

# Profile Data Contents

[ [Count](#) ] [ [Inputs](#) ]

Independent of scanning mode, the returned profile data from the scan head always contains:

Name	.NET API	C/C++ API	Description
Count	Profile.Count	tagProfile.numberPoints	The number of data points in the profile
Flags	Profile.Flags	tagProfile.flags	The flags attached to this scan. See Table below.
Inputs	Profile.Inputs	tagProfile.inputs	The values of the input lines (Channel A/Channel B/Start Scan) when the scan was taken. See Table below.
LaserIndex	Profile.LaserIndex	tagProfile.laserIndex	Index of the laser used for this profile. For single-laser models, this is always 0, for multi-laser models, the laser at end farthest away from the camera has the index 0.
LaserOnTime	Profile.LaserOnTime	tagProfile.laserOnTime	The amount of time the laser was on for this scan. Unit: microseconds.
Location	Profile.Location	tagProfile.location	The encoder value at the time the scan was taken.
SendLocation	Profile.SendLocation	tagProfile.sendLocation	The encoder value when the scan was retrieved from the scanner, i.e., when it was taken out of the scanner queue.
SequenceNumber	Profile.SequenceNumber	tagProfile.sequenceNumber	The sequence number of the scan. This number is set to 0 when synchronized scanning is restarted.
TimeInHead	Profile.TimeInHead	tagProfile.timeInHead	The time since the scanner was powered on. Unit: milliseconds.
Data	n/a (see below)	tagProfile.data	Measured data points

In the .NET API, a profile is of data type `IEnumerable<ProfileDataPoint>` and provides an indexing operator. Therefore, accessing the actual data points can be achieved via simple indexing. In the C/C++ API, `tagProfile.data` is an array of fixed size that contains all measured points.

## Count

For each scan, a Profile data structure is returned, regardless of whether points were measured. If the count is always 0, check the scan window size, exposure parameters, laser light blockage or obstructions in the camera view.

## Flags

In each profile, these flags can be present. Your software should check for the error flags and take appropriate action.

Flag Name	Description
None	Normal scan, no errors detected.
Overrun	The scan data is valid but some data at the edges of the scan may be missing.
Internal Error	An internal error was encountered in the scan head. This data should not be used. This flag is rare and can be caused by scanning extremely fast.
SequenceError	A profile sequencing error occurred. This usually indicates that the scanner is being run too fast or the scan queue in the scanner was allowed to overflow. This can happen when the scanner is running in synchronized scan mode but <code>GetQueuedProfiles</code> has not been called for a period of time, allowing the internal queue to overflow.

## Inputs

This field describes the status of the hardware input lines at the time the scan was taken. The API maps the 32 bit unsigned integer to a struct with the following flags set. JSdiag displays the value as a bit field that also contains the Cable ID (for old models with a DB-25 connector).

Field	Description
None	No signal line active
EncoderB	Encoder B
EncoderA	Encoder A
StartScan	Start Scan signal active