the I/O interfaces
- Significant performance requirements due to high line speeds and very high number of signals
- Virtualization may not be possible due to I/O requirements

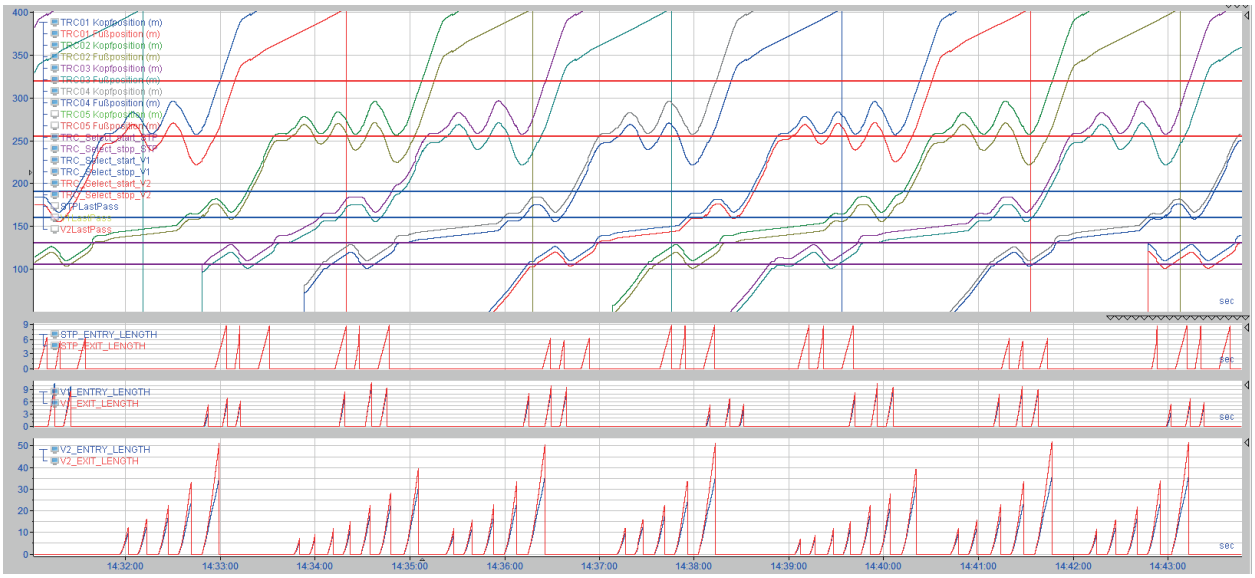
Typical measured values

Rolling forces, strip tension, width, thickness, material temperature, cooling water temperature

ibaQDR

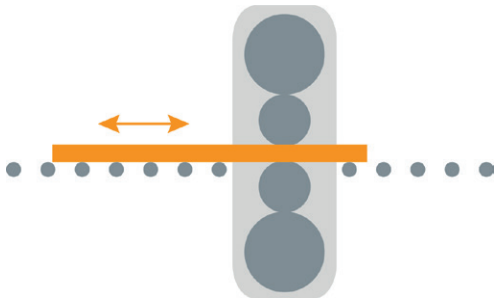
Property	Remark
Tracking ID	One ID per slab/rough strip/hot strip The ID signal must be set to 0 between the entry products.
Tracking start	Material detection at first measuring location, e.g. upsetting press
First measuring location	Identical to tracking start
“Finished” signal	End of strip detected at coiler or last measuring location before coilers
Length resolution	1 m

Example:
Only the head and foot position of each material on the roller table is supplied from the automation system. Therefore, there are two position signals for each of the numerous placeholders (e.g. slab 1-8). These can be seen in the first graph in the figure below. The position of the measuring locations on the roller table is known (see horizontal lines in first graph), which means that the length signals required for *ibaQDR* can be created - see second to fourth graph. Functions in the *ibaAnalyzer* expression builder have been used here to calculate the length signals.



6.4 Reversing stand

Roughing stand in hot rolling mill, skin pass mill / reversing stand in cold rolling mill



Key features

- Reversal of start and end of strip with each pass
- Material elongation
- On the reversing stand in a hot rolling mill, the material normally completely leaves the stand with each pass. By contrast, on a reversing stand in a cold rolling mill the material does not leave the stand, as it remains stretched on the coiler between the passes.

Note

In *ibaQDR* only the complete reversing is currently mapped, i.e. the material passes through the stand completely with each pass.

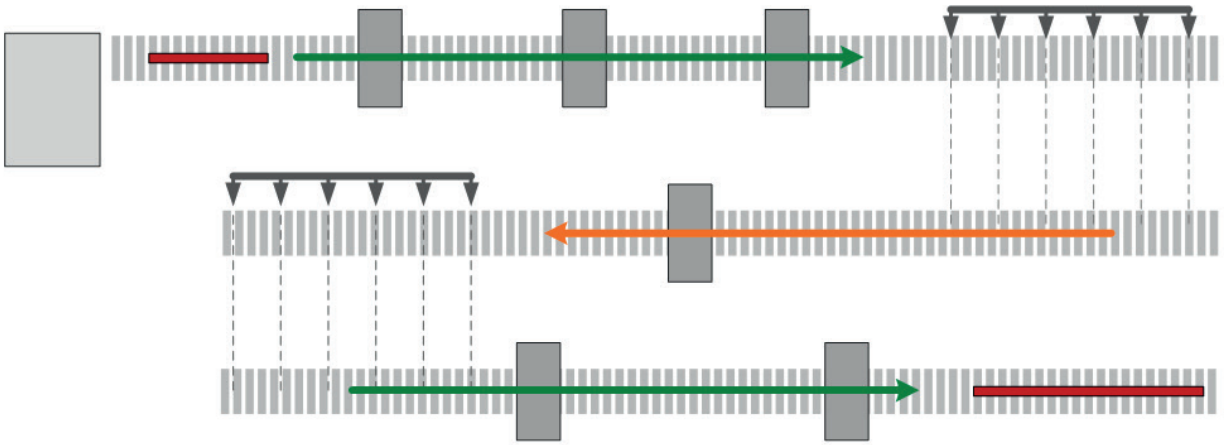
Typical measured values

Thickness, width, temperature, rolling force

ibaQDR

Property	Remark
Tracking ID	With initial pass
Tracking start	Material detection in stand
First measuring location	Identical to tracking start
“Finished” signal	Last pass + material leaves stand
Length resolution	1 m

6.5 Beam rolling mill



Key features

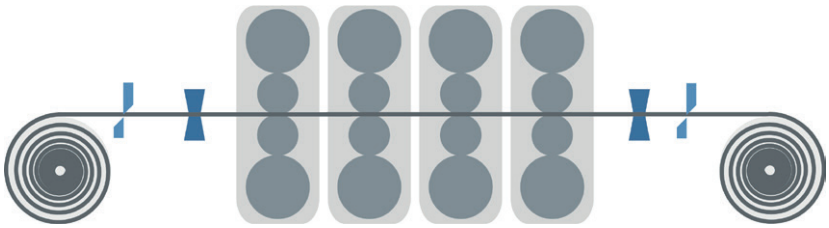
- Reversing function on individual stands
- Reversal of production direction from one roll stand to the next by lateral transportation
  Note that at each first measuring location after lateral transportation the *Mirror head and tail* option must be enabled.

Typical measured values
Width, thickness

ibaQDR

Property	Remark
Tracking ID	One ID per billet/beam The ID signal must be set to 0 between the entry products.
Tracking start	Material detection on first stand
First measuring location	Identical to tracking start
“Finished” signal	Product end detection at last measuring location
Length resolution	1 m

6.6 Tandem cold rolling mill (discontinuous)



Key features

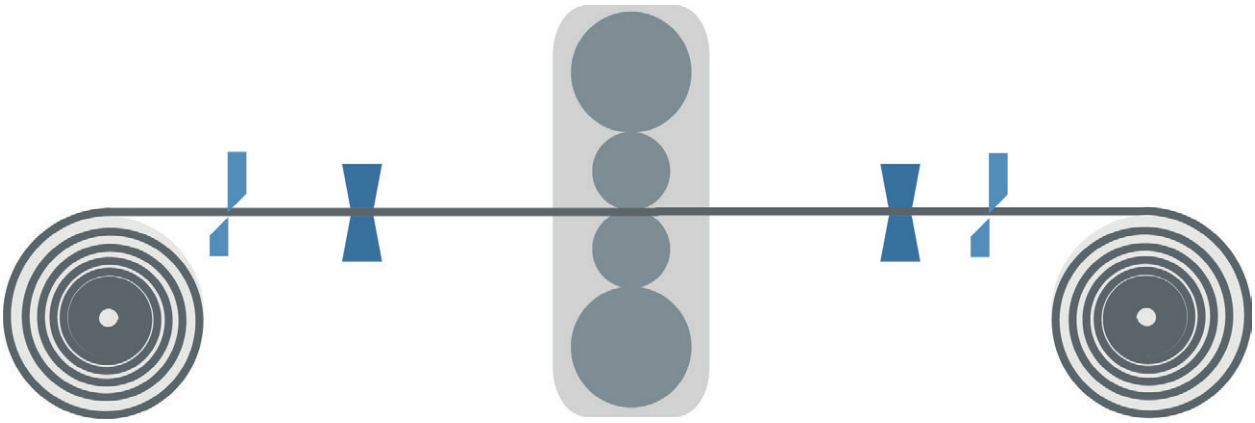
- Material elongation
- Only standard measuring locations possible

Typical measured values
Width, thickness, strip tension, rolling forces

ibaQDR

Property	Remark
Tracking ID	One ID per strip The ID signal must be set to 0 between the entry products.
Tracking start	Material detection at first measuring location before stands
First measuring location	Identical to tracking start
“Finished” signal	Strip end detection at coiler or shear cut
Length resolution	1 m

6.7 Skin pass mill



Key features

- Only standard measuring locations possible

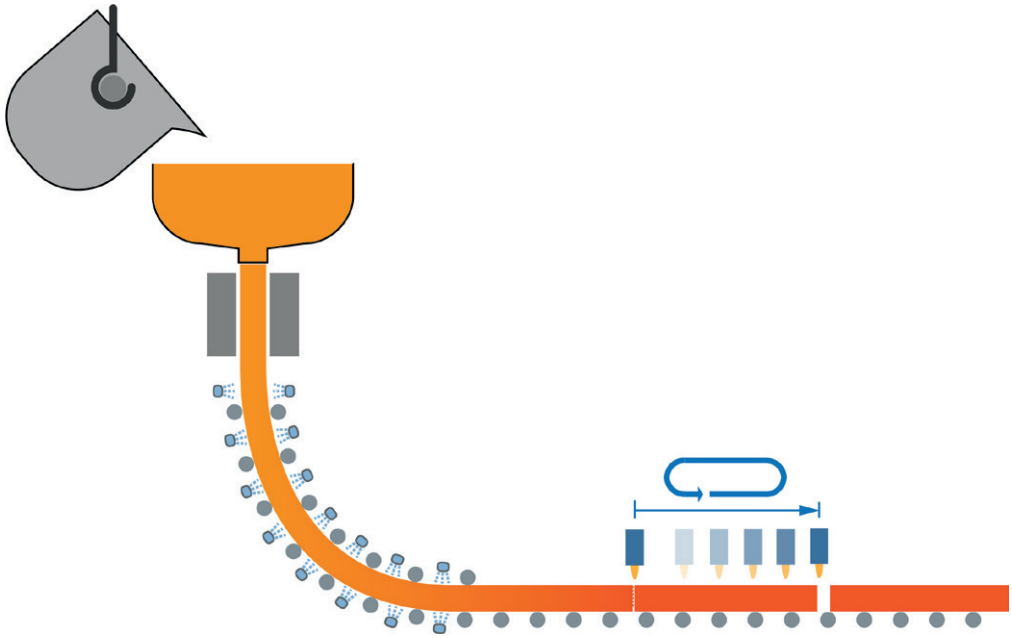
Typical measured values

Width, thickness, strip tension, rolling force, skin pass level

ibaQDR

Property	Remark
Tracking ID	One ID per strip The ID signal must be set to 0 between the entry products.
Tracking start	Material detection at first measuring location before stand
First measuring location	Identical to tracking start
“Finished” signal	Strip end detection at coiler or shear cut
Length resolution	1 m

6.8 Continuous casting lines



Key features
The particular challenge in continuous casting lines is mapping measured values from the liquid phase to the length of the strand and ultimately the finished slab.

In case of plants, which produce multiple strands in parallel, you can configure one QDR data store per strand. Each strand or slab finally gets its own product file. Hence, one *ibaQDR* system can cover multi-strand lines.

Note

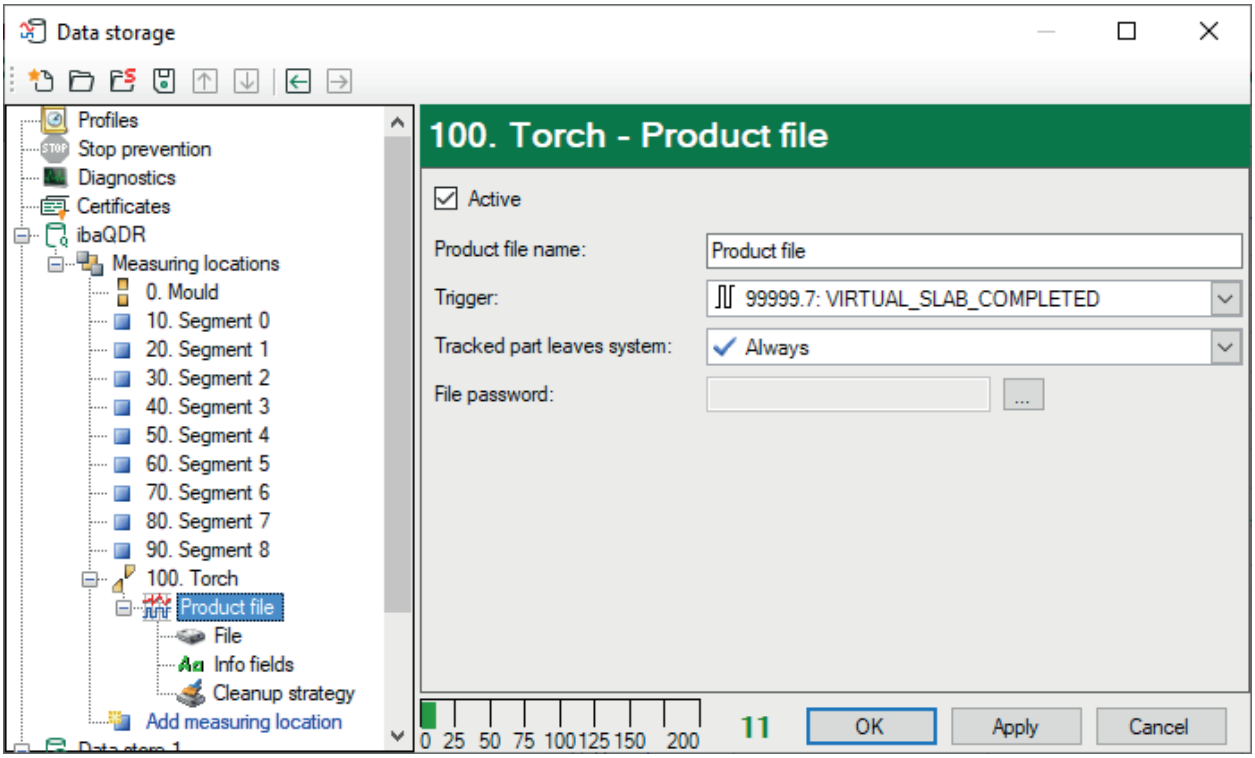
Please note that the number of measuring locations per strand is calculated from the total number of licensed measuring locations divided by the number of strands.

Typical measured values
Thickness, width, material temperature, cooling water temperature

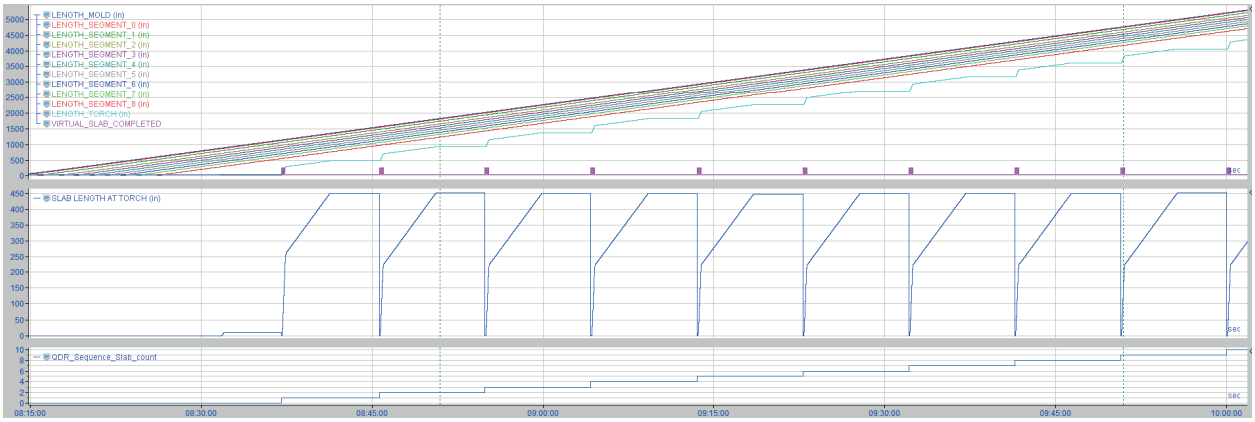
ibaQDR

Property	Remark
Tracking ID	ID = 1 during casting sequence, otherwise ID = 0
Tracking start	At start of casting sequence
First measuring location	Mold
“Finished” signal	Completion of flame cutting operation
Length resolution	cm/inch

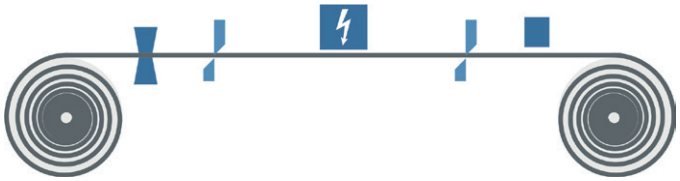
Example:
Example configuration for a continuous casting line with nine segments. The cut signal is created using a virtual signal at the completion of the flame cutting operation.



The tracking signals from the casting sequence show the movement of the flame cutting machine and the cut signal for generating the *ibaQDR* measurement file.



6.9 Inspection line



Key features

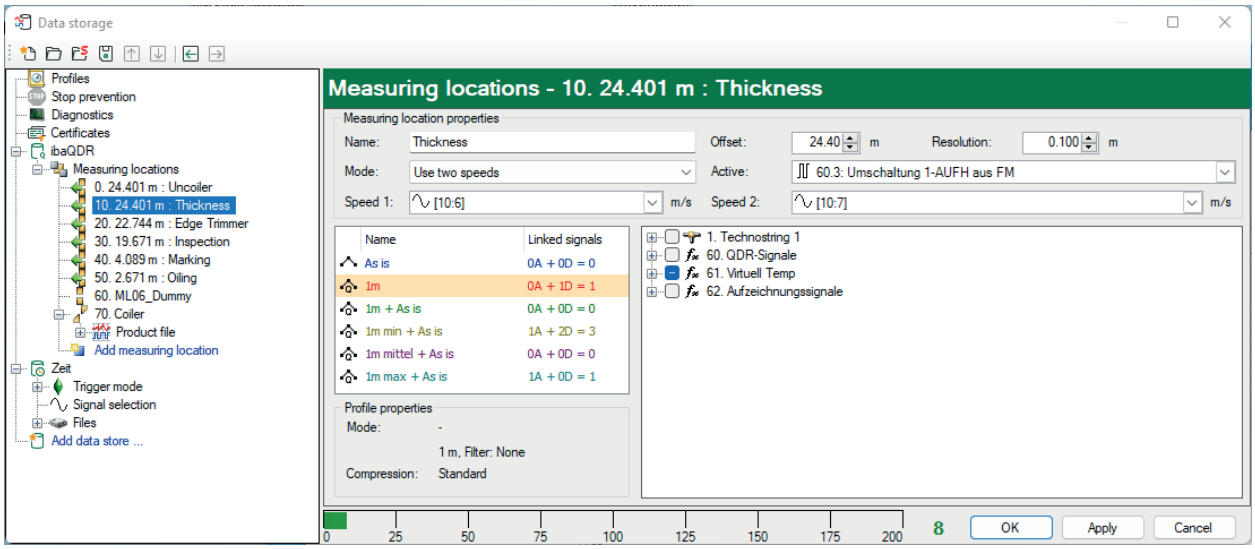
- No strict line operation, frequent changes of direction (forwards/backwards)
- Frequently: Division cuts, discharge of material and welding operations

Typical measured values
Width, thickness, strip tension, oil coverage

ibaQDR

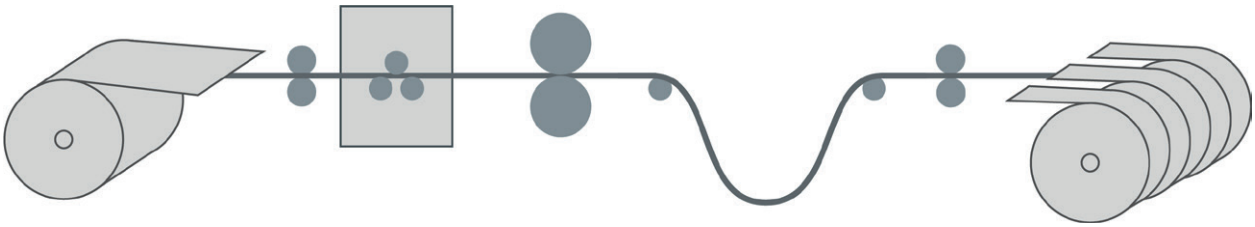
Property	Remark
Tracking ID	One ID per strip The ID signal must be set to 0 between the entry products.
Tracking start	Material detection at first measuring location
First measuring location	Identical to tracking start
“Finished” signal	Strip detection at coiler or shear cut
Length resolution	1 m

Example:
In this inspection line, as can be seen from the top diagram, there are two shears and thus two speeds to be taken into account. The coiler has been defined as the standard measuring location, with the tracking start measuring location before it as a dummy. The other measuring locations have been defined as “in front of the tracking start” type and the mode for two speeds has been set.



6.10 Slitting line

In slitting lines, a wider coil of metal sheet is unwound and cut lengthwise into several strips, usually of equal width. These strips are wound up again in parallel under constant tension and with straight edges. Usually, the coils are processed coil by coil and not from a sequence of welded coils. Each coiled strip is considered as a product coil.



Key features

- Multiple rings of equal length are produced from one entry coil.
- The measured values over the full strip profile are stored equally for all partial coils.
- One entry coil is used to create *n* product coils and, consequently, *n* product files.
- The assignment of location-specific measurement values to the respective strip can be performed during the post-processing of the measurement files, e.g., for strips at the edge and strips in the center.

Typical measured values

Thickness, width, strip width, strip tension,

ibaQDR

Property	Comment
Tracking ID	One ID per strip
Tracking start	Material detection at first measuring location in entry section
First measuring location	Identical to tracking start
“Finished” signal	Strip end detection at coiler or shear cut
Length resolution	1 m

7 Adjustments with virtual signals

The “Virtual” data interface provides a function in the I/O manager, which enables virtual, i.e. calculated or linked, signals to be used for acquisition.

Any virtual signals can be created using mathematical and logical operations. These virtual signals can be stored like other measurement signals or used to easily implement complex trigger conditions. Virtual signals enable calculations to be performed during the measurement, for example, calculating totals or differences, checking for tolerance violations, etc.

The procedure for creating virtual signals and a list of available functions can be found in the *ibaPDA* manual, Part 3 (Interfaces and modules) and Part 4 (Expression builder).

In terms of *ibaQDR* the virtual signals can be used for a wide range of conversions and adjustments, and even for planning material tracking.

For example, in *ibaQDR* the virtual signals enable a length measurement signal to be generated from meter pulses.

The following table shows a selection of the functions for calculating virtual signals, which are frequently used in *ibaQDR*.

Function in Expression builder	Application
OneShot	Detection of change of condition, value changes, edges; Use as trigger or for reset
INT	Integral function Calculation of length based on speed
COUNT	Counting level passes and edges Conversion of meter pulses into length
DELAY	Delaying a signal by a certain number of acquisition cycles (samples)
DelayLengthL / DelayLengthV	Dynamic delay of a signal by a distance in length units based on a length or speed signal

The following chapters contain some application examples.

7.1 OneShot

```
OneShot('Expression')
```

Arguments

'Expression'	Signal to be monitored for changes.

Description

This function returns the result TRUE, if the current value of 'Expression' is not equal to the previous one. It returns the result FALSE, if the current measured value is equal to the previous one. The function also supports text signals.

Therefore, the function can be used to detect changes in the value of analog signals and edges of digital signals.

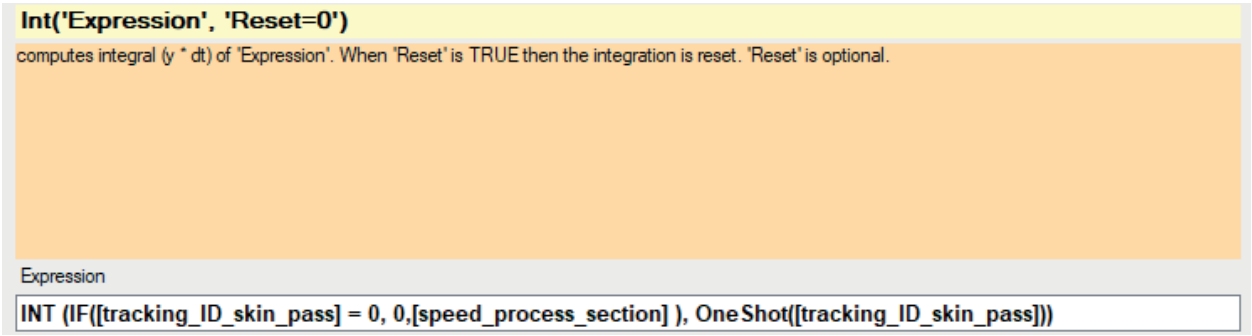
Application example

Use as a reset signal when calculating strip length using the integral function

```
INT (IF([tracking_ID] = 0, 0,[speed] ), OneShot([tracking_ID]))
```

If the tracking ID is zero, nothing is integrated (0). If the tracking ID is not equal to zero, the speed [speed] is integrated. If the tracking ID changes at the beginning of a new strip (tracked part), OneShot generates a pulse (TRUE for one cycle) which is used to reset the integral and thus the length value to zero.

The figure shows the example in the expression builder.



7.2 COUNT

```
COUNT('Expression', 'Level*', 'Hysteresis*', 'EdgeType*', 'Reset*')
```

Arguments

'Expression'	Signal (digital or analog) that supplies the meter pulse	
'Level*'	Specification of the level value If 'Expression' is a digital signal: Level = 0.5 If 'Expression' is an analog signal: Level = Tolerance value	
'Hysteresis'	Specification of a hysteresis band If 'Expression' is a digital signal: Hysteresis = 0 If 'Expression' is an analog signal: Hysteresis = Permitted value fluctuation about 'Level' without it being counted again	
'EdgeType*'	Indication of whether rising, falling, or rising and falling edges should be counted	
	'EdgeType' <0	Only falling edges (leaving hysteresis band in negative direction)
	'EdgeType' >0	Only rising edges (leaving hysteresis band in positive direction)
	'EdgeType' = 0	Falling and rising edges
'Reset'	Optional digital parameter that can be used to reset the counter. 'Reset' can also be an expression itself.	
	'Reset' > 0	Counter is reset
	'Reset' = 0	Counter value is retained / continues to count (default)

Description

The function counts the crossings of 'Expression' through 'Level'. The 'Hysteresis' parameter can be used to define a tolerance band which is above and below 'Level' by equal amounts. Only complete crossings through the tolerance band are counted. The 'EdgeType' parameter determines which kind of edges are counted. The 'Reset' parameter is used to reset the counter value to 0. 'Reset' can also be formulated as an expression.

Example

Calculation of strip length by counting meter pulses

```
COUNT ([meter pulse], 0.5, 0, 0, Oneshot([tracking_ID]))
```

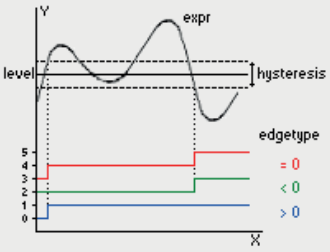
Each time the digital signal [meter pulse] changes from TRUE to FALSE or from FALSE to TRUE, the counter is incremented by 1. The counter value corresponds to the length in meters. If the tracking ID changes (= new tracked part), the OneShot expression returns the value TRUE for one cycle and thus resets the counter to zero.

The figure shows the example in the expression builder.

Count('Expression', 'Level*', 'Hysteresis*', 'EdgeType*', 'Reset=0')

Counts the number of 'Level' crossings of 'Expression'.
Changes in a range with size 'Hysteresis' around 'Level' are ignored.
'EdgeType' determines which edges are counted :
value < 0 : only falling edges
value > 0 : only rising edges
value = 0 : both rising and falling edges.
The count is reset when 'Reset' is 1.

Parameters ending with * are only evaluated once at the start of the acquisition.



Expression

COUNT ([meter pulse gauging roll 1], 0.5, 0, 0, Oneshot([tracking_ID_gauging_roll_1]))

7.3 Delay

```
Delay('Expression','NumberSamples*')
```

Arguments

'Expression'	Signal to be delayed
‘NumberSamples’	Number of samples or acquisition cycles by which the signal is to be delayed

Description

This function returns a delayed copy of the ‘Expression’ signal. The delay is specified in number of measurements ('NumberSamples'). The result is a signal curve with the values of the original signal for ‘NumberSamples’ before the current time.

To avoid a memory overload, ‘NumberSamples’ is limited to 10,000.

7.4 DelayLengthL / DelayLengthV

```
DelayLengthL('Expression','Length','MaxLengthDelta','DelayInMeter','Resolution*',
'Filter=0*'Argument')
```

```
DelayLengthV('Expression','Speed','DelayInMeter','Resolution*', 'Filter=0*')
```

Arguments

'Expression'	Input signal to be delayed	
'Length' or 'Speed'	Actual length signal value (in m) for DelayLengthL Actual speed value (in m/s) for DelayLengthV	
'MaxLengthDelta'	Only for DelayLengthL Upper limit to take into account changes in the length signal	
'DelayInMeter'	Distance to be delayed (in m)	
'Resolution*'	Resolution: Length base of the result (in m)	
'Filter'*	Optional parameter (default = 0) to set the filter for time-length conversion	
	'Filter' = 1	Minimum filter
	'Filter' = 2	Maximum filter
	'Filter' = 0 and others	No filter

Description

This function uses the 'Length' signal (in m) or the ‘Speed’ signal to create a length-based version of 'Expression' with a delay of 'DelayInMeter' meters. Changes in the length signal that exceed the 'MaxLengthDelta' are ignored. The ‘Resolution’ is the length base of the result (in m).

These functions can be used to delay a signal by a particular distance. For example, this enables a signal to be assigned a length offset relative to a measuring location. The function is suitable for individual measurement signals that have a longer distance to the associated measuring location (> 1 m), but for which setting up a separate measuring location is not worthwhile.

This also enables simple material tracking to be implemented, as described in the following application example.

Application example

Setting up simple material tracking using the example of the simulation model.

The material tracking system ensures that the change in the ID of the tracked part always takes place at the correct length at the measuring locations. It considers the standard measuring locations at the transitions between line sections with different speeds.

In the example of our simulation, these are the following line sections:

- From welder to gauging roll 2, including entry looper
- From gauging roll 2 to skin pass
- From skin pass to gauging roll 5, including exit looper

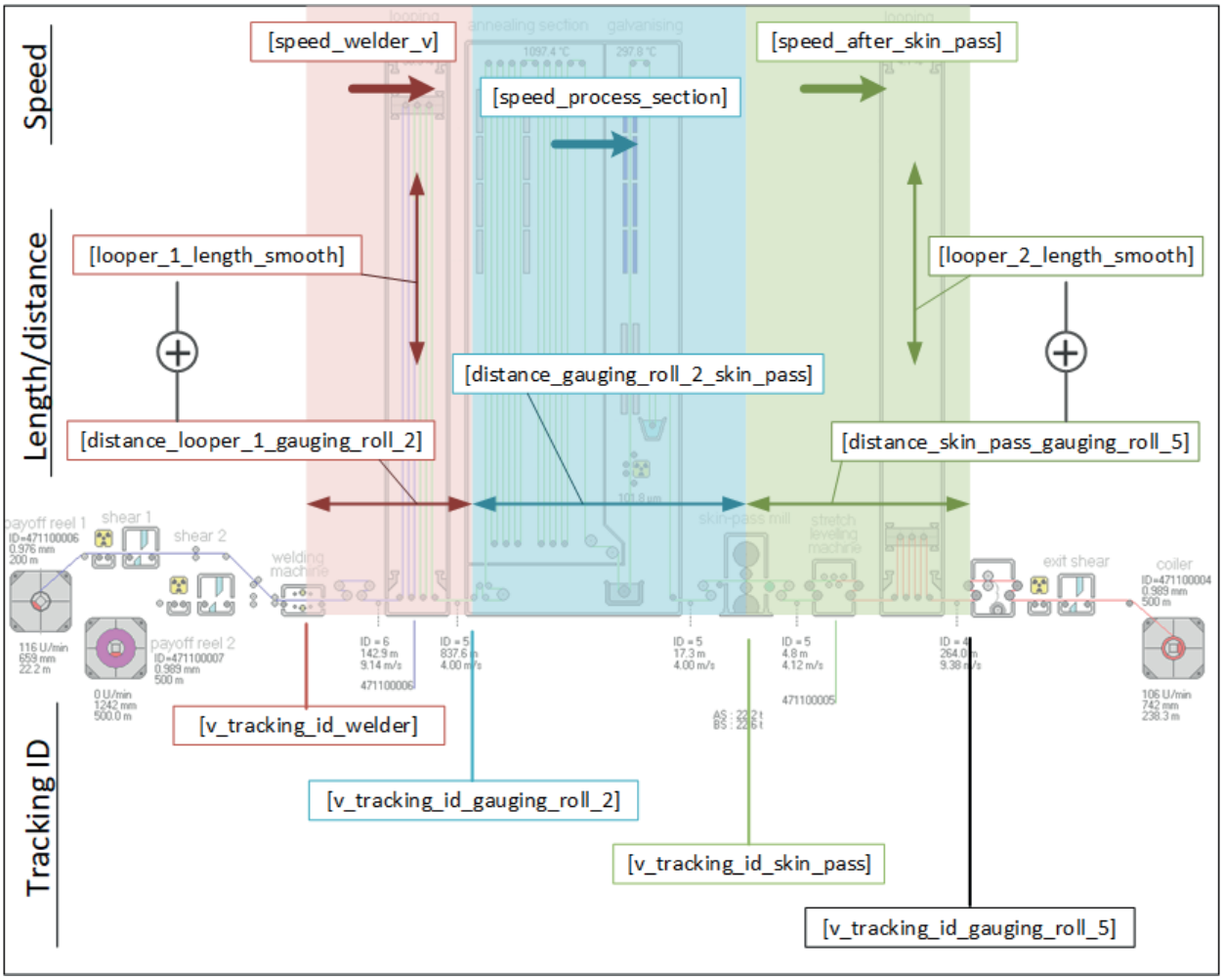
Therefore, the tracking IDs for 4 measuring locations have to be determined:

- Welder
- Gauging roll 2
- Skin pass
- Gauging roll 5

To change the tracking ID at the correct length, the following distances have to be determined:

- Strip length between welder and gauging roll 2, consisting of a constant component and a variable component (entry looper fill level)
- Strip length between gauging roll 2 and skin pass, in this case constant
- Strip length between skin pass and gauging roll 5, consisting of a constant component and a variable component (exit looper fill level)

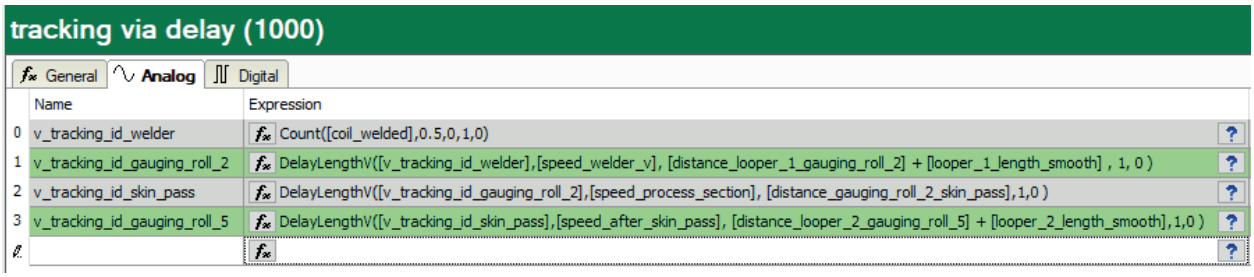
The following figure shows the principle and the signal names, as used in the simulation example.



The principle is based on the idea that the ID assigned at the welder is transferred to the second standard measuring location delayed by the strip length, and from there to the third standard measuring location and so on.

The distances traveled between the measuring locations are calculated from the speed signals for the corresponding line sections using the *DelayLengthV* function.

The definition of the virtual signals looks like this:



The ID is first determined at the welder. In the simulation this is done using a simple counter, which increments by one with each “Welded on” signal.

To determine the tracking ID at the next measuring location “[v_tracking_id_gauging_roll_2]” the *DelayLengthV* function is used:

```
DelayLengthV([v_tracking_id_welder],[speed_welder_v], [distance_
looper_1_gauging_roll_2] + [looper_1_length_smooth] , 1, 0 )
```

The value for the tracking ID is only adopted at the next measuring location when the strip has traveled the distance corresponding to the spacing of the two measuring locations, taking into account the fill level of the entry looper.

The same method is used for the subsequent line sections.

Measuring locations between the standard measuring locations do not have to be taken into account, as they are configured as measuring locations with offset and consequently do not require a separate tracking ID signal.

8 Appendix

8.1 Sales conditions

Qualification requirements for ibaQDR integrators

The *ibaQDR* software product is a data acquisition system for quality data, with the distinctive feature of simultaneous storage of time and length-based measurement values.

It is used particularly in strip lines (e.g. processing lines, cold and hot rolling mills). Here, multiple entry materials (“tracked parts” - coils, slabs, etc.) are transported through the line and processed, either welded together or individually. At the end of the process, a subset or superset of the entry materials is cut as a product and/or sample.

The same principle applies to production lines in other industrial fields, where long or web-shaped products are manufactured, such as rubber calanders, paper machines or extrusion lines, as well as plants for plate glass and wood panels. Here, *ibaQDR* can be used as well.

In *ibaQDR* the monitored production area is divided into different sections or locations (measuring locations). A distinctive feature of *ibaQDR* is the necessity for the measuring locations at which measured values occur to be assigned to a material tracking system, which supplies *ibaQDR* with tracking IDs, product speeds or lengths, etc. There are other adjustment options for assignment of text signals and possible special measuring location types (*In front of tracking start, with offset, reversing, dividing measuring location*).

The use of *ibaQDR* thus requires that the various interfaces to the material tracking system and to the control system are very carefully planned and configured appropriately on the *ibaQDR* side.

To provide and practice this knowledge, iba AG has developed an example simulator (hot-dip galvanizing line process).

Standard training on this simulator provides a fundamental understanding of the technical process and the *ibaQDR* configuration options. Further, more extensive training opportunities are provided by the playback mode for time-based *ibaPDA* data stores from real-world systems.

All training and support activities that are mandatory for initial delivery to an equipment supplier are listed below. After successful training and support on an initial project, the equipment supplier is awarded an *ibaQDR* certification and can then offer the product to their end customers and purchase it from iba AG without any further training requirements.

Training and support activities for equipment suppliers as new ibaQDR user

1. ibaQDR standard training (Fürth training center) ■ Introduction to *ibaQDR* ■ Configuration options on the simulator ■ From entry to product coil using the simulator ■ Configuration of *ibaQDR*, creating the measuring locations, assigning the measuring signals ■ Analysis using *ibaAnalyzer*, checking accuracy and correctness of strip tracking ■ Processing of time and length-based measured values in the data file (*.dat) through to storage in the database and creation of reports	1 day
2. Support for system planning (remote) ■ Presentation of *ibaQDR* operating modes ■ Configuration of *ibaQDR* with real-world line data ■ Detailed clarification of line operating modes in relation to *ibaQDR* and interface information between *ibaQDR* and the line automation.	1 day
3. Support for commissioning (remote) ■ Testing via playback ■ Analyses and plausibility checks with *ibaAnalyzer*	3 days
4. Equipment supplier certification ■ Presentation of certificate to equipment supplier	

8.2 Monitoring ibaQDR

Because it acquires, analyzes and documents quality data, correct functioning of the *ibaQDR* system is essential for the production process. Therefore, it is advisable to continuously monitor the functioning of the *ibaQDR* system so that any faults can be immediately identified and resolved.

ibaQDR provides the options familiar from *ibaPDA* for this purpose:

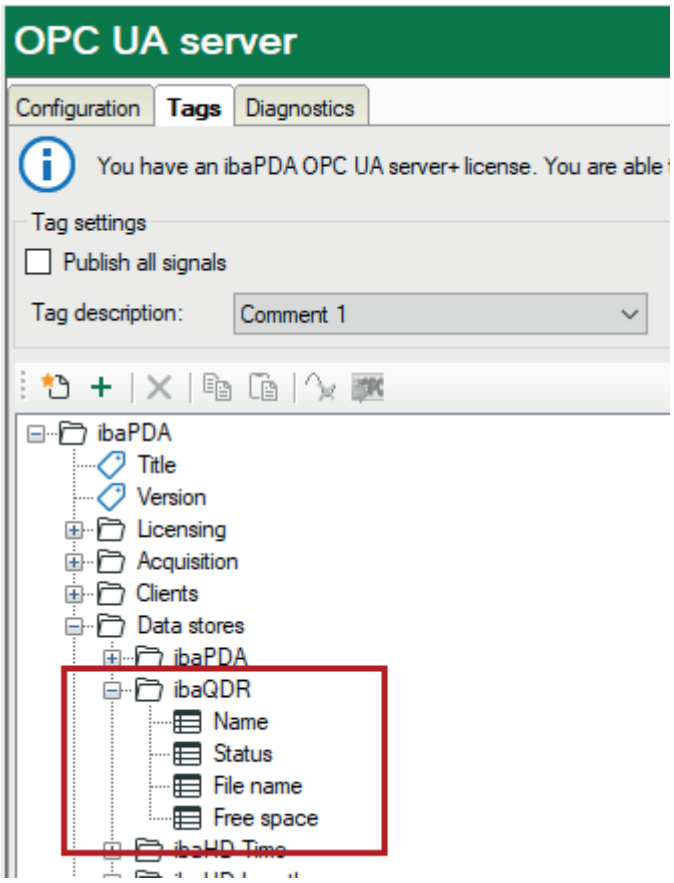
- OPC UA server
- SNMP server
- Watchdog telegram
- Virtual signals

8.2.1 OPC UA server

By default, *ibaPDA* provides the OPC UA server functionality in order to make data and information about its own status publicly available.

In addition to the standard information, for *ibaQDR*, it contains the information for the *ibaQDR* data store:

- Name
- Status
- DAT file name
- Free space on disk



Other documentation

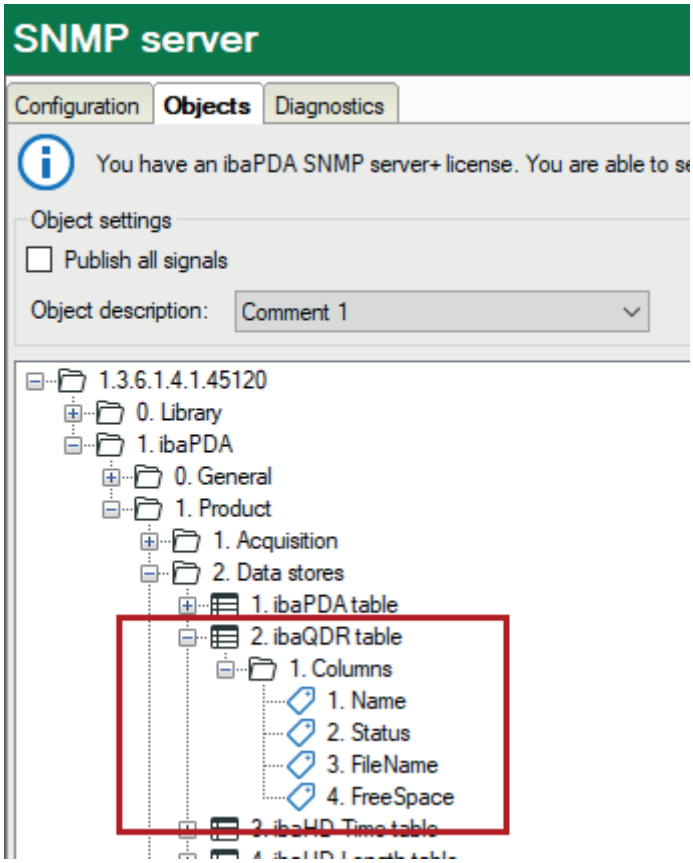
You can find further information in the manual for the *ibaPDA* product, Part 2 “I/O Manager”.

8.2.2 SNMP server

ibaPDA has a built-in SNMP server.

In addition to the standard information, for *ibaQDR*, it contains the information for the *ibaQDR* data store:

- Name
- Status
- DAT file name
- Free space on disk



In combination with the additional license *ibaPDA-SNMP-Server+* (order no. 30.670050), you can make every acquired or calculated signal available as object on the the SNMP server.

Other documentation

You can find further information in the manual for the *ibaPDA* product, Part 2 “I/O Manager”.

8.2.3 Watchdog telegram

The watchdog function in *ibaPDA* provides telegrams in binary or ASCII form, containing the most important information about the system status.

In addition to the standard information, for *ibaQDR* they include information about whether the QDR program is running (QDR, Running) and whether it has been stopped (QDR, Idle).

In addition, the telegram contains information about the *ibaQDR* data store:

- Status: Inactive/Not synchronized/Synchronized
- Current directory
- Free space on disk

The watchdog function can be used via TCP or UDP.

Other documentation

You can find further information in the manual for the *ibaPDA* product, Part 2 “I/O Manager” and Part 7 “Appendix”.

8.2.4 Virtual signals for monitoring

There are some helpful functions available in the expression builder, which provide information about the system status. The derived information can be visualized, recorded and processed as virtual signals.

For that purpose create a *Virtual* module under the *Virtual* interface in the I/O Manager. You will find the functions in the expression builder under the *Diagnostics* node.

In the following functions for monitoring the data store and the license are described.

8.2.4.1 DataStoreInfo

```
DataStoreInfo('DatastoreIndex*','InfoType*')
```

Parameters ending with * are only evaluated once at the start of the acquisition..

Description

This function provides information about the selected data storage. This information may be used for controlling other functions or for display and diagnostic purposes.

For normal (PDA) data storage, use 'DatastoreIndex' >= 0.

For *ibaQDR* data storage, use 'DatastoreIndex' < 0.

The index can easily be obtained by looking at the tree structure in the configuration dialog of the data record. Index increases top-down.

Specify the information type ('InfoType') you want to receive.

The following information types are supported:

Information types	Possible results
0: recording status	0 = stopped 1 = waiting for trigger 2 = recording 3 = posttrigger recording
1: Saving in the backup directory:	0 = base directory is used 1 = backup directory is used
2: recorded time in the current file expressed in seconds	Value is updated every second.
3: the free space on the current hard disk expressed in MB	Value is updated every minute.
4: is ibaQDR synchronized?	0 = ibaQDR is NOT synchronized 1 = ibaQDR is synchronized
5: Image triggers storing to backup directory	-1=No active or configured image triggers 0=All image triggers are using the base directory 1=All image triggers are using the backup directory 2=Some image triggers are using the base directory and some are using the backup directory
6: Number of overlapping data files	Actual value

Table 4: Information types and possible results of the DataStoreInfo function

8.2.4.2 DongleInfo

```
DongleInfo('InfoType*')
```

Parameters ending with * are only evaluated once at the start of the acquisition.

Note

This function is still supported for backward compatibility purposes. Please use instead the *LicenseInfo* function, which serves the same purposes while supporting soft licenses as well.

Description

This function provides information about various properties related to the dongle. This information can be used for display and diagnostic purposes.

Specify the information type ('InfoType') you want to receive. This type of information is determined and output at the start of the measurement. If you want to receive more information from the dongle then you need to configure the function several times, with a different type of information each time.

The following information types are supported:

Information types	Possible results
0: Dongle available	TRUE = dongle available FALSE = no dongle or dongle defective
1: Dongle time limit in days	Value of the remaining lifetime of the dongle
2: Demo time limit in days	Value of the remaining lifetime of the dongle for trial periods / demo versions
3: ibaQDR acquisition time limit in seconds	Value of the remaining time which continues to run the ibaQDR system after the dongle has been removed.
10: Dongle inserted - counter	Number of times the dongle was inserted
11: Dongle removed - counter	Number of times the dongle was removed
12: Dongle changed - counter	Number of times the dongles were changed

Table 5: Information types and possible results of the DongleInfo function

Note

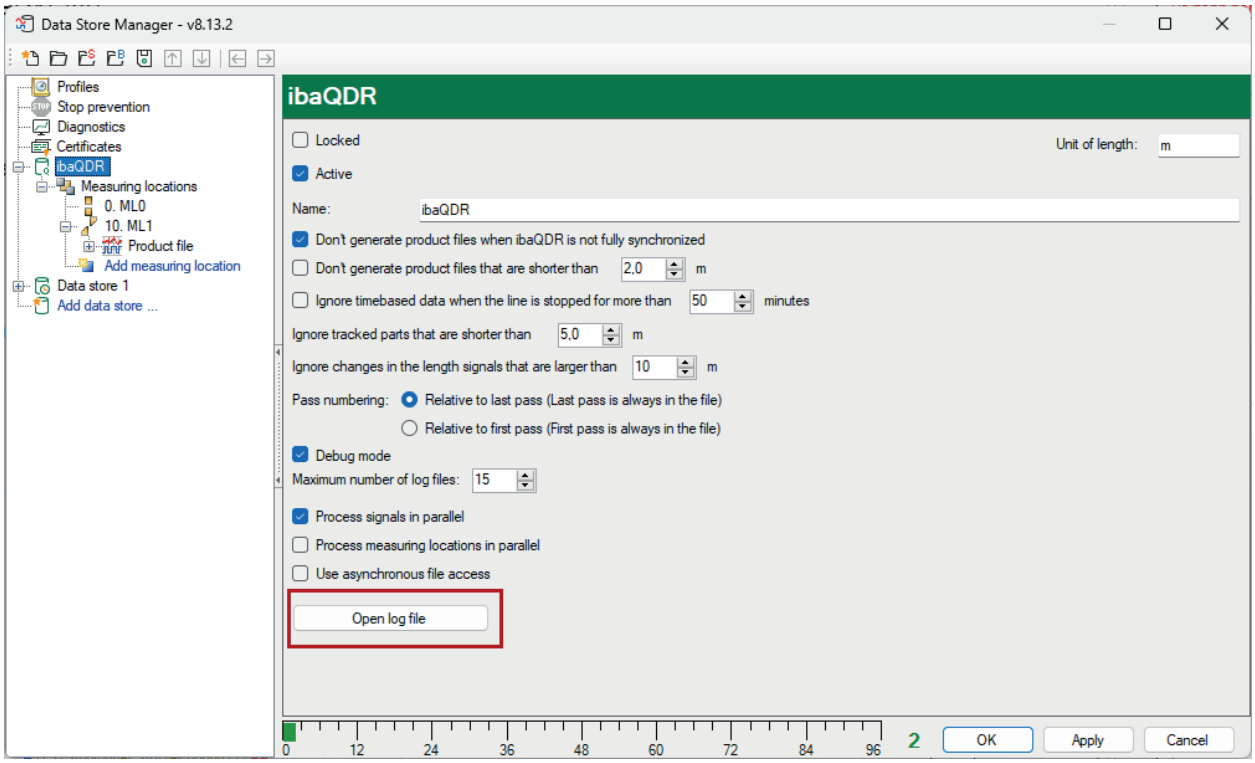
By evaluating information type 0, you can monitor whether a license container is present or available. In practice, you can use this function in *ibaQDR* to generate an email when it is detected that the license is missing (e.g., dongle removed). With *ibaQDR*, data collection continues for up to 4 days without a license to give you the opportunity to procure a new license container without jeopardizing production. This grace period does not exist with *ibaPDA*.

8.3 Error messages in the ibaQDR event log

The files for the *ibaQDR* event log can be found at:

`C:\ProgramData\iba\ibaPDA\Log`

However, the current log file can also be called up directly from the data store configuration using the <Open log file> button.



ibaQDR error messages
{x} is replaced by a value

{ML} is replaced by the measuring location name

Error message	Description	Visible in client with active debug mode
Peak detected of {x}s on coil ID of {ML}	This error occurs if the tracked part ID has changed twice within 2 s. This is normally an indication of a tracking error in the ID signal.	x
No time data found for {x} in entrycoil {y}	No time data available for signal x in entry product y.	
Stopping QDR datastore because an error occurred in measuring location {ML}	An unexpected error occurred at the measuring location, which led to the *ibaQDR* data store being stopped. Check the *ibaQDR* and *ibaPDA* server log files for more information about the error.	

Error message	Description	Visible in client with active debug mode
Product file {x} already exists	A file with the name {x} already exists. *ibaQDR* will attempt to assign an alternative file name by appending a number.	
Error creating file {x}: {y}	Generation of product file {x} failed. {y} specifies the reason. This error leads to *ibaQDR* being stopped.	
Error creating directory {x}: {y}	The product directory {x} could not be created. {y} specifies the reason. If {x} was the base directory, *ibaQDR* attempts to use the backup directory to generate the product files. If {x} was the backup directory, *ibaQDR* is stopped.	
Exception in QDR thread: {x}	Error {x} occurred during processing of *ibaQDR* product files. This error leads to ibaQDR being stopped.	x
There are entrycoils missing for {ML} --> this measuring location and all measuring locations in front of it are being ignored!!!	*ibaQDR* cannot find any data for the current product at measuring location {ML}. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty. The product file will not contain any data for this measuring location and for all upstream measuring locations.	
There is a {x}% difference between the exit length at {y} and the entry length at {z} for the product file {a}. Please check the length signals.	The difference between the exit length at measuring location n and the entry length at measuring location n+1 is more than 10 %. In theory, the two lengths should be exactly the same. This error can occur if the tracking is faulty or if the configuration of the elongation for measuring location n+1 is incorrect. *ibaQDR* deals with this difference by stretching the exit length from measuring location n uniformly to the entry length for measuring location n+1.	x
Processing time-based data failed: {x}	Error {x} occurred during processing of the time-based data from measuring location n. *ibaQDR* ignores the time-based data from measuring location n and generates a product file at measuring location n-1.	

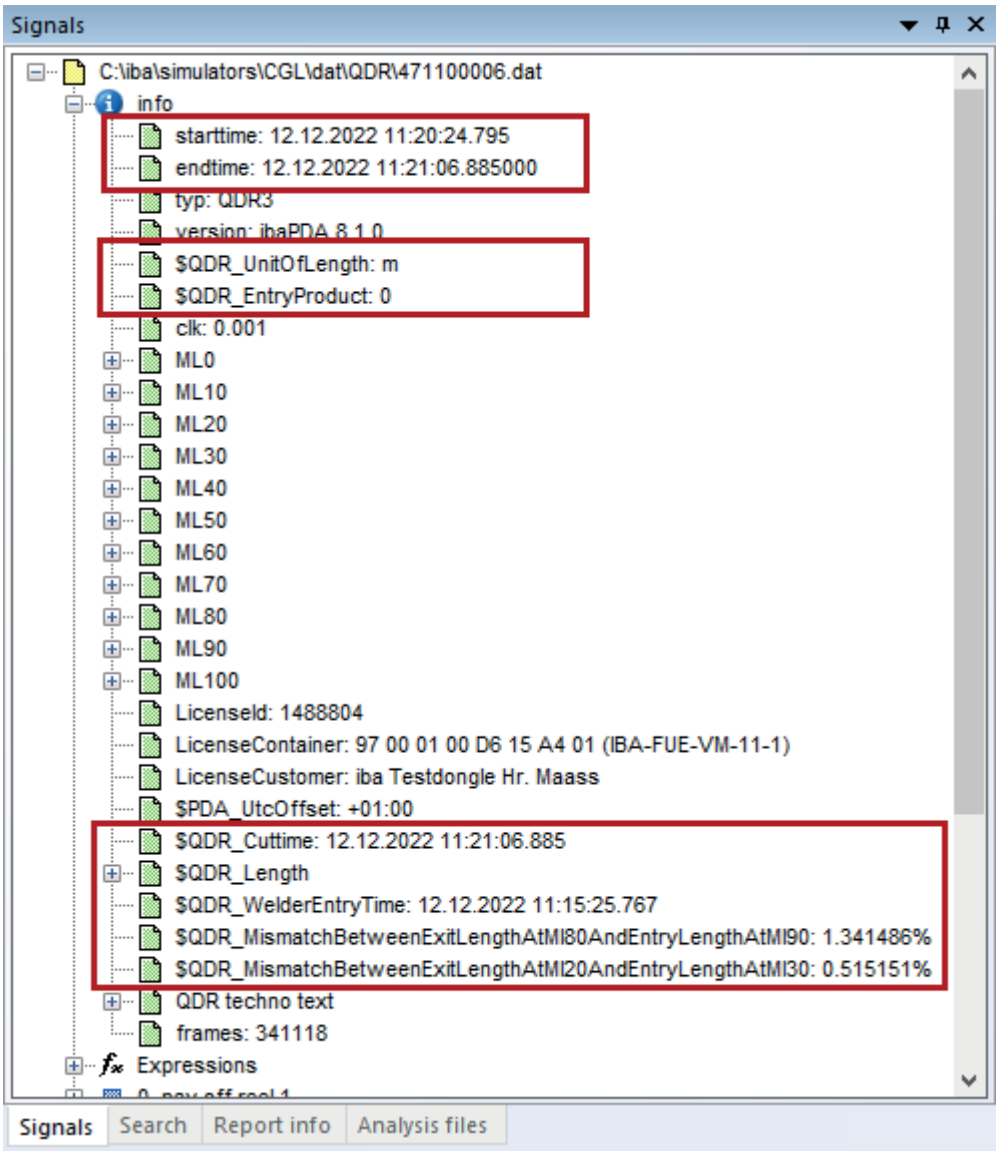
Error message	Description	Visible in client with active debug mode
Data is missing in QDR productfile {x}. Examine the qdr log for more details.	More than 5 % of the data in product file {x} is missing. This is indicated in the info field *$QDR_DataMissing*. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty (incorrect length signals). The following errors then appear in the *ibaQDR* log file.	x
ID={x} --> Insufficient data: needed data up to {y}m, but data only available up to {z}m	This error occurs if no data is available at {y} m of ID {x}. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty (incorrect length signals). *ibaQDR* has data up to {x} m. The missing data {y} - {z} is ignored.	
ID={x} --> Data needed up to {y}m, but data only available from {z}m --> No data available for this product coil at this measuring location	This error occurs if no data is available at {y} m of ID {x}. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty (incorrect length signals). The product file will not contain any data for this measuring location and all upstream measuring locations.	
ID={x} --> Insufficient data: needed data from {y}m, but data only available up to {z}m	This error occurs if no data is available at {y} m of ID {x}. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty (incorrect length signals). *ibaQDR* has data up to {z} m. The missing data {y} - {z} is ignored.	
ID={x} --> Data needed from {y}m, but data only available from {z}m	This error occurs if no data is available at {y} m of ID {x}. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty (incorrect length signals). *ibaQDR* starts the coil at {z} m instead of {y} m. The missing data {z} - {y} is ignored.	

Table 6: ibaQDR error messages in the log file

8.4 Info fields in the ibaQDR product files

The DAT files generated by *ibaQDR* contain some information in the form of info fields.

In *ibaAnalyzer* you will find the info fields in the signal tree for the opened DAT file under the Info node.

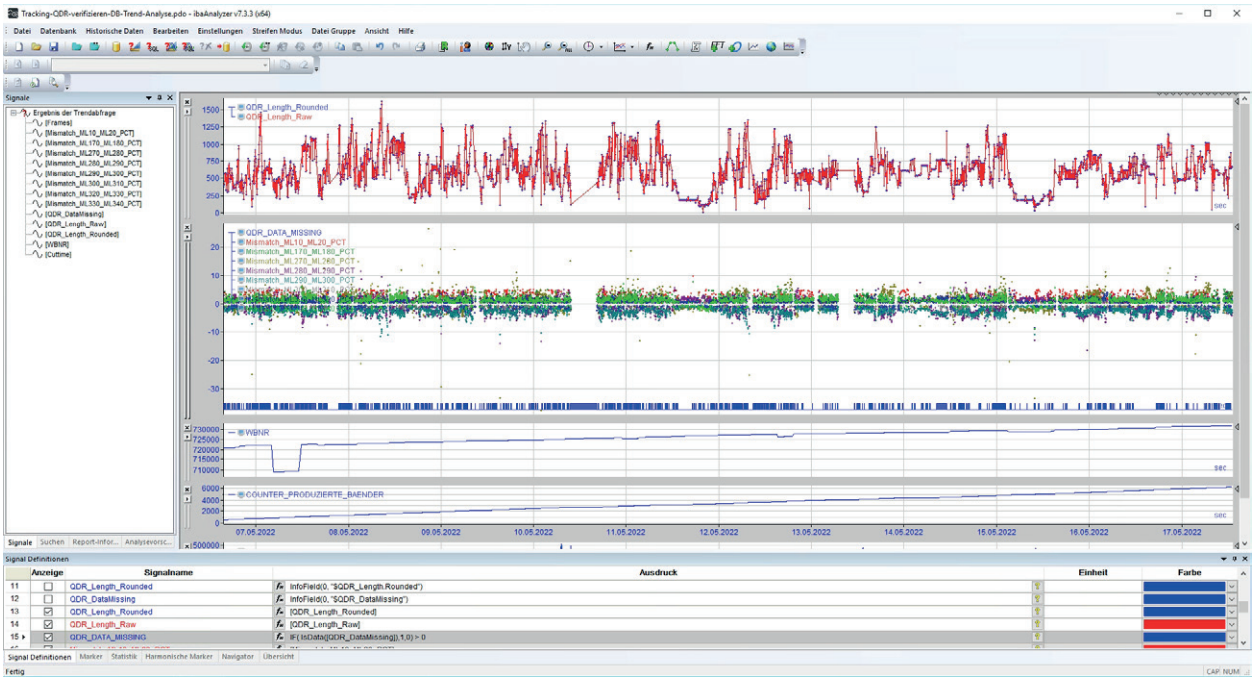


Info field	Description
$QDR_NotSynchronized	*ibaQDR* was not fully synchronized. This means that data could be missing, particularly at the first measuring locations.
$QDR_DataMissing	More than 5 % of the data is missing. This can happen if *ibaQDR* is not yet fully synchronized or if the tracking is faulty.
starttime	The time at which the beginning of the finished product (finished coil) passed the last measuring location (shear).
endtime	The time at which the end of the finished product (finished coil) passed the last measuring location (shear). This is normally the cut time.
$QDR_UnitOfLength	Length unit for representation and labeling
$QDR_EntryProduct	0: If data file is a finished product file 1: If data file is an entry product file
$QDR_Cuttime	The time at which the end of the finished product (finished coil) passed the last measuring location (shear). This normally corresponds to the 'endtime'
$QDR_Length ... Raw ... Rounded ... LBase	Determined length of the product Original measured length value (measurement device, control) Length value rounded up/down to LBase Length base
$QDR_VirtualCut	This file was generated by a virtual cut. The value of the info field specifies the reason for the virtual cut.
$QDR_WelderEntryTime	Beginning of production from measuring location where tracking starts.
$QDR_MismatchBetweenExitAndEntryLengthAtMlx	The difference between the exit length at measuring location x and the entry length at measuring location x+1 is more than 0.5 %.

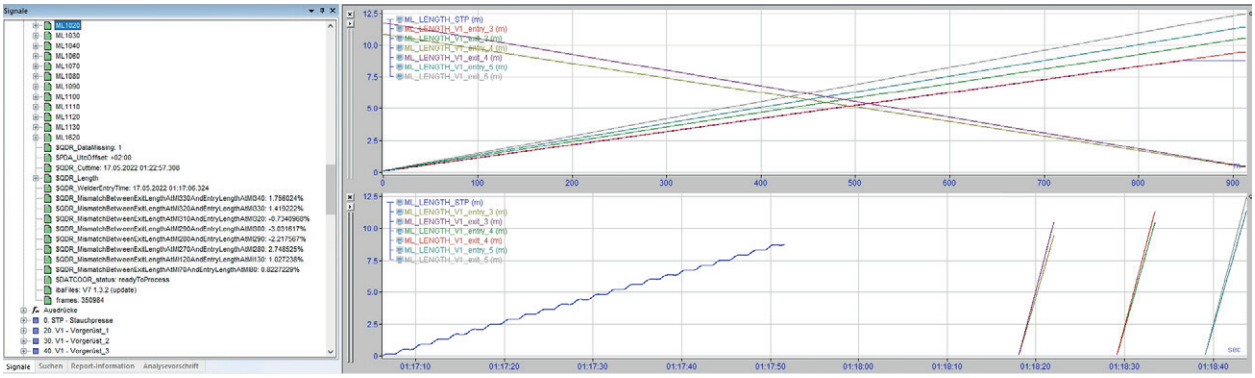
Table 7: Info fields in the ibaQDR product file

Example

The following example shows the trend for the warning info fields from the *ibaQDR* DAT files. To do this, an analysis has been developed in *ibaAnalyzer* and the values have been extracted to an SQL database using *ibaDatCoordinator-DB Extract*. These analyses are an effective way of monitoring the tracking performance.

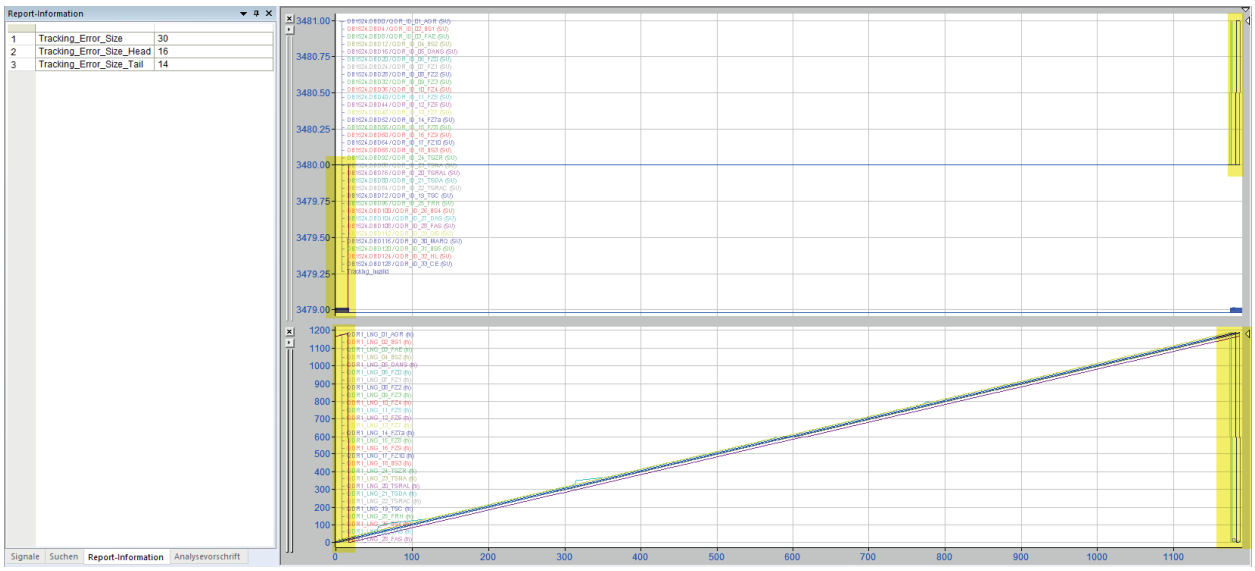


Here are some example extracts from an *ibaQDR* DAT file

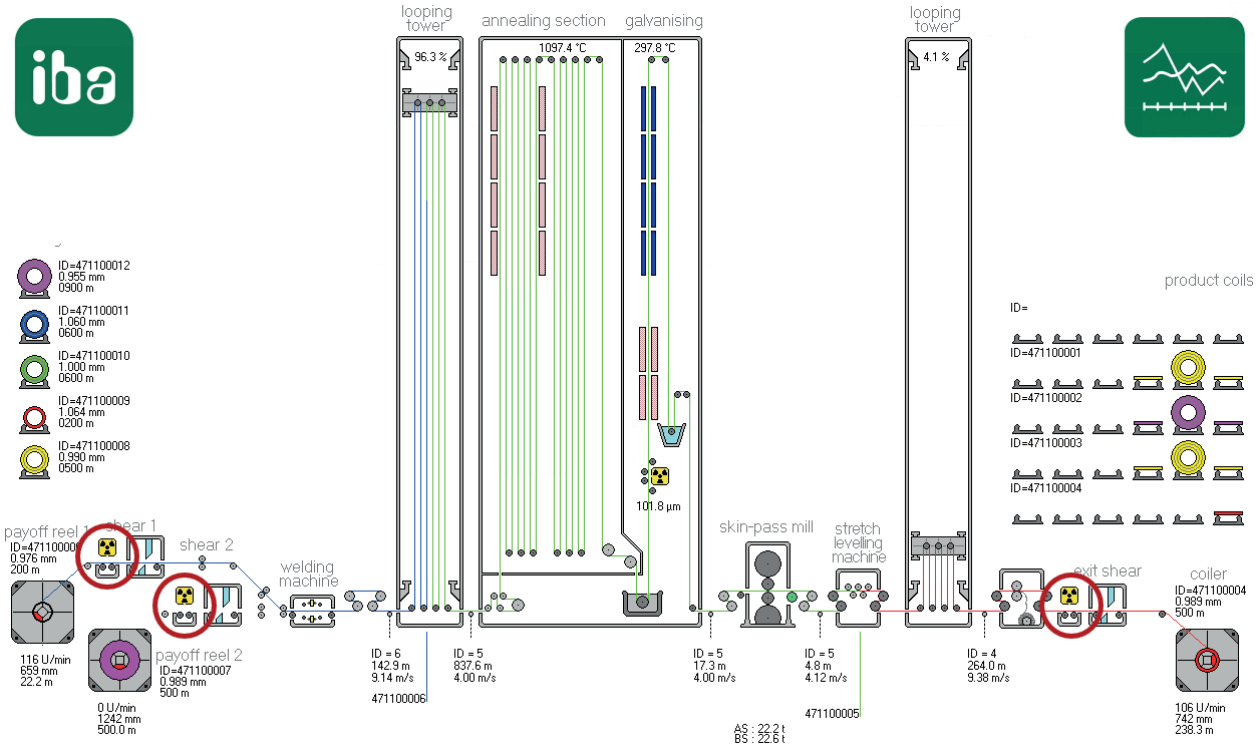


In the *ibaPDA* event log window, you will find warnings that require further analysis.

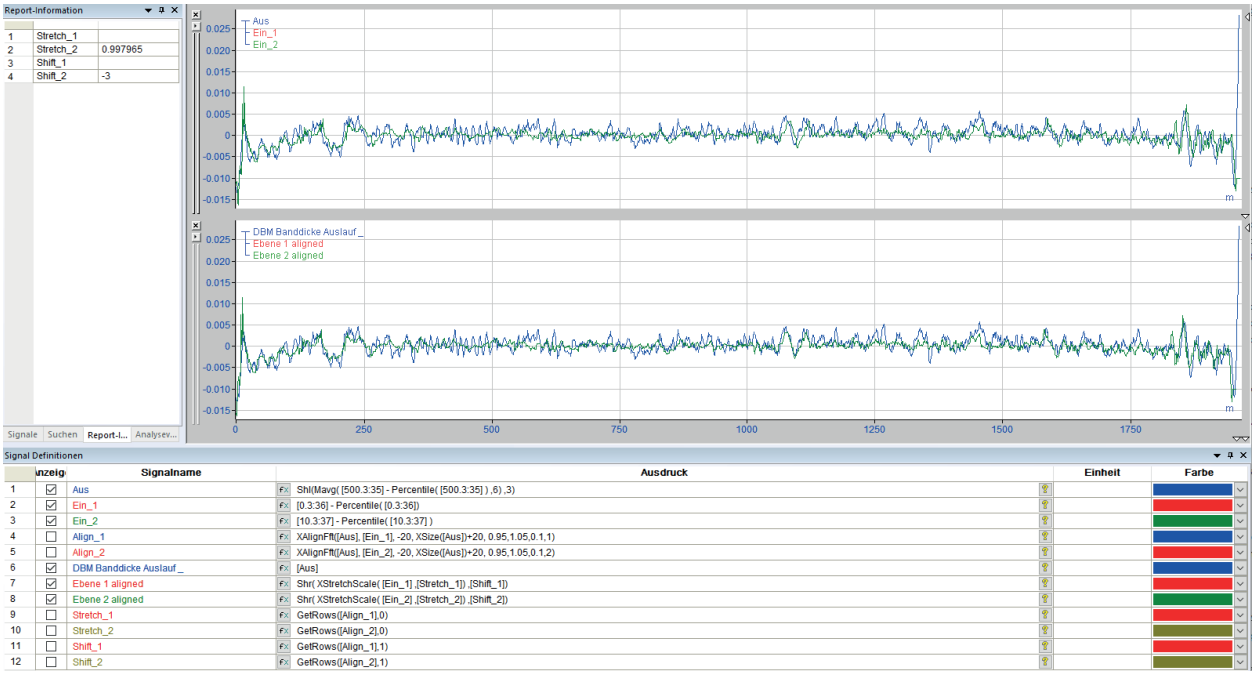
Looking at the tracking signals in the *ibaQDR* DAT file often provides a log of useful information. The following example shows that the tracking IDs and length signals demonstrate implausible “jumps”. The length of these areas “Tracking_Error_Size” has been highlighted and determined and can be monitored in conjunction with other KPIs, for example using a DB trend.



It may also be helpful to logically check the progression of useful signals. In the following example of a coating line, the thickness measurements from the entry section are compared with the thickness measurement in the exit section.

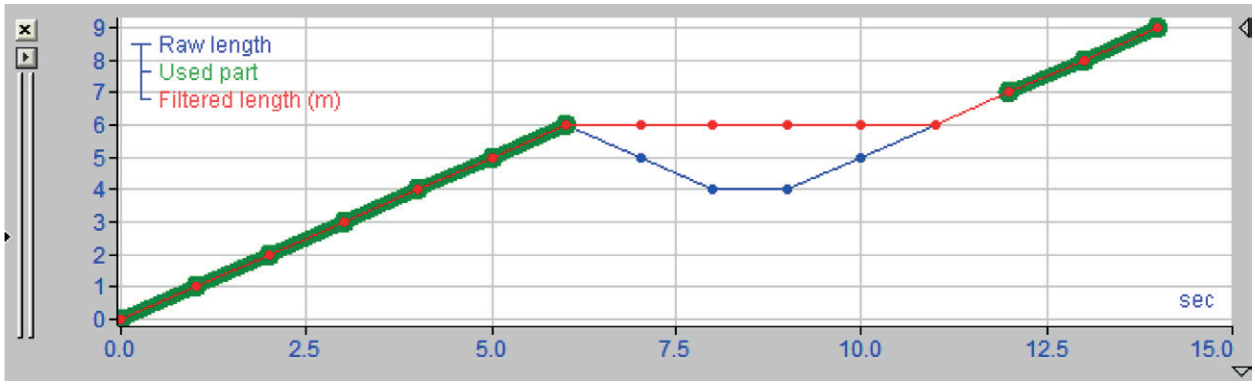


The *XAlignFFT* function even allows the offset or stretching of the relevant pairs of signals to be determined. With optimum tracking, the offset would be SHIFT = 0 and the stretching factor STRETCH = 1.0, but here the stretch is 0.998 and the shift is -3.



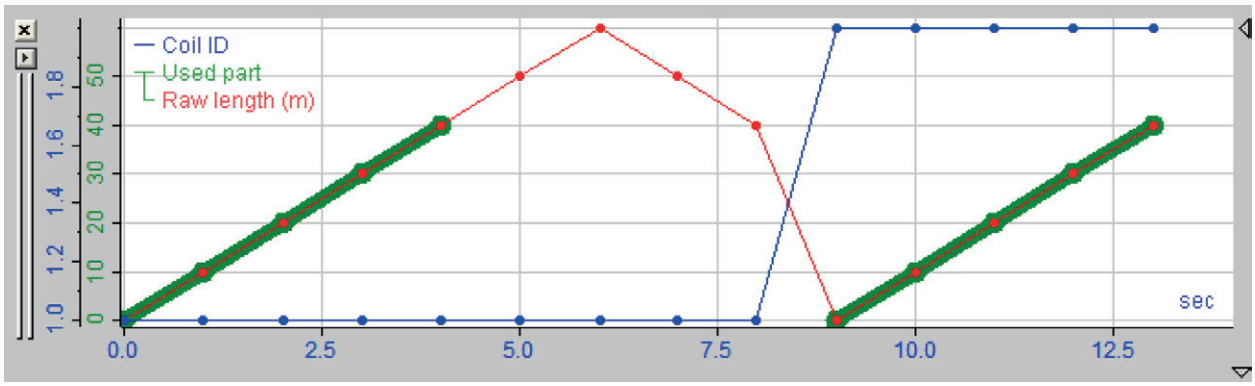
8.5 Mapping backwards travel

If an entry product (tracked part) moves backwards, the tracking length signal decreases. However, the internally filtered length signal in *ibaQDR* does not decrease; it remains at the last maximum value. When the entry product starts moving forwards again, the tracking length signal increases again. When the tracking length signal is greater than its previous maximum, the internally filtered length signal in *ibaQDR* also starts to increase.



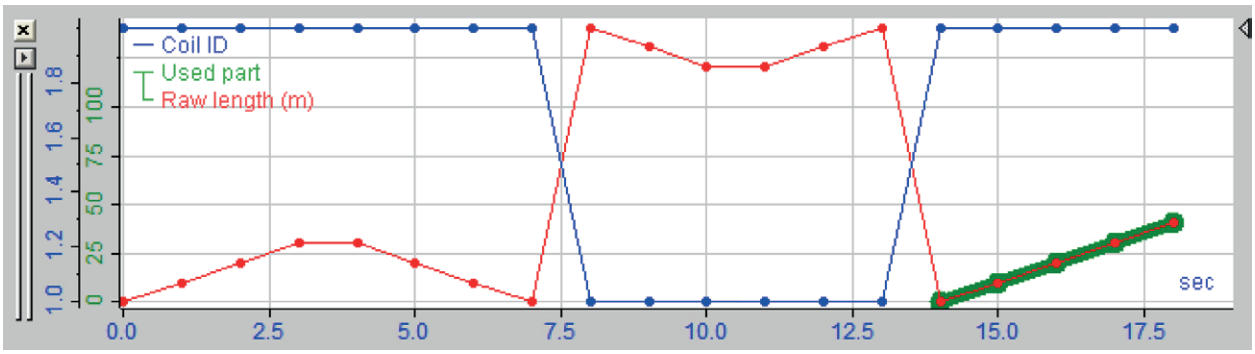
Measured length value (blue), filtered length signal (red) and length values used (green)

It is possible that an entry product is moved backwards and a piece of it is cut off. The entry product length at the change of ID is then shorter than the previous maximum length. *ibaQDR* detects this and the measured values for the missing length sections are removed. In the following example, the entry product was 60 m long. It was then moved back 20 m and cut. A new entry product with tracking ID (coil ID) 2 is then welded on. *ibaQDR* now has an entry product with ID 1 and a length of 40 m.



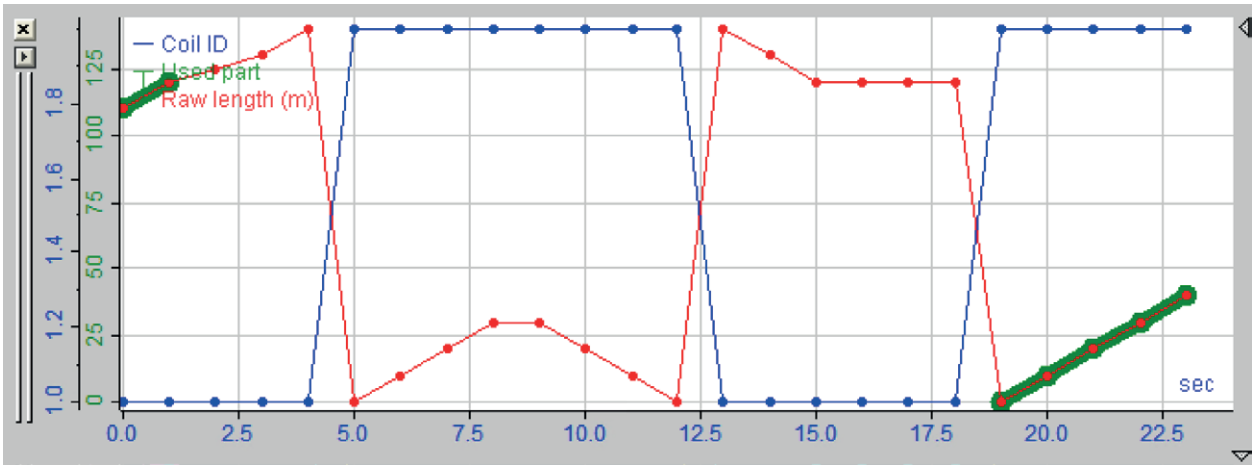
Entry product ID (blue), measured length value (red) and length values used (green); note: The right-hand green line belongs to the next entry product with ID 2.

Moving back can be more complicated if an entry product is completely discharged from the line. The following example shows this situation.



Entry product ID (blue), measured length value (red) and length values used (green)

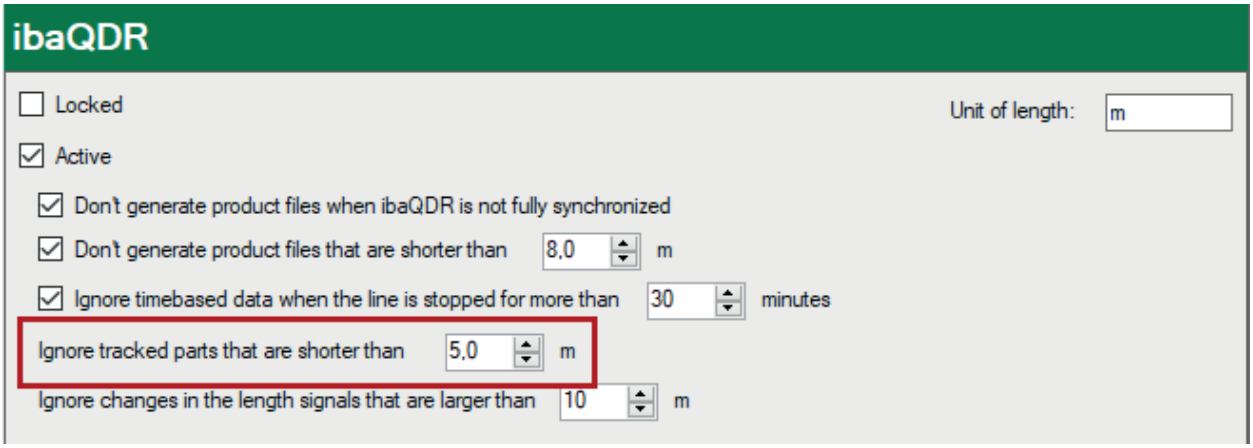
Entry product 2 moves into the line and moves 30 m. It is then moved back. The file for entry product 2 is closed at this point. The total length of entry product 2 is 0, as it has been completely discharged. Entry product whose length is less than MinEntrycoilLength (default = 5 m) are removed from *ibaQDR*. The entry product file for ID 2 is therefore removed. The line continues running, which means that ID 1 appears at this measuring point again. It begins at its final length of 140 m and then reverts to 120 m. ID 1 then advances back to 140 m. The filtered length for ID 1 is then 0 m. Therefore, this entry product file is also removed. ID 2 then moves back into the line. A new entry product file is generated and this is used. This section is shown in green in the illustration. There is also another situation that can occur.



Entry product ID (blue), measured length value (red) and length values used (green)

Entry product 2 moves into the line and is completely discharged. Entry product 1 is moved back 20 m and then cut. Entry product 2 moves in again. *ibaQDR* has removed the first file for entry product 2 and the second file for entry product 1. *ibaQDR* has also detected that entry product 1 has been cut, i.e. it will remove 20 m from the first file for entry product 1. The green sections in the figure show the data used for entry products 1 and 2.

The value for MinEntrycoilLength can be specified in the dialog box for configuring the data store:



9 Support and contact

Support

Phone: +49 911 97282-14

Email: support@iba-ag.com

Note

If you need support for software products, please state the number of the license container. For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Gebhardtstrasse 10-20
90762 Fuerth
Germany

Phone: +49 911 97282-0

Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and worldwide

For contact data of your regional iba office or representative
please refer to our web site:

www.iba-ag.com