Okuma America Corporation

THINC-API
Release Notes for Lathe

Document No.: S5015-008-28

Revision History

Date	Version	Description	Author
5/21/2007	S5015-008-00	Public release for Lathe THINC-API version 1.0.0.0	LHuynh
6/04/2007	S5015-008-01	Public release for Lathe THINC-API version 1.1.0.0	Lhuynh
08/15/2007	S5015-008-02	Public release for Lathe THINC-API version 1.2.0.0	Lhuynh
2/22/2008	S5015-008-03	Public Release 1.6.0.0 for Lathe THINC-API	Lhuynh
4/11/2008	S5015-008-04	Public Release 1.6.0.0 for Lathe THINC-API	Lhuynh
06/27/2008	S5015-008-05	Public Release 1.6.3.0 for Lathe THINC-API	Lhuynh
07/25/2008	S5015-008-06	Public Release 1.6.4.0 for Lathe THINC-API	Lhuynh
10/27/2008	S5015-008-07	Public Release 1.7.0.0 for Lathe THINC-API	Lhuynh
12/18/2008	S5015-008-08	Public Release 1.8.0.0 for Lathe THINC-API	Lhuynh
04/30/2009	S5015-008-09	Public Release 1.9.1.0 for Lathe THINC-API	Lhuynh
10/12/2009	S5015-008-10	Public Release 1.10.0.0 for Lathe THINC-API	Lhuynh
04/28/2010	S5015-008-11	Public Release 1.11.0.0 for Lathe THINC-API	Lhuynh
09/14/2010	S5015-008-12	Public Release 1.11.1.0 for Lathe THINC-API	Lhuynh
01/04/2011	S5015-008-13	Public Release 1.12.0.0 for Lathe THINC-API	Lhuynh
02/02/2011	S5015-008-14	Public Release 1.12.1.0 for Lathe THINC-API	Lhuynh
11/14/2011	S5015-008-15	Beta Release 1.14.0.0 for Lathe THINC-API	Lhuynh
01/15/2012	S5015-008-16	Beta Release 1.14.1.0 for Lathe THINC-API	Lhuynh
09/21/2012	S5015-008-17	Beta Release 1.14.2.0 for Lathe THINC-API	Lhuynh
12/04/2012	S5015-008-18	Public Release 1.15.0.0 for Lathe THINC-API	Lhuynh

03/06/2013	S5015-008-19	Beta Release 1.15.1.0 for Lathe THINC-API	Lhuynh
03/18/2013	S5015-008-20	Beta Release 1.15.2.0 for Lathe THINC-API	Lhuynh
09/25/2013	S5015-008-21	Public Release 1.16.0.0 for Lathe THINC-API	Lhuynh
01/15/2014	S5015-008-22	Public Release 1.17.0.0 for Lathe THINC-API	Lhuynh
04/01/2014	S5015-008-23	Public Release 1.17.1.0 for Lathe THINC-API	Lhuynh
10/08/2014	S5015-008-24	Public Release 1.17.2.0 for Lathe THINC-API	Lhuynh
10/18/2015	S5015-008-25	Public Release 1.18.0.0 for Lathe THINC-API	Lhuynh
10/18/2016	S5015-008-26	Public Release 1.19.0.0 for Lathe THINC-API	Lhuynh
12/07/2017	S5015-008-27	Public Release 1.20.0.0 for Lathe THINC-API	Lhuynh
06/14/2018	S5015-008-28	Public Release 1.21.1.0 for Lathe THINC-API	Lhuynh

Table of Contents

Release Notes for Lathe

1. Introduction

1.1 Disclaimer of Warranty

Okuma America Corporation makes no representations or warranties, either expressed or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

1.2 Purpose

The purpose of the ***Release Notes*** document is to communicate major new features and changes in this release of the THINC-API for Lathe libraries. It also documents known problems and workarounds.

1.3 Scope

This document describes Public Release 1.21.1.0 of THINC-API for Lathe.

1.4 Definitions, Acronyms, and Abbreviations

GAC – Global Assembly Cache Windows folder located in 'C:\WINDOWS\assembly'

1.5 References

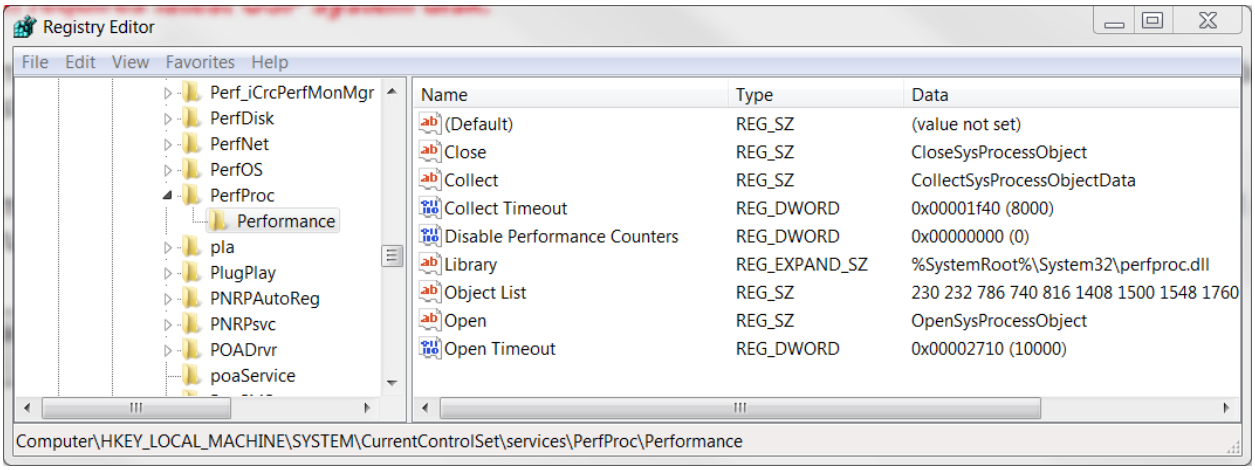
None.

2. About This Release

Public Release of the THINC-API library for Lathe supports the following:

From this release and forward, THINC-API libraries will check dependency libraries during installation. THINC-API will fail to install if version of dependency OCJ libraries cannot support current version of THINC-API.

From this release and forward, API Notifier will delay the checking of API for an approximately of 1 minutes or so after NC is running. API Notifier service does use Windows Performance Counters service. As a result, API Notifier will not run correctly if Windows Performance Counters service is not enable. During the installation of THINC-API, Windows Performance Counters will be reset to 0 in the Windows system registry as shown below:



All applications compiled with Beta Release from version 1.15.X.X must be compiled with Public Release version 1.17.0.0 or higher when it is available.

In current version of THINC-API, some of the existing functions related to ATC, Tool, TailStock, and Chuck Data from DATA-API or Command API might not function correctly on OSP-P300S (SLP) and OSP-P300L control.

Please refer to the help file for detail usage and compatibility information of each function.
This version requires latest OSP system disk.

Libraries included in this release for Lathe are compiled with .NET Framework 4.0:

Version of Okuma.CLDATAPI.dll in this release is 3.1.0.0

Version of Okuma.CLCMDAPI.dll in this release is 3.1.0.0

Version of APINotifierService.exe in this release is 1.21.1.0

Version of APINotifierStatus.exe in this release is 1.3.0.0

Version of Okuma.Flexnet.Net4.dll in this release is 1.0.0.0

Version of Okuma.Apilog2.dll in this release is 1.2.0.0

This release requires OCJ custom API version 003U on target machine. THINC-API will verify the existing of OCJ custom API version before performing the installation.

The PLC system package listed in the table per control type is also required.

OSP	PLCS package
P100II/P200	From 110A to 110C
P200A Type1	From 120A to 130A
P200A Type2/P300	From 201B to 201G, or 300A and over

Important: All applications designed with THINC-API libraries must use THINC-API version 1.21.1.0 or higher to be able to run on Microsoft Windows 10.

The following functions of Okuma.CLDATAPI.dll library will not be available in OSP-P100II control machines:

Classes	Interfaces
CMachine	OnOffStateEnum GetNCStatus(NCStatusEnum enNCStatus)
CMachine	CCurrentAlarm* GetCurrentAlarm()
CIO	OnOffStateEnum GetUserTaskIOVariable(IOTypeEnum enIO, Int32 intIndex) ;
CIO	void SetUserTaskOutputVariable(Int32 intIndex, OnOffStateEnum enValue) ;
CIO	OnOffStateEnum GetProtectedUserTaskOutputVariable(Int32 intIndex) ;
CIO	void SetProtectedUserTaskOutputVariable(Int32 intIndex, OnOffStateEnum enValue) ;

The following functions of Okuma.CLCMDAPI.dll library will not be available in OSP-P100II control machines:

Classes	Interfaces
CATC	Sub RegisterToolPot(ByVal intPotNo As Integer, ByVal intToolNo As Integer, ByVal enSettingToolKind As SettingToolKindEnum, ByVal enSettingToolSize As SettingToolSizeEnum, ByVal enReturnMagazine As ReturnMagazineEnum)
CATC	Sub SetNextTool(ByVal intToolNo As Integer, ByVal enSettingToolKind As SettingToolKindEnum, ByVal enSettingToolSize As SettingToolSizeEnum, ByVal enReturnMagazine As ReturnMagazineEnum)
CATC	Sub SetToolInStation(ByVal intToolNo As Integer, ByVal enSettingToolKind As SettingToolKindEnum, ByVal enSettingToolSize As SettingToolSizeEnum, ByVal enReturnMagazine As ReturnMagazineEnum, ByVal enTurretStation As TurretStationEnum)
CATC	Sub UnRegisterToolPot(ByVal intPotNo As Integer)
CProgram	CancelMainProgram()
CProgram	SelectMainProgramRSide(ByVal strMainProgramFileName As String, Optional ByVal strSubProgramFileName As String = "", Optional ByVal strSystemSubtituteProgramFileName As String = "", Optional ByVal strProgramName As String = "")
CProgram	SelectMainProgramLSide(ByVal strMainProgramFileName As String, Optional ByVal strSubProgramFileName As String = "", Optional ByVal strSystemSubtituteProgramFileName As String = "", Optional ByVal strProgramName As String = "")
CTools	CalcualteToolOffset(ByVal intToolNo As Integer, ByVal enAxisIndex As OffsetAxisIndexEnum, ByVal enSubSystem As SubSystemEnum, ByVal dblValue As Double)
CTools	AddConstantToolOffset(ByVal intOffsetNo As Integer, ByVal enAxisIndex As OffsetAxisIndexEnum, ByVal enSubSystem As SubSystemEnum, ByVal enCuttingPosition As CuttingPositionEnum)
CTools	AddConstantNoseRadiusCompensation(ByVal intOffsetNo As Integer, ByVal enAxisIndex As OffsetAxisIndex2Enum, ByVal enSubSystem As SubSystemEnum, ByVal enCuttingPosition As CuttingPositionEnum)
CTools	AddConstantToolWear(ByVal intOffsetNo As Integer, ByVal enAxisIndex As OffsetAxisIndex2Enum, ByVal enSubSystem As SubSystemEnum, ByVal enCuttingPosition As CuttingPositionEnum)
CTools	SubtractConstantToolOffset(ByVal intOffsetNo As Integer, ByVal enAxisIndex As OffsetAxisIndexEnum, ByVal enSubSystem As SubSystemEnum, ByVal enCuttingPosition As CuttingPositionEnum)
CTools	SubtractConstantNoseRadiusCompensation(ByVal intOffsetNo As Integer, ByVal enAxisIndex As OffsetAxisIndex2Enum, ByVal enSubSystem As SubSystemEnum, ByVal enCuttingPosition As CuttingPositionEnum)

CTools	SubtractConstantToolWear(ByVal intOffsetNo As Integer, ByVal enAxisIndex As OffsetAxisIndex2Enum, ByVal enSubSystem As SubSystemEnum, ByVal enCuttingPosition As CuttingPositionEnum)
CProgram	SelectScheduleProgramLSide(ByVal strScheduleProgramFileName As String)
CProgram	SelectScheduleProgramRSide(ByVal strScheduleProgramFileName As String)
CMachine	Public Sub SetUserAlarm(ByVal enAlarm As UserAlarmEnum, Optional ByVal strAlarmMessage As String = "", Optional ByVal enUserAlarmSubSystem As UserAlarmSubSystemEnum = 0)
CMachine	Public Sub ClearUserAlarmD(ByVal enUserAlarmSubSystem As UserAlarmSubSystemEnum)

Note: User Alarm will require OKUMA.Lathe.UserAlarm license feature in order to function on OSP-P200 or newer controls if machine can support.

3. Features

Please refer to the help file for detail information of each function.

3.1 Command API

The following functions are affected by LTIM NC 2 spec code [27, 2] P300L or TMIM NC 2 spec code [3, 6] P300S machines:

States	Controls	Classes	Interfaces
CHANGE	P100II P200	CATC	Public Sub UnRegisterToolPot(ByVal intPotNo As Integer)
CHANGE	P100II P200	CATC	Public Sub RegisterToolPot(ByVal intPotNo As Integer, ByVal intToolNo As Integer, ByVal enSettingToolKind As SettingToolKindEnum, ByVal enSettingToolSize As SettingToolSizeEnum, ByVal enReturnMagazine As ReturnMagazineEnum)
CHANGE	P100II P200	CATC	Public Sub SetNextTool(ByVal intToolNo As Integer, ByVal enSettingToolKind As SettingToolKindEnum, ByVal enSettingToolSize As SettingToolSizeEnum, ByVal enReturnMagazine As ReturnMagazineEnum)
CHANGE	P100II P200	CATC	Public Sub SetToolInStation(ByVal intToolNo As Integer, ByVal enSettingToolKind As SettingToolKindEnum, ByVal enSettingToolSize As SettingToolSizeEnum, ByVal enReturnMagazine As ReturnMagazineEnum, ByVal enTurretStation As TurretStationEnum)
CHANGE	P300L P300S	CATC	Public Sub AttachTool(ByVal intToolNo As Integer, ByVal intTarget As Integer, ByVal enToolLocation As ToolLocationEnum)

CHANGE	P300L P300S	CATC	Public Sub DeleteTool(ByVal intToolNo As Integer)
CHANGE	P300S	CATC	Public Sub RegisterTool(ByVal intToolNo As Integer, ByVal intTargetNo As Integer, ByVal enEdgeNo As ToolEdgeEnum, ByVal enSettingToolKind As Enumerations.SettingToolKindEnum, ByVal enSettingToolSize As Enumerations.SettingToolSizeEnum, ByVal enSpindleAxis As Enumerations.BasicPositionSpindleEnum, ByVal enBaseCuttingPositionSetting As Enumerations.BaseCuttingPositionSettingEnum, ByVal dblAngle As Double, ByVal enToolLocation As ToolLocationEnum)
CHANGE	P300L	CATC	Public Sub RegisterTool(ByVal intToolNo As Integer, ByVal intTargetNo As Integer, ByVal enSettingToolKind As Enumerations.SettingToolKindEnum, ByVal enSpindleAxis As Enumerations.BasicPositionSpindleEnum, ByVal enToolLocation As ToolLocationEnum)
CHANGE	P300L P300S	CATC	Public Sub DetachTool(ByVal intTarget As Integer, ByVal enToolLocation As ToolLocationEnum)
CHANGE	P300L P300S	CATC	Public Sub RegisterToolEdge(ByVal intToolNo As Integer, ByVal enEdgeNo As ToolEdgeEnum, ByVal enSpindleAxis As Enumerations.BasicPositionSpindleEnum, ByVal enBaseCuttingPositionSetting As Enumerations.BaseCuttingPositionSettingEnum, ByVal dblAngle As Double)
CHANGE	P300L P300S	CATC	Public Sub RegisterToolCuttingPosition(ByVal intToolNo As Integer, ByVal enEdgeNo As ToolEdgeEnum, ByVal enToolCuttingPosition As Enumerations.ToolCuttingPositionEnum, ByVal enSpindleAxis As Enumerations.BasicPositionSpindleEnum, ByVal enBaseCuttingPositionSetting As Enumerations.BaseCuttingPositionSettingEnum, ByVal dblAngle As Double)
CHANGE	P300L P300S	CATC	Public Sub DeleteToolEdge(ByVal intToolNo As Integer, ByVal enToolEdge As ToolEdgeEnum)
CHANGE	P300S	CATC	Public Sub DeleteToolCuttingPosition(ByVal intToolNo As Integer, ByVal enToolEdge As ToolEdgeEnum, ByVal enToolCuttingPosition As Enumerations.ToolCuttingPositionEnum)
NEW	P100II/P200	CTools	OneTouchCalculateToolOffset(ByVal intToolNo As Integer, ByVal enAxisIndex As OffsetAxisIndex2Enum, ByVal enSubSystem As SubSystemEnum)

REMOVE	P100II/P200	CTools	AutoCalculateToolOffset(ByVal intToolNo As Integer, ByVal enAxisIndex As OffsetAxisIndex2Enum, ByVal enSubSystem As SubSystemEnum)

3.2 DATA-API

3.2.1 Multi-Tool Function on P300L

The functions listed below are changed to support Multi-Tool spec NC 2 [24, 7] on P300L.

States	Controls	Classes	Interfaces
CHANGE	ALL	CMSpindle	GetCurrentToolInH1Turret() As Integer
CHANGE	ALL	CTools	AddActualToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, dblValue As Double)
CHANGE	P300S_TD P300L	CTools	GetActualToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As Double
CHANGE	P300S P300L	CTools	GetGroupNo (intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As Integer
CHANGE	P300S_TD P300L	CTools	GetReferenceToolOffset3 (intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As Integer
CHANGE	P300S P300L	CTools	GetToolGroupEntry3 (intGroupNo As Integer) As String()
CHANGE	ALL	CTools	GetToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer) As Double
CHANGE	P100II P200 P300S_TL P300L	CTools	GetToolNumberInGroup (intGroupNo As Integer) As Integer

CHANGE	P300S P300L	CTools	SetGroupNo (intToolNo As Integer, enEdgeNo As ToolEdgeEnum, intGroupNo As Integer)
CHANGE	ALL	CTools	SetGroupNo (intToolNo As Integer, intGroupNo As Integer)
CHANGE	P300S_TD P300L	CTools	SetReferenceToolOffset2 (intToolNo As Integer, enEdgeNo As ToolEdgeEnum, intOffsetNo As Integer)
CHANGE	P100II P200 P300S_TL P300L	CTools	SetReferenceToolOffset2(intToolNo As Integer, intOffsetNo As Integer)
CHANGE	P300S_TD P300L	CTools	SetReferenceToolOffset3 (intToolNo As Integer, enEdgeNo As ToolEdgeEnum, intOffsetNo As Integer)
CHANGE	P100II P200 P300S_TL P300L	CTools	SetReferenceToolOffset3(intToolNo As Integer, intOffsetNo As Integer)
CHANGE	P300S_TD P300L	CTools	AddActualToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, enEdgeNo As ToolEdgeEnum, dblValue As Double)
CHANGE	ALL	CTools	AddToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, dblValue As Double)
CHANGE	P300S_TD P300L	CTools	AddToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, enEdgeNo As ToolEdgeEnum, dblValue As Double)
CHANGE	ALL	CTools	GetActualToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer) As Double

CHANGE	P100II P200 P300S_TL P300L	CTools	GetCurrentEdge (intGroupNo As Integer) As Integer
NEW	P300L	CTools	GetCurrentToolEdge As ToolEdgeData
CHANGE	ALL	CTools	GetGaugeStatus (intToolNo As Integer) As ToolGaugeStatusEnum
CHANGE	P300S_TD P300L	CTools	GetGaugeStatus (intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As ToolGaugeStatusEnum
CHANGE	ALL	CTools	GetGroupNo (intToolNo As Integer) As Integer
CHANGE	P100II P200 P300S P300L	CTools	GetGroupToolLifeStatus (intGroupNo As Integer) As GroupToolLifeStatusEnum
CHANGE	P100II P200 P300S P300L	CTools	GetGroupToolLifeStatus (intGroupNo As Integer) As GroupToolLifeStatusEnum
CHANGE	ALL	CTools	GetLifeStatus (intToolNo As Integer) As ToolLifeStatusEnum
CHANGE	P300S_TD P300L	CTools	GetLifeStatus (intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As ToolLifeStatusEnum
CHANGE	P100II P200 P300S_TL P300L	CTools	GetReferenceToolOffset1 (intToolNo As Integer) As Integer
CHANGE	P300S_TD P300L	CTools	GetReferenceToolOffset1 (intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As Integer

CHANGE	P100II P200 P300S_TL P300L	CTools	GetReferenceToolOffset2 (intToolNo As Integer) As Integer
CHANGE	P300S_TD P300L	CTools	GetReferenceToolOffset2 (intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As Integer
CHANGE	P100II P200 P300S_TL P300L	CTools	GetReferenceToolOffset3 (intToolNo As Integer) As Integer
CHANGE	P300S_TD P300L	CTools	GetToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, enEdgeNo As ToolEdgeEnum) As Double
CHANGE	ALL	CTools	SetActualToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, dblValue As Double)
CHANGE	P300S_TD P300L	CTools	SetActualToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, enEdgeNo As ToolEdgeEnum, dblValue As Double)
CHANGE	P100II P200 P300S_TL P300L	CTools	SetCurrentEdge (intGroupNo As Integer, intEdgeNo As Integer)
CHANGE	ALL	CTools	SetGaugeStatus (intToolNo As Integer, enValue As ToolGaugeStatusEnum)
CHANGE	P300S_TD P300L	CTools	SetGaugeStatus (intToolNo As Integer, enEdgeNo As ToolEdgeEnum, enValue As ToolGaugeStatusEnum)
CHANGE	P100II P200 P300S P300L	CTools	SetGroupToolLifeStatus (intGroupNo As Integer, enValue As GroupToolLifeStatusEnum)

CHANGE	ALL	CTools	SetLifeStatus (intToolNo As Integer, enValue As ToolLifeStatusEnum)
CHANGE	P300S_TD P300L	CTools	SetLifeStatus (intToolNo As Integer, enEdgeNo As ToolEdgeEnum, enValue As ToolLifeStatusEnum)
CHANGE	P300S_TD P300L	CTools	SetReferenceToolOffset1 (intToolNo As Integer, enEdgeNo As ToolEdgeEnum, intOffsetNo As Integer)
CHANGE	P100II P200 P300S_TL P300L	CTools	SetReferenceToolOffset1 (intToolNo As Integer, intOffsetNo As Integer)
CHANGE	ALL	CTools	SetToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, dblValue As Double)
CHANGE	P300S_TD P300L	CTools	SetToolLife (enToolLifeCondition As ToolLifeConditionEnum, intToolNo As Integer, enEdgeNo As ToolEdgeEnum, dblValue As Double)

3.2.2 Misc.

The functions listed below are changed to support P300.

States	Controls	Classes	Interfaces
CHANGE	P100II P200 P300S	CATC	GetNextTool() As ToolProperty
CHANGE	P100II P200 P300S	CATC	GetToolInStation() As Integer

CHANGE	P100II P200 P300S	CATC	GetToolNo (intPotNo As Integer) As Integer
NEW	ALL	CBASE	Boolean IsLTMIMSpecOn()

When the function IsLTMIMSpecOn() returns FALSE, all tool data related functions designed specifically for P300L or P300S will not work. As a result, tool data related functions are designed for P100II/P200 machines must be used. For functions that are compatible with different control types, it will work and behave as it is designed for that particular control type.

4. Known Bugs and Limitations

This section identifies known problems in this release and describes any work-arounds.

4.1 Defect

4.1.1 General Defect

DATA-API library, Okuma.CLDATAPI.dll, cannot create directly under ASP.NET web application.

Solution/Work-arounds: Create and initialize DATA-API in a separate thread. All function calls must be called from objects created inside separated thread mentioned above.

DATA-API library can only support applications designed with single-threaded apartment of COM threading model. The underlying library, LDATAPI.dll, cannot be loaded during call to CMachine.Init function when an MTAThread attribute is applied to the application.

Solution/Work-arounds: None

4.1.2 Firebird Database Log File

THINC-API libraries compiled with .NET Framework 1.1 use Firebird Database engine. The log file created by Firebird database engine can be corrupted, unexpectedly.

Solution/Work-arounds: A bat file is created during the installation of THINC-API and will be executed every time Windows is first started to replace the existing log.fdb located in “D:\Program files\Okuma\LoggingService” to prevent file corruption.

New THINC-API libraries that compiled with .NET Framework 4.0 will use SQLite database for its logging service. All applications should use new version THINC-API if possible to prevent using Firebird database and to be compatible with Windows 10.

4.1.3 Data-API

4.1.3.1 MacMan.COperationHistory class

Function:

Int32 COperation* GetOperationHistory(Int32 intIndex) ;

ArrayList* GetOperationHistory(Int32 intFromIndex, Int32 intToIndex);
Int32 GetMaxCount() ;
Int32 GetCount() ;

Symptom: Failed to get correct data for Subsystem L and R side if MacMan screen is different than current setting of subsystem. It always gets the data from current MacMan screen.

Solution/Work around: None

4.1.3.2 CAxis class

Function:

Double GetActualPositionProgramCoord(AxisIndex1Enum enAxisIndex);
Double GetTargetPosition (AxisIndex1Enum enAxisIndex);

Symptom: The Z-axis position for sub system NC-AL and NC-AR does not return data correctly. It is based on the current selection of spindle in NC panel or command program G140/G141.

Solution/Work around: None

4.2 Defects Fixed in this Release

4.2.1 API Notifier Status/Service

Symptom: API Notifier Status cannot update NC running status correctly on some Windows.
If THINC-API is installed on Windows x64, API Notifier service cannot detect NC process.
If THINC-API is installed on Windows XP and performance counter is not enabling, API Notifier service cannot detect NC process.

Solution/Work around: New version of ApiNotifier service compiled with .NET Framework 4.0 corrects all of these issues on both Windows 7/XP x86 or x64.