

# Blade repair analysis user's guide v2.1.0

© 2018 Renishaw plc. All rights reserved.

Renishaw® is a registered trademark of Renishaw plc.

This document may not be copied or reproduced in whole or in part, or transferred to any other media or language, by any means, without the prior written permission of Renishaw.

The publication of material within this document does not imply freedom from the patent rights of Renishaw plc.

### **Disclaimer**

Considerable effort has been made to ensure that the contents of this document are free from inaccuracies and omissions. However, Renishaw makes no warranties with respect to the contents of this document and specifically disclaims any implied warranties. Renishaw reserves the right to make changes to this document and to the product described herein without obligation to notify any person of such changes.

### **Trademarks**

All brand names and product names used in this document are trade names, service marks, trademarks, or registered trademarks of their respective owners.

---

## Contents

1	Introduction.....	6
2	Command line arguments.....	7
3	Return codes and text output.....	12
4	Units.....	13
5	Input files.....	14
5.1	Input sweep scan data files.....	14
5.2	Input tolerances file.....	14
5.3	Input application settings file.....	18
5.4	Input nominal file.....	20
6	Output files.....	22
6.1	Output chord analysis file.....	22
6.2	Output blend analysis file.....	23
6.3	Output combined blend analysis file.....	25
6.4	Output offset sections.....	26
6.5	Output nominals.....	26
6.6	Output deltas.....	27
6.7	Output unfiltered features.....	27
7	DMIS curves.....	29
8	Guide to upgrading from version 1.....	30
8.1	Input file changes.....	30
8.2	Output file changes.....	31
9	Troubleshooting.....	32

## 1 Introduction

This document serves as a user guide for how to run the “BladeRepairAnalysis.exe” program. This program is designed to take sweep scan measurement data of a blade along with other settings, and then determine various parameters regarding this measurement, such as chord lengths or the location and characterisation of any perceived edge repairs.

Note that this document is primarily concerned with the running of the application itself and is not concerned with the technical details. Please see the Blade repair analysis process (Renishaw part number H-1000-5447) for the technical details.

## 2 Command line arguments

The executable demands a series of command line arguments of the form “/Argument:Value”. There are a series of mandatory arguments which must be provided, and if they are not, the application will fail, reporting an initialisation error. There are then a series of optional arguments which can alter the behaviour further.

Input arguments are either “values” or “switches”. In a “value” argument, the “Value” in the form shown above must be given. A “switch” argument is a Boolean state which is either on or off. A switch does not require a value to be given (i.e. “/Switch” is sufficient). A switch is deemed “on” if given and “off” if not given. Note that switches can also be turned on or off explicitly by passing “/Switch:true” or “/Switch:false” respectively. Arguments can be given in any order.

When interpreting arguments, the Windows command prompt delimits parameters based on whitespace characters. As such, if any value contains whitespace (e.g. a file path), it should be ensured that the entirety of the value part of the argument is encapsulated in quotation marks. File paths can be given either as paths relative to the current working directory or as absolute paths.

The blade repair analysis executable package contains sample batch files demonstrating how these arguments can be used. Additionally, calling the executable with no command line arguments will cause it to print a help message, detailing all the available arguments.

The table below lists out the set of supported arguments.

Argument type	Mandatory	Description	Argument format
Input Sweep Scan File Path	Yes	The path to the file containing the sweep scan measurement data (see the Input Files section for more details).	/InputSweeps:"SweepScans.out"
Input Tolerances File Path	Yes	The path to the file containing information on the allowable tolerances (see the Input Files section for more details).	/InputTolerances:"Tolerances.ini"
Input Application Settings File Path	Yes	The path to the file containing information on settings which affect the internal workings of the application (see the Input Files section for more details).	/InputSettings:"Settings.ini"
Output Chord Length Analysis Results File Path	Yes	The path to the file to which the chord length analysis results will be written (see the Output Files section for more details).  Note that if this file already exists, it will be	/OutputChords:"Chords.txt"

		overwritten by the program.	
Output Blend Analysis Results File Path	Yes	The path to the file to which the blend analysis results will be written (see the Output Files section for more details).  Note that if this file already exists, it will be overwritten by the program.	/OutputBlends:"Blends.txt"
Output Combined Blend Analysis Results File Path	Yes	The path to the file to which the combined blend analysis results will be written (see the Output Files section for more details).  Note that if this file already exists, it will be overwritten by the program.	/OutputCombinedBlends:"CombinedBlends.txt"
Measurement Ball Radius	Yes	A decimal value designating the ball radius used in the measurement of the input data. This value must be greater than zero.  This is assumed to be the same value for all sweep scans.	/MeasurementBallRadius:2.0
Blade Name	Yes	The name of the blade to be written in the output files.	/BladeName:"Blade_01"
Input Nominal	No	The path to the file containing the nominal curve forms to compare against (see the Input Files section for more details).  If a nominals file is given, features are found in the deviations of the measured edges to the given nominals. Otherwise features are found directly in the measured data.	/InputNominal:"Nominal.txt"
Output Nominal (Reference Edge)	No	The path to the file to which nominal curves will be written (see the	/OutputFNominal:"Nominal.txt"

		<p>Output Files section for more details). The nominals written will be the reference edge curves. If this argument is not given, this data is not exported.</p> <p>Note that if this file already exists, it will be overwritten by the program.</p>	
Output Nominal (Edge Actual)	No	<p>The path to the file to which nominal curves will be written (see the Output Files section for more details). The nominals written will be the edge actual curves. If this argument is not given, this data is not exported.</p> <p>Note that if this file already exists, it will be overwritten by the program.</p>	/OutputFANominal:"Nominal.txt"
Output Offset Section Results File	No	<p>The path to the file to which the offset sections of the sweep scan data will be written (see the Output Files section for more details). If this argument is not given, this data is not exported.</p> <p>Note that if this file already exists, it will be overwritten by the program.</p>	/OutputSections:"OffsetSections.txt"
Output Delta Curves File	No	<p>The path to the file to which the delta curves will be written (see the Output Files section for more details). If this argument is not given, then this data is not exported.</p> <p>This argument is only valid when processing with the input nominals argument. If the input nominals argument has not been given,</p>	/OutputDeltaCurves:"Deltas.txt"

		<p>supplying this argument constitutes an error.</p> <p>Note that if this file already exists, it will be overwritten by the program.</p>	
Output Unfiltered Features File	No	<p>The path to which the full set of features prior to filtering (i.e. including those that did not meet the minimum depth parameter) will be written (see the Output Files section for more details). If this argument is not given, this data is not exported.</p> <p>Note that if this file already exists, it will be overwritten by the program.</p>	/OutputUnfilteredFeatures:"AllFeatures.txt"
Message Output Switch	No	<p>A switch argument controlling the output of messages from the application. If 'on', the executable will output messages to standard output and any errors to standard error. If 'off', no messages will be output.</p>	/Verbose
Log File Path	No	<p>The path to the file to log processing information to. If given, all messages/errors will be written to this log file. If not, no log is created.</p> <p>Note that if this file already exists, this is either overwritten or appended to based on the /AppendLog switch described below.</p>	/Log:"BladeRepairLog.txt"
Log File Append Switch	No	<p>A switch argument which can be given in addition to designating the log file path argument. If this is given and the log file already exists, the program will append to this log rather than</p>	/AppendLog

---

		<p>overwrite it. If the /Log command was not given, this has no effect.</p> <p>This can be particularly useful for ensuring that the log output for each blade on a blisk is consolidated into a single file.</p> <p>By default, the switch is off and so the log file will be overwritten.</p>	
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Note that the program does not expect any argument to be given more than once. If multiple values are given then the input arguments are deemed invalid. Commas in argument strings are interpreted as separators between multiple values and as such will be deemed invalid, causing the program to report that the argument “did not designate a single value”.

If the program is given any arguments it does not recognise, this also constitutes an initialisation error and the program will fail. Arguments not of the form indicated above (i.e. not beginning with a forward slash) are also deemed invalid and will report an error.

All command argument parameter names are case sensitive so must be given with the exact casing shown above.

### 3 Return codes and text output

By default, the program will run silently and yield one of the following return codes when the process is complete:

Return code	Brief	Details
0	Success	<p>The program ran successfully with no errors. In this scenario, all output files will have been generated.</p> <p>Note however that this does not mean that all results were in tolerance, only that they were generated correctly.</p>
1	Invalid Licence	<p>The program did not detect a valid licence key and could not run.</p> <p>Note that the program requires a valid MPCs licence on a DESkey dongle to run.</p>
2	Initialisation Error	<p>The program encountered an error during initialisation, which could be due to multiple factors. This meant that it did not attempt to perform any processing. Possible reasons could be:</p> <p>Input command arguments being incorrectly formatted (including casing) or having invalid values.</p> <p>Input files failing to exist or to be opened.</p> <p>Input files failing to be of the correct syntax.</p>
3	Process Error	<p>The program initialised successfully, but then encountered an error during processing.</p> <p>In most circumstances, this is likely to occur as the program fails to write the output files (e.g. they were designated as invalid paths or the program did not have permissions to create the files).</p> <p>However, it is also plausible that this occurs due to an internal error in processing. For example, if the offset sections through sweep scans failed to generate.</p>

## 4 Units

It should be stressed that the blade repair program is effectively “unitless”. All parameters and files given to it are assumed to be in a common unit system. The program will work in this common unit system and any outputs it creates will also be in this common unit system.

It is the responsibility of the user to ensure that all inputs are given in a consistent system.

## 5 Input files

The format of all the required input files is described below. For examples of all these file types, see the samples shipped with the application.

### 5.1 Input sweep scan data files

The input sweep scan file is used to convey the sweep scan measurements of the blade. The format of the file is a MODUS “.out” file.

The specific “.out” file should contain only the sweep scan measurements for the blade of interest. If scans of other parts or any other type of measurement is present in the file, the program is liable to read this data and to try and process it, which could cause the program to fail or adversely affect the results. For example, AES side sweeps and curve scan measurements for the blade must not be included.

When reading the file, the parser scans the file for lines beginning with “FA” indicating a feature actual. After this, any line beginning with a forward slash will be read as a series of 6 comma delimited values, representing the x/y/z coordinates of a point and the i/j/k of its deflection vector, in that order. Note that the deflection vector is not used by the program but is still expected to be present in the file. Points will be attributed to this feature until an “ENDAT” line is reached.

Note that other data in the file will be ignored.

Additionally, it should be stressed that sweep scan data within this file must be reported at tip centre within the expected datum. This is described in more detail in the Blade repair analysis process (Renishaw part number H-1000-5447).

### 5.2 Input tolerances file

The input tolerances file contains information about the tolerance limits and key section positions used in the program. The format is a plain text file.

The file itself is delimited by a series of “block separators”. These are strings surrounded by square brackets (e.g. [Header]). Lines following this block separator are then associated with that particular block. The string inside the square brackets is the block name. These must be matched **exactly** (i.e. case and whitespace sensitive) against the set of expected block names described below.

Blocks can be given in any order (except the Version block which must come first) and can be given multiple times. Each time they are given, all lines following them until the next block separator are read and associated with that particular block.

The set of expected blocks are then as follows:

- [Version] – This is a special block which must be the first block given. It must also contain one and only one data line. This indicates the version number of the file, which currently must be set to either ‘2’ or ‘3’ (the impact of this choice is explained later). Note that old version ‘1’ variants of this file are no longer supported; please see the **Version 1 Upgrade Guide** section for details of how to migrate old files. Future versions of the application may change this version number.
- [Blends] – This specifies a set of tolerances for the blend parameters. Each line in the block is a set of 3 comma separated values, representing “maximum depth”, “leading edge AU” and “trailing edge AU” respectively (see **Blend Table** for details).

- [RAC] & [RAD] – These specify the set of tolerances for the RAC and RAD analysis respectively (i.e. these are two separate blocks, but the information represented follows a common format). Each line in the block is a set of 2 comma separated values, representing “depth” and “min curvature radius” respectively (see **Curvature Table** for details).
- [RACFull] & [RADFull] – These specify the tolerance for the RAC and RAD analysis performed on the full blade edge profiles. Each block must contain one and only one data line. This should be a single value designating the desired tolerance. These values must be greater than or equal to zero, and less than or equal to the smallest RAC/RAD value in their respective curvature tables. Passing these as zero will ensure that the analysis passes if it is not of interest.
- [Chords] – This specifies the set of allowable chord lengths along the blade. Each line in the block is a set of 4 comma separated values, representing “section number”, “AU value”, “CUmin” and “CCmin” respectively (see **Chord Length Table** for details).
- [CumulativeLengths] – This specifies the set of allowable cumulative lengths of all blends found. This is a set of 2 comma separated values, representing “cumulative length per blade” and “cumulative length per blisk” respectively. Note that the program expects these arguments but **does not make use of them at this time**.
- [MaxFeatureAU] – This block must contain one and only one data line. The blade repair instructions allow for chamfer and fillet repairs to be made on the tips of the blades, however these should not be analysed by the application. This block specifies the maximum allowable position a feature is allowed to start at on each edge, and features starting past these values will be omitted from the results (see **Max Feature AU Values** for details).
- [MinExpectedEdgeValues] – This block must contain one and only one data line. It specifies four comma delimited values designating the minimum values the edges are expected to have at their start/end points (see **Min Expected Edge Values** for details).

If any block is given which is not recognised, the file is deemed invalid.

Lastly, any line beginning with a semicolon is treated as a comment and is ignored. All lines are set to ignore trailing whitespace, and data lines are set to ignore leading whitespace. Lines which are entirely empty or entirely whitespace are also ignored.

### 5.2.1 Blend table

The [Blends] block governs the allowable blend depths for any blends found. This maximum depth is dependent upon the AU position of the blend (i.e. its Z value). Each row in the table represents an “interval”, defining the maximum of that interval, with the minimum being the last maximum (or zero if it is the first entry).

This is best considered by an example. Consider the table below:

```
[Blends]
;MaxDepth, LE_AU, TE_AU
0, 190.2, 196.3
0.7, 197.7, 203.8
1.2, 220.4, 220.4
2.0, 250, 250
```

This table implies that for a leading edge, the ranges are:

```
0 < Z <= 190.2: Max Depth = 0
190.2 < Z <= 197.7: Max Depth = 0.7
197.7 < Z <= 220.4: Max Depth = 1.2
220.4 < Z <= 250.0: Max Depth = 2.0
```

For the trailing edge, the ranges are then:

```
0 < Z <= 196.3: Max Depth = 0
196.3 < Z <= 203.8: Max Depth = 0.7
203.8 < Z <= 220.4: Max Depth = 1.2
220.4 < Z <= 250.0: Max Depth = 2.0
```

For each blend found, the program will find the intervals that the blend spans. If it belongs entirely in one interval, its maximum depth is just the value specified by this interval. If it spans multiple intervals, its maximum depth is the minimum depth value across any interval it spans.

The user should ensure that the maximum values designated in the last entry are greater than any Z position in the processing range (defined by the extreme section positions). If any interval is invalid (i.e. the interval end values are not strictly increasing), the file is deemed invalid.

Whilst the example table above indicates 4 intervals, it can designate any number of intervals in practice. Additionally, whilst the maximum depth values are shown as increasing, this is not required.

### 5.2.2 Curvature table

The curvature tables defined by the [RAC] & [RAD] blocks define the curvature limits for the RAC and RAD analysis. The tables are formatted the same in both cases. Like the blend depth table, these tables denote a series of intervals. For any blend found, the program determines its maximum depth. It then finds the interval this depth lies within, and uses that to determine the allowable radius of curvature for the blend.

An example table is given below:

```
[RAC]
;Depth,MinRAC
0.5,4
1,6
2,10
3,16
```

Each row in the table represents an “interval”, defining the maximum of that interval, with the minimum being the last maximum (or zero if it is the first entry). As such, this table should be read as follows:

```
0 < Depth <= 0.5: Minimum Radius of Curvature = 4
0.5 < Depth <= 1.0: Minimum Radius of Curvature = 6
1.0 < Depth <= 2.0: Minimum Radius of Curvature = 10
2.0 < Depth <= 3.0: Minimum Radius of Curvature = 16
```

The RAC and RAD analysis will then be performed using the determined minimum radius of curvature.

Note that the last entry’s maximum depth is effectively ignored (i.e. if we have a blend with a depth greater than this last maximum, it will perform the analysis with the last radius specified in the table). It should be stressed that it is expected that the maximum blend depth criteria will cause any deeper blend to report a failure in such a scenario anyway, meaning the RAC/RAD analysis is less critical.

If any interval is invalid (i.e. the interval end values are not strictly increasing), the file is deemed invalid. Whilst the example table above indicates 4 intervals, it can designate any number of intervals in practice. Additionally, whilst the radius values are shown increasing, this is not required.

Curvature analysis is also performed on the full blade using the values from the [RACFull] and [RADFull] sections. These values must be less than or equal the smallest value from their respective table (e.g. in the example above, the full RAC value must be less than or equal to 4).

### 5.2.3 Chord length table

The chord length tolerance table specifies restrictions on the allowable chord lengths across the entirety of the measurement.

The table defines a series of “key positions” in Z. Each position then has an “uncountable” and “countable” minimum (CU and CC value respectively). For Z values not at these key positions, the uncountable/countable minima are then interpolated based on their neighbouring key positions. These chord length limits are used to assess validity of identified blends. The Blade repair analysis process (Renishaw part number H-1000-5447) details how these are used.

An example table is given below:

```
[Chords]
;Section number,AU,CUmin,CCmin
0,195,34.3,33.6
1,198,34.3,33.6
2,201,34.7,34
3,204,35.1,34.4
4,207,35.4,34.4
5,210,35.8,34.8
6,213,36.1,35.1
7,216,36.4,35.4
8,219,36.7,35.7
9,222,37,35.5
10,225,37.3,35.8
11,228,37.5,36
12,231,37.7,36.2
13,232,37.7,36.2
0,235,37.7,36.2
```

The values on each line are in order the “section number”, “AU value”, “CUmin” and “CCmin”.

The AU values must be given in ascending order, and for each line its CU minimum must be greater than the CC minimum. If either of these criteria do not hold, the file is deemed invalid. Section numbers, however, only matter as being either zero or non-zero (i.e. they do not have to be well ordered).

The program will attempt to analyse the data between the first and last AU values designated in the table (e.g. 195 and 235 above). Inside this range, it will attempt to use MPCS to generate offset sections through the sweep scans to achieve tip compensated results. As such, sufficient coverage of the sweep scans must be present to generate a correct offset within the range designated. If the data does not have the required coverage, the offsets in that region may fail outright, and if this occurs the program will fail reporting a processing error. However, missing coverage may also distort the sections without causing them to fail, which would then be present in the results. As such, the user should ensure that sufficient coverage of the data exists over this range.

The chord result output file will report the chord length at each section with a non-zero section number. Values which have zero section numbers will influence the CU and CC curves and will cause the program to attempt to process between these values, but will not report as key sections in the final chord analysis output file.

Zero section numbers should be used to designate the range to process on or to influence the CU/CC curves, but where the user is not interested in the exact chord length at that particular value.

### 5.2.4 Max feature AU values

The blade repair instructions allow for chamfer and fillet repairs to be made on the tips of the blades, however these should not be analysed by the application. The intent of this section is to designate a maximum allowable feature start position, such that features starting after this position (which are likely to be chamfers or fillets) are discarded.

The content of this block differs depending on the version of the file. In either case, the section must specify one and only one line. In a version '2' file, this line must be a single numeric value, which is the maximum value used for both edges. A version '3' file allows a single value to be used (in the same manner as a version '2' file), but also supports the ability to designate different values per edge by giving two comma separated numeric values. If two values are given, then the first is used for the limit on the leading edge, and the second is used for the limit on the trailing edge.

Note that there are no validation checks on these parameters beyond ensuring they are numeric. If a limit is given which is greater than the maximum AU value of the data, then it will never cause features to be discarded. Conversely, if the user gave a value less than the minimum AU value of the data, then all features would be discarded.

### 5.2.5 Min expected edge values

This section must specify one and only one line. The content of this line is expected to be four comma delimited numeric values. Each value represents the minimum value expected in the edge actual curves generated by the processing for the start and end of the leading and trailing edge curves.

The order is: Leading Edge Start, Leading Edge End, Trailing Edge Start, Trailing Edge End.

These parameters are used when applying the main filter to produce the reference edge curves, and influence generation of the reference edge at the start/end of the data. This is described in more detail in the Blade repair analysis process (Renishaw part number H-1000-5447).

## 5.3 Input application settings file

The input application settings file contains various additional settings which govern the behaviour of the application. This is structured in a similar manner to the tolerance file in that it is separated by a series of "block separators" surrounded in square brackets (e.g. [Header]). The line that follows a block separator then denotes the value associated with that block. Unlike the tolerance table however, each block is associated with one and only one value. If a block appears more than once or tries to designate multiple values (i.e. has multiple associated value lines), the file will be deemed invalid. All blocks are mandatory in this file (i.e. if any are missing, the file is deemed invalid). Bar the version block (which must come first), the remaining blocks can be given in any order.

The list of block names and their brief descriptions are as follows (for detailed descriptions, please see the Blade repair analysis process (Renishaw part number H-1000-5447)):

- [Version] – This is a special block which must be the first block given. This indicates the version number of the file, which currently must be set to '2'. Note that old version '1' variants of this file are no longer supported; please see the **Version 1 Upgrade Guide** section for details of how to migrate old files. Future versions of the application may change this version number.
- [FilterRadius] – Defines the radius used in the main ball filter used to construct the reference edge curves. This must be greater than zero.

- 
- [FeatureCutOffDepth] – Defines the tolerance used to determine where blends begin and end. This must be greater than or equal to zero.
  - [MinFeatureDepth] – Defines the minimum depth of a feature for it to be considered in the analysis. This must be greater than or equal to zero. If this is given as less than the cut off depth, it has no effect.
  - [FeaturePaddingLength] – When a blend is found and output to the results file, this output will contain various graphs for the blend. The range of the graph will be the range of the blend, plus this additional length either side (where the data exists). Note however that any “padded” length added to the graph in this way is for visualisation purposes only and is NOT considered during any tolerance evaluation. This must be greater than or equal to zero.
  - [FeatureCurvatureAnalysisTolerance] – When the program performs the RAC and RAD analysis on a feature, this determines the maximum difference allowed between the original and filtered set before it is deemed to violate the allowed curvature. This must be greater than or equal to zero.
  - [FullCurvatureAnalysisTolerance] – When the program performs the RAC and RAD analysis on the full edges, this determines the maximum difference allowed between the original and filtered set before it is deemed to violate the allowed curvature. This must be greater than or equal to zero.
  - [PointSpacing] – Initial points are computed at the section plane spacing (described below), but are linearly infilled to this required point spacing for the feature analysis. This is required as all filtering is performed on discrete points, and the section spacing will generally be too coarse.
  - [CurveDownSamplingFactor] – An integer value which determines how curves are downsampled from the point spacing value described above in the output files. Effectively, for this number of points in the internal set, the output set will contain a single point. The other way to think about this is that the point spacing in all curves in the output files will be equal to the point spacing above multiplied by this value. This must be greater than zero.
  - [OutputDecimalPlaces] – An integer value representing the number of decimal places to write out numeric values to in the output files. This must be greater than or equal to zero.
  - [SectionSpacing] – The program initially performs tip compensation by generating a series of offset sections at specific Z values through the sweep scan data. This parameter determines the distance in Z between successive planes. Lower values will lead to more accurate results but will take more time to run. This must be an exact multiple of the point spacing parameter. The suggested value is 100 microns.
  - [SectionMinEdgeLength] – The MPCS parameter determining the minimum edge length to use when generating offset sections. The suggested value is 50 microns.
  - [SectionMaxEdgeLength] – The MPCS parameter determining the maximum edge length to use when generating offset sections. The suggested value is 25 millimetres.
  - [SectionMinAngleDegrees] – The MPCS parameter determining the minimum angle (in degrees) to use when generating offset sections. The suggested value is 178 degrees.
  - [SectionMaxProjectionError] – The MPCS parameter determining the maximum projection error when generating offset sections. The suggested value is 2.5 microns.

- [ChordDirection] – This value controls how the program determines which is the leading and which is the trailing edge from the measurement data. This value must be set to either 0 or 1. If this value is changed from one to the other, the program will swap what it considers to be the leading and trailing edge. The way this is determined is described in more detail in the Blade repair analysis process (Renishaw part number H-1000-5447).

Lastly, any line beginning with a semicolon is treated as a comment and is ignored. All lines are set to ignore trailing whitespace, and data lines are set to ignore leading whitespace. Lines which are entirely empty or entirely whitespace are also ignored.

## 5.4 Input nominal file

The input nominal file is an optional input. If this file is given, features are found in the deviations of the measured edges to the given nominals. If this file is not given, features are found directly in the measured data. This is discussed in more detail in the Blade repair analysis process (Renishaw part number H-1000-5447).

The file is structured in a similar manner to the tolerance/settings files in that it consists of “block separators” surrounded by square brackets (e.g. [Header]) which delimit particular blocks of data. The line that follows a block separator then denotes the value associated with that block. Unlike the other files, this contains only a very limited number of blocks.

The file must first begin with a [Version] block. This will then contain a single value indicating the version number of the file. This must be set to ‘1’ in the current version of the program. Future versions of the application may change this version number.

The file then contains two further blocks which can be given in any order, which are [LeadingEdgeNominal] and [TrailingEdgeNominal]. Each block must then contain one and only one DMIS curve, which is interpreted as the expected nominal curve form for that particular edge. If either block is not present, does not designate a curve or attempts to designate more than one curve, the program will treat this as an invalid nominal file and will fail to process. The general format of this curve is described in the **DMIS Curves** section, but the following should be noted:

- The input curve can be designated as either F or FA. The choice has no effect on the processing.
- The name of the curve (i.e. content between the brackets following F/FA) is not used.
- The six values representing the plane of the curve must be present and must be valid numbers, but their exact values are not used.
- All y values in the curve must be explicitly given as zero.
- All z values in the curve must be strictly increasing (i.e. each z value must be greater than the preceding z value).
- The three values representing the normal for each point must be present and must be valid numbers, but their exact values are not used.

If the rules above do not hold, then the nominal file will be deemed invalid and the application will fail to process. The nominal curve must also meet the criteria in that it fully encapsulates the requested AU analysis range of the program (i.e. the nominal exists for each position in the analysis range).

The Z values are then assumed to represent AU values, and the X values are the distance of the edge away from the stacking axis in terms of chord lengths.

Nominal files can be produced by the program via the “/OuptutFNominal” and “/OutputFANominal” arguments. These are described in more detail in the **Output Files** section. However, a user can also generate this file themselves outside the program based on whatever is believed to constitute a suitable nominal.

Similar to the other input files, any line beginning with a semicolon is treated as a comment and is ignored. All lines are set to ignore trailing whitespace, and data lines are set to ignore leading whitespace. Lines which are entirely empty or entirely whitespace are also ignored.

## 6 Output files

The format of all the files output by the application is described below.

Note that curves output to any of these files are written in a DMIS format (which is described in its own section). As part of this, the curve is given a name detailed below. In all cases, these names are prefixed with the blade name given as a command argument (shown as <BladeName>).

### 6.1 Output chord analysis file

This output file records the chord length analysis of the part. It is written out as a plaintext file separated into a series of blocks preceded by block separator lines. The blocks are given in the following order:

- [Version] – The version number of the output file. This is currently set to '2', though this may change in later versions of the program if the form of the output file changes.
- [Feature Curves] – This is a series of 4 curves in DMIS format. In order, these curves are:
  - The “nominal” set of chord lengths. This is a dummy curve that is the mean of the CU and CC curves, required to report correctly in MODUS. The name is set to “<BladeName>\_ChordCurve”. This is written as a nominal curve (i.e. F prefix).
  - The “actual” set of chord lengths. These are computed chord lengths against section height for the blade. The name is set to “<BladeName>\_ChordCurve”. This is written as an actual curve (i.e. FA prefix).
  - The interpolated CUmin curve (this is at the same spacing as the other curves). The name is set to “<BladeName>\_CU\_MIN”. This is written as a nominal curve (i.e. F prefix).
  - The interpolated CCmin curve (this is at the same spacing as the other curves). The name is set to “<BladeName>\_CC\_MIN”. This is written as a nominal curve (i.e. F prefix).
- [Chord Sections] – A series of entries, one line per section demanded in the chord section of the input tolerances file (noting that sections with zero section numbers are ignored). Each entry is then a decimal value indicating the chord length at that section. These are given in the same order the sections were given (i.e. ascending Z order).
- [Minimum Chord Length] – Describes the location where the chord length minus the CCmin curve was minimised. This is a set of four entries (one per line), which are:
  - The chord length at this position.
  - The CUmin value at this position.
  - The CCmin value at this position.
  - The AU position (Z value) this occurred at.
- [Maximum Chord Length] - Describes the location where the chord length minus the CCmin curve was maximised. This again is a set of four values as used to in the [Minimum Chord Length] block above.
- [Countable Length] – Describes the length of the curve which was deemed countable (i.e. chord lengths were less than the CU curve).

- [Violating Length] – Describes the length of the curve which was deemed violating (i.e. curve lengths were less than the CC curve).

## 6.2 Output blend analysis file

This output file records the details of any blends found across both the leading and trailing edges. The file first contains a “header” which describes overall information. After that, it then will contain repeating blocks for each blend found. As with other files, this is delimited by a series of block separators surrounded by square brackets, followed by the data for that block. Where a block contains multiple values, they are written out one per line.

Note that where names of DMIS curves are given, <Edge> is either “LE” or “TE” (for leading/trailing edges) and <BlendName> is the name of the blend.

The initial header consists of the following blocks:

- [Version] - The version number of the output file. This is currently set to ‘2’, though this may change in later versions of the program if the form of the output file changes.
- [Number of Blends] – The total number of blends found across both edges.
- [Number of LE Blends] – The number of blends found on the leading edge.
- [Number of TE Blends] – The number of blends found on the trailing edge.
- [Complete 2D Leading Edge Profile] / [Complete 2D Trailing Edge Profile] – Two blocks given in this order. Each consisting of four DMIS curves associated with either the leading or trailing edge, as follows:
  - The “nominal” edge, which is the “reference edge curve” described in the Blade repair analysis process (Renishaw part number H-1000-5447). This is named “<BladeName>\_<Edge>\_2D\_Profile”. This is written as a nominal curve (i.e. F prefix).
  - The “actual” edge, which is the “edge actual curve” described in the Blade repair analysis process (Renishaw part number H-1000-5447). This is named “<BladeName>\_<Edge>\_2D\_Profile”. This is written as an actual curve (i.e. FA prefix).
  - The RAD curvature reference curve for the entire edge. This is named “<BladeName>\_<Edge>\_FullRADCurve”. This is written as a nominal curve (i.e. F prefix).
  - The RAC curvature reference curve for the entire edge. This is named “<BladeName>\_<Edge>\_FullRACCurve”. This is written as a nominal curve (i.e. F prefix).
- [Cumulative Countable LE Blend Length] – The total countable length of all blends across the leading edge.
- [Cumulative Countable TE Blend Length] – The total countable length of all blends across the trailing edge.
- [Full Leading Edge RAD/RAC Validity] - The Boolean results denoting whether or not the RAD/RAC checks (in order) for the full leading edge passed. These are written as DMIS style Boolean values (“.TRUE.” or “.FALSE.”).
- [Full Trailing Edge RAD/RAC Validity] - The Boolean results denoting whether or not the RAD/RAC checks (in order) for the full leading edge passed. These are written as DMIS style Boolean values (“.TRUE.” or “.FALSE.”).

If no blends were found on either edge, this will be the end of the file. Otherwise, the set of data below is then given for each blend found. Blends on the leading edge are given first, in ascending Z order, followed by the blends on the trailing edge, again in ascending Z order.

- [BlendName] – The name of the blend. This name is used in the names of the DMIS curves and in the combined blend file (described later). They are always named “Blend\_N” where N is the index of that blend on its respective edge, starting at 1. As such if blends are found on both edges there will be a Blend\_1 for both the leading and trailing edge.
- [Edge] – The entry is either “LE” or “TE” to denote whether the blend is on the leading or trailing edge respectively.
- [Profiles] – A set of 4 DMIS curves representing the blend. These are given in the range of the blend plus the additional length designated in the application settings file (where data is present). These are as follows:
  - The “nominal” blend (i.e. the sub set of the full reference edge curve output in the header section for the current blend). This is named “<BladeName>\_<Edge>\_<BlendName>\_Profile”. This is written as a nominal curve (i.e. F prefix).
  - The “actual” blend (i.e. the sub set of the full edge actual curve output in the header section for the current blend). This is named “<BladeName>\_<Edge>\_<BlendName>\_Profile”. This is written as an actual curve (i.e. FA prefix).
  - The RAD curvature reference curve for the blend. This is named “<BladeName>\_<Edge>\_<BlendName>\_RAD\_Profile”. This is written as a nominal curve (i.e. F prefix).
  - The RAC curvature reference curve for the blend. This is named “<BladeName>\_<Edge>\_<BlendName>\_RAC\_Profile”. This is written as a nominal curve (i.e. F prefix).
- [BlendRange] – This is a set of two values, which designate the AU values where the blend begins and ends respectively.
- [Countable Blend Length] – The length of the blend within this range which was deemed “countable”.
- [Maximum Depth Info] – The information regarding the position where the depth of the blend is maximised. This is given as three values, namely: the maximum depth, the AU position where it occurs, and the chord length at this location respectively.
- [Minimum Chord Info] – The information regarding the position where the chord length minus the CCmin value is minimised. This is given as three values, namely: the depth at this position, the AU position where it occurs, and the chord length at this position respectively.
- [RAD/RAC Validity] – The Boolean results denoting whether or not the RAD/RAC checks (in order) passed. These are written as DMIS style Boolean values (“.TRUE.” or “.FALSE.”).

- [Evaluation Tolerances] – This is then the set of tolerances which were used when analysing the feature. This consists of 7 values in the following order:
  - The maximum allowable depth of the feature (based on the intervals that its range spans).
  - The RAD filter radius (based on the maximum depth of the feature).
  - The RAC filter radius (based on the maximum depth of the feature).
  - The CCmin value at the maximum depth location.
  - The CUmin value at the maximum depth location.
  - The CCmin value at the minimum chord location.
  - The CUmin value at the minimum chord location.

### 6.3 Output combined blend analysis file

This is a simple file which stores details of any “combined” blends found. These occur where blends exist in both the leading and trailing edges with overlapping ranges. In this situation, the program also verifies the maximum “combined” depth of the two blends.

As with the main blend file, this initially begins with a header containing data delimited by “block separators” surrounded by square brackets, as follows:

- [Version] - The version number of the output file. This is currently set to ‘1’, though this may change in later versions of the program if the form of the output file changes.
- [Number of Combined Blends] – The total number of combined blends found.

If there are no combined blends, this is the end of the file (which is likely to be the case in the majority of scenarios). Otherwise, we then have a series of repeating blocks beginning with block separator [Combined Blend N] (where N is the number of the paired blend, starting at 1). This then contains a series of 10 entries as follows:

- The name of the blend in the leading edge (from the main blend analysis file).
- The name of the blend in the trailing edge (from the main blend analysis file).
- The AU value where the blends begin overlapping.
- The AU value where the blends stop overlapping.
- The AU value where the combined depth (i.e. the sum of the depths of the two blends) is maximised.
- The depth of the leading edge blend at this combined maximum depth position.
- The depth of the trailing edge blend at this combined maximum depth position.
- The maximum combined depth.
- The maximum allowable depth.
- A Boolean result indicating whether the feature was within allowable depth or not (“.TRUE.” for a pass, “.FALSE.” for a failure).

The maximum allowable depth is determined as the minimum of the maximum allowable depths for the two individual blends. As such this is the minimum of the maximum allowable blends of the entire range of the two features, not just the common range.

## 6.4 Output offset sections

This file is an optional output, which will be written out only if the “/OutputSections” command is given. This file contains the full MPCs sections generated by the program from the sweep measurement data.

The file is a simple text file. As with other files, this is delimited by a series of block separators surrounded by square brackets. This has the following format:

- [Version] - The version number of the output file. This is currently set to ‘1’, though this may change in later versions of the program if the form of the output file changes.
- [Num Sections] - The number of offset sections stored in the file.
- The remainder of the file is then a series of [Section] blocks. After a section delimiter, the lines following it are then the points in that section. Each point is a set of three comma delimited values representing X, Y and Z coordinates respectively. Note that within a section, all points will share the same Z value (as they are sections through XY planes). Once another [Section] delimiter is found, the current section ends and any subsequent points belong to the next section. Sections are written out in ascending Z order.

If a section fails to generate (as there is not sufficient coverage in the data to generate the offset), that section will contain zero points. This effectively means that the [Section] delimiter will be immediately followed in the next line by another [Section] delimiter. A section failure is deemed a process error and the application will terminate without performing feature/chord analysis.

This data is not necessary for reporting in MODUS, hence why it is an optional output. However, it may be of interest if any other statistics about the form of any blade sections are required.

## 6.5 Output nominals

This file is an optional output, which will be written out only when either the “/OutputFNominal” or “/OutputFANominal” commands are given. This file stores the data from the current process such that it can be passed to a subsequent run of the process via the “/InputNominal” command.

The file is a simple text file. As with the other file formats described earlier, it consists of block separators surrounded by square brackets which delimit fields. This has the following format:

- [Version] - The version number of the output file. This is currently set to ‘1’, though this may change in later versions of the program if the form of the output file changes.
- [LeadingEdgeNominal] – A DMIS curve representing the nominal leading edge.
- [TrailingEdgeNominal] – A DMIS curve representing the nominal trailing edge.

The “/OutputFNominal” switch will write the curves out which match the reference edge curves written out to the blends file. The “/OutputFANominal” switch will write out the curves which match the edge actual curves written out to the blends file. In either case the curves will match exactly (i.e. same name, same values, same F/FA designation).

The application can take neither, one or both of these switches. Note that if both are given then they should designate different file paths (otherwise one file will overwrite the other).

## 6.6 Output deltas

This file is an optional output, which will be written out only if the “/OutputDeltas” command is used.

This file is a simple text file. It is designed to provide an optional output when the program is run with the optional input nominals file. In this mode the process operates using a “delta space” which consists of the difference between the measurements of the given curve against the given nominal. Results however are returned in standard space.

This file however writes out the reference edge and edge actual curves when considered in delta space (i.e. differences against the nominal). This is a simple file consisting of block separators surrounded by square brackets which delimit fields. This has the following format:

- [Version] – The version number of the output file. This is currently set to ‘1’, though this may change in later versions of the program if the form of the output file changes.
- [Leading Edge Delta Space Profile] – This section contains two DMIS curves representing the delta curves for the leading edge. The first curve is the reference edge curve when considered in delta space, the second curve is the edge actual curve when considered in delta space. Both curves will have the same name, with the former written out as a nominal (F prefix) and the latter written out as an actual (FA prefix).
- [Trailing Edge Delta Space Profile] – This section contains two DMIS curves representing the delta curves for the trailing edge. The first curve is the reference edge curve when considered in delta space, the second curve is the edge actual curve when considered in delta space. Both curves will have the same name, with the former written out as a nominal (F prefix) and the latter written out as an actual (FA prefix).

## 6.7 Output unfiltered features

This file is an optional output, which will be written out only if the “/OutputUnfilteredFeatures” command is used.

In standard processing, blends/features are identified when the actual curves differ from the filtered curves by more than the “FeatureCutOffDepth” specified in the input settings file. From these, the application then determines a “depth” for each feature. Features are then only analysed if this depth exceeds the “MinFeatureDepth” specified in the input settings file.

This means though that there may be “small features” which are dismissed by the application as they are below the minimum depth. This optional output provides basic information on all features found (both those which were kept and those which were discarded) such that a user can identify whether or not there is a feature which is still being picked up but may be smaller than anticipated.

The file is a simple text file. As with the other file formats described earlier it consists of block separators surrounded by square brackets which delimit fields. This has the following format:

- [Version] - The version number of the output file. This is currently set to ‘1’, though this may change in later versions of the program if the form of the output file changes.
- [LeadingEdgeFeatureCount] – The number of unfiltered features on the leading edge.
- [TrailingEdgeFeatureCount] – The number of unfiltered features on the trailing edge.
- [LeadingEdgeFeatureDetails] – The details of all unfiltered features on the leading edge. The number of lines in this block will be equal to the number of features in the leading edge.

- [TrailingEdgeFeatureDetails] – The details of all unfiltered features on the trailing edge. The number of lines in this block will be equal to the number of features in the trailing edge.

Each feature (in either the leading or trailing edge details block) is represented by a single line of four comma delimited values. These values represent in order:

- The AU position at which the feature starts.
- The AU position at which the feature ends.
- The maximum depth of the feature.
- The AU position at which this maximum depth is achieved.

Note that features discarded by the application based on the maximum feature AU values will also be included in the unfiltered features file.

---

## 7 DMIS curves

Several different output files write out curve outputs in DMIS format. Additionally, the input nominals file also expects input curves in DMIS format. An example of this format is given below:

```
F(Blade_ChordCurve)=FEAT/GCURVE, CART, 0, 0, 0, 0, 1, 0, PTDATA, 33.950, 0, 198.000, 1, 0, 0, 33.957, 0, 198.050, 1, 0, 0, ...
```

The format of this is as follows:

- The line begins with “F” for a nominal feature, or “FA” for an actual feature.
- The name of the curve resides within the brackets (i.e. “Blade\_ChordCurve” above).
- The “0, 0, 0, 0, 1, 0” above defines the plane the curve resides in (x, y, z, i, j, k). These numbers are used for all curves written out in the program, as curves are all planar and output in the ZX plane.
- The curve is then described by a series of points in the form “x, y, z, i, j, k”. All y values are set to zero. The Z values represent AU values whereas X values represent the data of interest at that AU value. All y values are written as zero and all normals are set to positive X (1, 0, 0).

The input nominal file is the only place where curves of this format are read back in. Particular notes around this are described in the **Input Nominal File** section.

## 8 Guide to upgrading from version 1

This section is designed to offer guidance on how to update files which were written for the original version of the application to work for the current version. Input files have changed versions and the application now rejects old files. Certain output files also contain more information and write out their data differently.

This section concerns itself with the primary differences, but does not go into detail in describing the meaning of any new input/output parameters. For more details, see either the **Input Files** or **Output Files** sections.

This section does not concern itself with technical differences between the applications. Different versions may yield different numerical results.

### 8.1 Input file changes

Both the input settings and tolerance files have been updated from version 1 to version 2. Any old file indicating version 1 will be rejected as invalid.

Upgrading a version 1 settings file to version 2 requires the following changes:

- Change the number under the [Version] heading from '1' to '2'.
- Rename the [CurvatureAnalysisTolerance] block to [FeatureCurvatureAnalysisTolerance].
- Add the new [FullCurvatureAnalysisTolerance] block and designate a value. A suggested initial value would be the same value as the feature curvature analysis tolerance.

Upgrading a version 1 tolerance file to version 2 requires the following changes:

- Change the number under the [Version] heading from '1' to '2'.
- Add in the new [RACFull] and [RADFull] blocks and designate a value for each.

If these changes are made, then the program should then run.

Note that the tolerance file can also be updated to version '3'. However, the only difference of this to a version '2' file is that the version '3' file allows a "maximum feature AU" parameter to be set differently for the two edges (see the tolerance file section for more details).

---

## 8.2 Output file changes

The chord and blend analysis files will have changed as a result of running the newer application.

The chord analysis file will have changed as follows:

- The version number recorded under the [Version] heading will now be '2'.
- The end of the file then contains two new sections, [Countable Length] and [Violating Length].

The blend analysis file will have changed as follows:

- The version number recorded under the [Version] heading will now be '2'.
- The [Complete 2D Leading Edge Profile] and [Complete 2D Trailing Edge Profile] blocks now contain four curves each rather than two (the latter two are the full RAD/RAC analysis curves).
- After the [Cumulative Countable LE Blend Length] block, there are two new blocks – [Full Leading Edge RAD/RAC Validity] and [Full Trailing Edge RAD/RAC Validity].

The other output files (combined blends, optional section outputs) are unchanged.

The program is also liable to write out different messages to the console and the log file. These are not recorded in detail as this is designed as a human-readable log rather than something that contains meaningful data which users may be interested in parsing.

## 9 Troubleshooting

The following table gives a series of possible issues that can arise from the application and suggested solutions for each. If these steps fail to resolve the issue, please contact Renishaw.

Problem	Solution
The program is returning an error code.	Turn on diagnostics via the /Verbose or /Log commands for extra details of why this is occurring to try and pinpoint the cause of the error (if these switches are not given, the program will run silently).
The /Log command was designated, but the program is still returning an error code and not creating a log.	If /Log is designated and the log file cannot be created, this constitutes an initialisation failure. Turn /Verbose on and see if this gives more detailed information.
The program reports that the licence is invalid.	Ensure a dongle with a valid MPCS licence is present.
The program reports that invalid or unrecognised command line arguments were supplied.	<p>Valid arguments are of the form "/Argument:Value". As such, any command line argument given which is not of this form (e.g. missing the forward slash) is treated as invalid. Similarly, any argument that does not match those expected by the program is deemed unrecognised. Note that these arguments are case sensitive.</p> <p>Ensure that the arguments given are valid and correctly formatted.</p>
The program reports a file could not be opened.	<p>This implies the program parsed all arguments successfully, but when it tried to then read a particular input file, it failed to open it. This could be because the file does not exist, the program does not have permission to open the file, or the file path designated was invalid.</p> <p>The program can support both relative and absolute file paths. If the path is given as a relative path, care should be taken to ensure the working directory is as expected.</p> <p>Ensure that the files exist with valid file paths and the program has permissions to access them.</p>
The program reports that a file was incorrectly formatted.	<p>This implies the program was parsing this particular file and encountered a syntax error which meant it deemed the file invalid (this may also include a parameter being invalid or inconsistent). The error message in question should indicate which line of the file the problem occurred at and why it was invalid.</p> <p>Note that if the file contains multiple syntax errors, the program will only recognise the first error before aborting.</p> <p>Ensure that the file in question is formatted as described in the Input Files section.</p>
The program reports a process error.	This return code implies the program successfully parsed all input arguments and read all input files, but a problem then occurred which prevented it from fully completing the process.

	<p>The first potential cause of that MPCS sectioning failed due to bad data. The program requires that all scans in the input file are valid sweep scans, and that these give coverage in the required range. If any section fails, the program reports this as an error and proceeds no further with the analysis. If this is the case, the "/OutputSections" command can be used to write out sections to determine which sections are failing.</p> <p>The other potential cause of this is that it failed to write out the output files. This could occur if any output file path designated is not a valid file path, or if the program does not have permissions to write to this location. In this scenario, ensure that the paths are valid.</p> <p>Either way, the return code should be accompanied by a message attempting to indicate what the problem is. If the nature of the problem cannot be determined from the message, please contact Renishaw.</p>
<p>The application processes without reporting an error, but the data has unexpected features.</p>	<p>The application expects the sweep measurement data to give full coverage over the requested range. In some scenarios, it is possible for the MPCS sections to succeed, but give malformed outputs due to lack of coverage. This would likely manifest itself in exaggerated chord lengths in the location of missing coverage.</p> <p>In this scenario the part program should be modified to increase coverage of the part. If this is not possible, adjust the analysis range designated in the chord tolerance table to avoid the area of missing coverage</p>

**Renishaw plc**  
New Mills, Wotton-under-Edge,  
Gloucestershire, GL12 8JR  
United Kingdom

**T** +44 (0)1453 524524  
**F** +44 (0)1453 524901  
**E** [uk@renishaw.com](mailto:uk@renishaw.com)  
[www.renishaw.com](http://www.renishaw.com)

**RENISHAW**   
**apply innovation™**

**For worldwide contact details,  
please visit our main web site at  
[www.renishaw.com/contact](http://www.renishaw.com/contact)**



H - 1000 - 5448 - 01