

OpenCNC 6.6 Integrator Release Notes

What is Covered in this Document

This version of the Integrator Release Notes describes changes made to OpenCNC from version 6.5 to 6.6. Previous Integrator Release Notes for OpenCNC can be found in the MDSI Documentation.

OpenCNC Product Descriptions

OpenCNC Pro - This version supports 5-10 axes of simultaneous motion. The United States government prohibits export of this product to any country that is not on the approved export list for controlled technology. Violating this law can result in large fines and imprisonment. To ensure compliance with this law, you must obtain prior written approval from MDSI before you can ship this product outside of the United States or Canada. This product cannot be exported to Mexico without first obtaining an Export License.

OpenCNC Plus - This product is limited to four axes of simultaneous motion. Unlike OpenCNC Pro, you may export this product to most, but not all, countries in the world. However, to assure compliance you must obtain prior written approval from MDSI before you can ship this product outside of the United States or Canada.

OpenCNC Demo - This is the demonstration version of OpenCNC. It has the following features and limitations.

- Demo version runs on Windows XP, 2000, NT, ME, and 98 Second Edition.
- This product does not require Ardence RTX or an OpenCNC dongle to run.
- Maximum interrupts are 100 per second.
- Does not support I/O or hardware interrupts.

Operating Systems and Minimum Requirements

- Windows 2000 Service Pack 4 or Windows XP Service Pack 2

NOTE: 

Windows NT is no longer supported for OpenCNC.

- 256 megabytes RAM (512 megabytes recommended). Memory chips in the PC must all be the same size (i.e. do not use a 256 MB chip and a 128 MB chip in the same PC). MDSI highly recommends that you only use memory chips from the same vendor. During Windows 2000 testing we discovered several OpenCNC startup problems that were directly related to the memory in the PC.
- Intel Pentium Processor - 300 MHz or above including Celeron, and PII, PIII, and P4 processors (recommended 1 GHz PIII).
- Intel manufactured motherboard.
- Disk space of at least 5 gigabytes of free disk space for Windows 2000 or Windows XP.
- CD-ROM Drive.

Users upgrading an OpenCNC system from Windows NT to Windows 2000 or Windows XP have three options. Only the first is totally supported by MDSI.

- The recommended option is to add a new hard drive to the computer and do a clean install of Windows 2000 or Windows XP on the new hard drive.
- The second option is to install Windows 2000 or Windows XP on a blank partition of the current hard drive. This option is discouraged since having an NT and 2000 partition on a single hard drive is not fully supported by Microsoft and MDSI did experience some failures during installation testing.
- Upgrading a partition from NT to 2000 or XP is not supported by MDSI due to the inconsistencies in the Windows directory structure after the upgrade has occurred.

New Features

The following new features have been added to OpenCNC 6.6.

Lathe Finish Macro

Added S0 for the macro to output to tool tip positions with no offset or tool angles applied.

The S parameter has been changed as follows:

Parameter	Description
S0	No cutter compensation will be used. The purpose of S0 is to output the tool tip positions without offsets or tool angles applied. The part program may be using the output of a post processor that has already calculated TNR and oriented values into the programmed data. With S0, first corner chamfer or rounds, UVW, tool leading/trailing angles, and TNR are ignored, although no faults will be displayed if they are specified.
S1	Cutter radius compensation - left. A fault is flagged if the return angle (R) specified is negative.
S2	Cutter radius compensation - right. A fault is flagged if the return angle (R) specified is negative.

NOTE:



S0 is only permitted with L0. An error will be displayed if specified with either L1 or L2.

Lathe Rough Macro

The following changes have been made to the lathe rough macro:

- Corrected line-to-line to include check for line 2 reversal after extending line 1 to correct for line 1 reversal at the start of the part program.
- Restricted the start position of a "move at trail angle into the next rough pass sector" to the first profile position plus cut clearance.
- Fixed (rough pass) problem where cutter cut through the bottom of an island when going from the last rough pass for recess #1 to recess #2. The given rough pass line was tangent to arcs at the bottom.
- Corrected calculations for returning to the next rough pass start point in a recess when the tool is defined so that it enters the part at the trail or lead angle after retracting along the part.
- Improved rough pass return moves when tool is defined with trailing angle. The two possible cases:

Case 1 Intersection of return path line with extension of next rough pass line at trail angle is beyond stock boundary by more than cut clearance: axes rapid back to cut clearance position and then feed into part (perpendicular to just finished rapid move) to intersection with trail angle line.

Case 2 Original intersection is not beyond cut clearance point: axes rapid to position short of this intersection (cut clearance distance) if not already within cut clearance distance of intersection point. The axes then feed to this intersection.

jsToolActiveNum[]

Added **toolDefOrient** in tool definition table. This value is stored in the table for a specified tool number by the tool definition macro (G305: Q, Orient) and is retrieved by roughing, finishing, or grooving macros when the new variable, [jsToolActiveNum\[\]](#) is set by the PLC to a value greater than 0.

If [jsToolActiveNum\[\]](#) is less than or equal to 0, the active tool number is not found in the tool definition table, or a valid tool orientation value (0-8) has not been pre-stored in the table for the active tool, the macros retrieve the tool orientation value using [jsToolRadOffNum\[\]](#) as an index into the OpenCNC table.

RTX 6.5

Upgraded the RTX real-time software from 5.5 to 6.5. Version 6.5 allows for more Windows versions of HAL (i.e. ACPI Multiprocessor PC and ACPI Uniprocessor PC) and many more hardware options.

RTX 6.5 Pack Number

5222-PAC-32106-23288-55866

With the new version of RTX, TCP/IP is no longer a choice during installation. It is automatically installed.

winDataTool

The following changes have been made to [winDataTool](#):

- If more than 65535 points are collected, a prompt to save only the first 65535 points to the .csv file (which is the Microsoft Excel limit) is displayed. The user may also save all points.
- [winDataTool](#) is now allowed to trigger on byte (char) variables, and to use them as a frequency variable. ISaGRAF booleans are byte variables.
- Fixed problem where it crashed at startup if a previously selected variable no longer exists.

Job Loader/Job Creator

The following changes have been made to [Job Loader/Job Creator](#):

- These programs now start without a warning when OpenCNC is not running and the **-noFixture** argument is not used, displaying the Fixture Offset line to allow a fixture file to be added to a job. Don't prompt for a Fixture Offset file after entering a Tool Offset file in the Job Creator when **-noFixture** is used. If **-noFixture** is used, an error is displayed if job loaded is previewed with a fixture file, ignore the line with blank [defFixFileName](#) (for backward compatibility), do not clear fixture table when loaded, and do not save fixture line to job file.

- The load button on [Job Loader](#) is disabled when the program is loaded but OpenCNC is not running.
- The **Load**, **Preview**, and **Add** buttons has been disabled until a file is selected.
- The **Cancel** button has been renamed to **Close** in [Job Creator](#).
- [Job Creator](#) now prompts the user to save unsaved changes if the **Close** button is selected.

Rate of Compensation Limiting

Added a new feature which limits the amount of compensation added to the servo loop. Three new variables have been added to control compensation limits: [axExtCompRate\[\]](#) (external compensation - [axExtComp](#)), [axLSCompRate\[\]](#) (lead screw compensation), and [axBackLashRate\[\]](#) (backlash - [axBackLash\[\]](#)). If the variable values are set to their default value of 0, the system applies the whole compensation in one interrupt. This maximum rate is specified in distance units (du). For example, if the total backlash to be applied is 100 DU and [axBackLashRate\[\]](#) is set to 30 DU, then the backlash applied over the next 4 interrupts will be 30, 60, 90, 100.

Other Improvements

The following improvements have been made to OpenCNC 6.6.

- Calibration mode for [winOff](#) display as diameter now takes into account that the machine coordinates are returned as radius values and part coordinates are returned as diameter values.
- Added check to see if [defFixFilePath](#) was specified in [Job Creator](#). If not, it is now treated as [-noFixture](#) and displays an error message.
- Added return code to allow error message to remain visible on Load Job.
- Added [defMCut\(\)](#) which allows M Code execution without breaking cut-to-cut motion.
- Added [defMArg\(\)](#) which adds an additional user defined argument to M codes, which is passed to the PLC. When an M code is sent to the PLC for execution, the new optional argument, if present, is added to the end of the string of passed data. The user can call [cmdGetLo\(\)](#) to attempt to get that optional argument. Calling [cmdError\(\)](#) afterwards will tell him if the optional argument was missing or present. The optional argument is added in the GM program with a ":value", (i.e. M19:45).
- Corrected [blockReverse](#) so that two cycle starts are no longer required when retracing through blocks with [G09](#) code.


- Added a check for the direction of the main thread motion being reversed for the threading cut axis between the programmed direction (Z-K for turn thread, X-I for face thread) and final depth pass. If there is a direction reversal, a fault is flagged and the [threading macro](#) is aborted.
- Corrected checks of the second arc/arc intersection point for the [lathe macros](#).
- Added the **-b** argument to [winDataTool](#) allowing you to set the buffer size, which defaults to 1000000 bytes.
- Fixed a problems with very small TNR values when a linear block is needed to move the groove tool from one side of the tool to the other.
- Fixed problem with the [finish macros](#) to skip parsing of arc blocks if arc start = arc end position when rounded off to active dimensional units.
- Fixed a problem where the tool rapid through the part on the way to the programming start point.
- Fixed problem with the [grooving macro](#) when the current move is an arc and the next move is a line perpendicular to the cut angle.
- Fixed problem with the [grooving macro](#) during a face groove, the macro used to generate two cuts instead of just one when the groove width and tool width were the same.
- Corrected problems with arc-arc, line-arc and arc-line CRC. Some combinations were not detected correctly as inside or outside corners and therefore the wrong motions were generated.
- Changed code which defines the effective tool angle to be dependent on which axis and direction (+X, -X, +Z, -Z) is dominant in the cut angle and on the active tool orientation.
- Changed IPR velocity calculations to remove the lag caused by axis acceleration at the start and during thread cutting.
- Removed the ability to run multiple copies of [winAxTune](#) simultaneously.
- Removed Indramat SoftSERCANS 2V16 from the CDs. If 2V is required, please contact your MDSI sales representative.
- Corrected check that determines whether more than one rough groove pass is needed.
- Removed the ability to run multiple copies of [winAxTune](#) simultaneously.
- Fixed [winFeShow](#) to allow the indexes to go to the axis maximum (16) instead of job stream maximum (8).

- Fixed a problem where when using machine coordinates, the wrong offsets were displayed when the display mode was set to Diameter.
- Fixed [winFeShow](#) to allow the indexes to go to the axis maximum (16) instead of job stream maximum (8).
- Fixed a problem during successive [G33](#) blocks where OpenCNC was preventing any wait before the first of the successive [G33](#) blocks.
- Fixed a problem where if a block only contained a CSS on or off ([G96](#) / [G97](#)), OpenCNC was not waiting for the correct RPM before starting the cutting or threading sequence.
- The start of threading motion now makes the programmed starting angles be measured from the pseudo-encoder index mark in the direction of spindle rotation. Previously this was only working with evenly spaced multistart threads which had all passes defined. i.e. three starts at 120 degrees apart. In some cases, it would measure forward or backward from the pseudo-index.
- Corrected retract along part return point when multiple stacks are active during a rough macro rough pass (Lathe macros).
- Corrected alteration of part profile for UVW (finish pass) or TNR offsets in line-arc-line sequences when it has been determined that the arc must be thrown away (due to the application of the offsets) in the lathe macros.
- Added more testing for end of retract position when retracting along a part within a given profile table arc or line move.
- Added a call to [Groove macro](#) to return to its start point for options L10-L14.
- IPR velocity calculations have been modified to remove the lag caused by axis acceleration at the start and during thread cutting.
- Added **Edit** menu with sub-menus **Sort**, **Remove**, **Select All**, and **Unselect All** to [winDataTool](#) to handle the variable list.
- Changed **intDemo** (Demo Driver) to force the variables [defFakeMachine](#), [defFakeInterrupt](#), and [defDisablePcCard](#) to 1 after the tune file is read, so that it will run real tune files as a fake machine. This is done during the .tun file read so units will be correct, such as for [axVelMax\[\]](#).
- Corrected problems with arc-arc, line-arc, and arc-line CRC. Some combinations were not detected correctly as inside or outside corners and therefore the wrong motions were generated.
- Corrected line-line code to include check for line 2 reversal after extending line 1 to correct for line 1 reversal at the start of the part program. Restricted the start position

of a "move at trail angle into the next rough pass sector" to the first profile position plus cut clearance.

- A command-line option, **-noIntTech80**, has been added to the [Tech80 Carrier Card](#) driver **ioISATech805641.rtss** that will allow it to work correctly if **intTech80.rtss** is not being used. In the start (.ost) file:

```
# using a PCI card instead of the Tech80 5641
#p "C:\Program Files\MDSI\x.x\Programs\intTech80" -io_total 144
p "C:\Program Files\MDSI\x.x\Programs\intPci.rtss" -io_total 24
p "C:\Program Files\MDSI\x.x\Programs\ioISAOpto22" -noIntTech80
-io_start 0 -io_end 143
```

NOTE:  **-noIntTech80** does not have any parameters. If **intTech80.rtss** is used, then **-noIntTech80** should not be included with **ioISATech805641.rtss**.

- Added the new Xycom Elo touch driver for model 5015T/R2 to the installation CD.
- Added feature to the [grooving macro](#) permitting the user to program unidirectional groove finish pass without retracting and switching sides in the middle of the finish pass. Extended legitimate L-values. L10 and 12-14, the new values, operate the same as L0 and L2-4, respectively, except that finish pass moves are unidirectional. L11 is invalid.
- Fixed problems where [startSoftSercans1V](#), [startSoftSercans2V](#), [intSercos1V](#), and [intSercos2V](#) were attempting to load the wrong language .msg files, causing errors in the diagnostic window.
- Added check for direction of main thread motion being reversed for the thread cut axis between the programmed direction (Z-K for turn thread; X-I for face thread) and the final depth pass. If direction reversal, a fault is flagged and the macro is aborted.
- Altered the [Finishing Macro](#) to flag a fault if total depth is equal to number of finish passes (P) * cut depth per finish pass (L).
- Corrected blockReverse so that two cycle starts are no longer required when retracing through blocks with [G09](#) code.
- Modified the start of threading motion to make the programmed starting angles be measured from the pseudo-encoder index mark after the mark in the direction of spindle rotation. Previously, it would correctly do only evenly spaced multistart threads which had all passes defined. i.e. three starts at 120 degrees apart. In some cases, it would measure forward or backward from the pseudo-index.

- It is now possible to set a string variable to an empty string in [winVar](#).
- The table probe cycles ([G37](#): S1, S3) which set the tool length were miscalculating the expected contact point when [G92](#), fixture, or tool offsets were nonzero for the probe axis (Z). As a result, the user was usually getting erroneous faults that indicated that the machine was starting too close to the expected contact point or that no contact was found within the expected limits.
- Fixed [winPlot](#) to handle a negative radius in the offset table, which is used to switch between cutter left and cutter right.
- Fixed [winPos](#) for rotary wrapped axes to display positions only within the limits [axLocAbsMin\[\]](#) and [axLocAbsMax\[\]](#).
- Fixed problem in [winMenu](#) where programs in the StartUp group were not restarted if at a higher authority level than what is displayed, such as sevCalc. Programs in the StartUp group are always initially started, no matter what the authority level. Restarting the StartUp group by picking it will now restart all programs, even those not displayed due to authority level. Also fixed a problem where a check mark was displayed in the menu indicating a program was running when it was not.
- Fixed a problem in winVar where loading variable lists would omit some two dimensional array elements. The maximum index values were applied incorrectly for rows and columns, which avoided loading some elements such as [axLSCompDirPos\[16\]\[0\]](#) and higher.
- Fixed **blkCalc** so it will terminate a macro if it generates a block with a scan error.
- Fixed a problem where subroutines may work incorrectly if run simultaneously in two parsers and when using nested subroutines.
- Made a change with [MC_TAPEPTR](#). If [MC_TAPEPTR](#) is sent from a macro, only update the displayed program line, do not stop the sequencer. This eliminates the erroneous second cycle start required at an [M01](#) and preserves the highlighted line in the Subroutine/Macro window.
- Improved the plot FIFO code to wrap the values for wrapped axes.
- Added a [living tooling](#) macro for milling machines.
- Made a change that causes the drill cycle final depth and R-plane values to be treated as diameter inputs when the drill axis is a diameter/radius axis and diameter mode is active.