

---

# **Okuma America Corporation**

---

## **THINC-API Release Notes for Lathe**

**Document No.: S5015-008-09**

THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

## 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

THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

## Table of Contents

1.	Introduction	4
1.1	Disclaimer of Warranty	4
1.2	Purpose	4
1.3	Scope	4
1.4	Definitions, Acronyms, and Abbreviations	4
1.5	References	4
2.	About This Release	4
3.	New Features	6
3.1	Data API	6
3.2	Command API	7
4.	Known Bugs and Limitations	7
4.1	Defect	7
4.2	General Defect	7
4.2.1	Data-API	8
4.3	Defects Fixed in this Release	9
4.3.1	Data API	9

THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

# 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.

**Copyright © 2009, Okuma America Corporation.**  
**All rights reserved.**

GOVERNMENT RIGHTS LEGEND: Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Okuma America Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

"Okuma America Corporation" and Okuma America Corporation's products are trademarks of Okuma America Corporation. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

### 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 1.9.1.0 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 1.9.1.0 of the THINC-API library for Lathe supports the following:

From this release, THINC API libraries will check dependency libraries during installation and at run-time. THINC API will fail to install or load if version of dependency libraries cannot support current version of THINC API.

Version of Okuma.CLDATAPI.dll in Public Release 1.9.1.0 is 1.9.0.0

Version of Okuma.CLCMDAPI.dll in Public Release 1.9.1.0 is 1.5.0.0

Version of APINotifierService.exe in Public Release 1.9.1.0 is 1.9.0.0

Version of Okuma.Flexnet.dll in Public Release 1.9.1.0 is 1.2.0.0

THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

The following functions of Okuma.CLDATAPI.dll library will be only available in OSP-P200 control machines:

Classes	Interfaces
CMachine	OnOffStateEnum GetNCStatus(NCStatusEnum enNCStatus)
CMachine	CCurrentAlarm* GetCurrentAlarm()
CSpec	String* GetMachineSerialNumber()

The following functions of Okuma.CLCMDAPI.dll library will be only available in OSP-P200 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 strSystemSubstituteProgramFileName As String = "", Optional ByVal strProgramName As String = "")
CProgram	SelectMainProgramLSide(ByVal strMainProgramFileName As String, Optional ByVal strSubProgramFileName As String = "", Optional ByVal strSystemSubstituteProgramFileName 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)

### 3. New Features

The new classes/functions available in Public Release 1.9.1.0 are:

#### 3.1 Data API

Classes	Interfaces
CATC	struct ToolProperty GetNextTool()
CATC	Int32 GetToolInReadyStation()
CATC	SetToolInReadyStation(Int32 intToolNo)
CATC	struct ToolProperty GetToolInStation(TurretStationEnum enTurretStation)
CATC	MagazineEnum GetMagazineProperty()
CATC	Int32 GetPotNo(Int32 intToolNo)
CATC	ATCTypeEnum GetATCType()
CATC	Int32 GetMagazinePosition(Int32 intMagazinePosition)
CATC	Boolean IsReservedPot(Int32 intPotlNo)
CMachine	OnOffStateEnum GetNCStatus(NCStatusEnum enNCStatus)
CTools	ToolGaugeStatusEnum GetGaugeStatus(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo)
CTools	void SetGaugeStatus(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo, ToolGaugeStatusEnum enValue)
CTools	ToolLifeStatusEnum GetLifeStatus(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo)
CTools	void SetLifeStatus(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo, ToolLifeStatusEnum enValue)
CTools	GroupToolLifeStatusEnum GetGroupToolLifeStatus2(Int32 intGroupNo)
CTools	void SetGroupToolLifeStatus2(Int32 intGroupNo, GroupToolLifeStatusEnum enValue)
CTools	Int32 GetGroupNo(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo)
CTools	void SetGroupNo(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo, Int32 intGroupNo)
CTools	Int32 GetReferenceToolOffset1(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo)
CTools	void SetReferenceToolOffset1(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo, Int32 intValue)
CTools	Int32 GetReferenceToolOffset2(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo)
CTools	void SetReferenceToolOffset2(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo, Int32 intValue)

CTools	Int32 GetReferenceToolOffset3(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo)
CTools	void SetReferenceToolOffset3(MultiEdgesAngleToolHolderEnum enToolHolder, MultiEdgesAngleToolEnum enMultiEdgesTool, Int32 intEdgeNo, Int32 intValue)
CTools	String* GetToolGroupEntry2(Int32 intGroupNo, Int32 intEntryToolNo)
CTools	Int32 GetToolNumberInGroup2(Int32 intGroupNo)
CTools	void SetToolNumberInGroup2(Int32 intGroupNo, Int32 intToolNo)
CTools	Int32 GetCurrentEdge(Int32 intGroupNo)
CTools	void SetCurrentEdge(Int32 intGroupNo, Int32 intEdgeNo)
CTools	GroupToolLifeStatusEnum GetGroupToolLifeStatus(Int32 intGroupNo) void SetGroupToolLifeStatus(Int32 intGroupNo, GroupToolLifeStatusEnum enValue)
CTools	Int32 GetToolGroupEntry(Int32 intGroupNo, Int32 intEntryToolNo)

## 3.2 Command API

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)

## 4. Known Bugs and Limitations

This section identifies known problems with Public Release 1.9.1.0 and describes any work-arounds.

### 4.1 Defect

### 4.2 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

THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

## 4.2.1 Data-API

### 4.2.1.1 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

### 4.2.1.2 CAxis class

Function:

```
Double GetActualPositionProgramCoord( AxisIndex1Enum enAxisIndex);
Double GetActualPositionMachineCoord( AxisIndex1Enum enAxisIndex);
Double GetDistanceToTargetPosition( AxisIndex1Enum enAxisIndex);
Double GetTargetPosition( AxisIndex1Enum enAxisIndex);
```

Symptom: Axis data for XI, ZI return -1 meanwhile NC displays \*,\*\*\*

Solution/Work around: None

```
enum AxisTypeEnum GetAxisType(enum AxisIndexEnum enValue)
```

Symptom: Return axis type for C axis as rotary axis on some machine with old version of NC Control software.

Solution/Work around: C is always a rotary axis. It could be hard code. It is best to upgrade NC Control software to support current THINC API version.

### 4.2.1.3 CBallScrew class

Function:

```
Public Function GetInterval(ByVal enAxisIndex As AxisIndex3Enum) As Double
```

Symptom: The function return data in metric unit even it has been set Inch unit.

Solution/Work around: Perform a unit conversion from metric to inch unit if needed after performing the operation

Function:

```
Public Sub SetInterval(ByVal enAxisIndex As AxisIndex3Enum)
```

Symptom: The function set data in metric unit even it has been set Inch unit.

Solution/Work around: Perform a unit conversion from metric to inch unit if needed before performing the operation

### 4.2.1.4 CChuck class

Function:

```
Public Sub CalGrippingFaceToProgramZeroDistance(ByVal enValue As ChuckIndexEnum, ByVal dblValue As Double)
```



THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

Symptom: The function allows performing a calculation over max setting limit.  
Valid setting range is -393.70078 to +393.70078 for inch unit.

Solution/Work around: Perform validation of input parameters before calling the function

#### 4.2.1.5 COptionalParameter class

Public Sub SetNCOptionalParameterLongWord(ByVal *intLongWordIndex* As Integer, ByVal *intValue* As Integer)

Symptom: This function fails to set value that is greater than 32767.

Solution/Work around: Perform multiple settings with a value less than 32768 until achieving the final result.

#### 4.2.1.6 CWorkpiece class

Public Sub CalZeroOffset(ByVal *enAxisIndex* As AxisIndex4Enum, ByVal *dblValue* As Double)

Symptom: This function allows performing over the max setting limit. The allowable setting limit is:

	Metric	Inch
0.1 micron option	-9999.9999 To 9999.9999	-393.70078 To 393.70078
Without 0.1 micron option	-99999.999 To 99999.999	-3937.0078 To 3937.0078

Solution/Work around: Perform validation of input parameters before calling the function

Public Sub CalZeroShift (ByVal *enAxisIndex* As AxisIndex4Enum, ByVal *dblValue* As Double)

Symptom: This function allows performing over the max setting limit. The allowable setting limit is:

	Metric	Inch
0.1 micron option	-9999.9999 To 9999.9999	-393.70078 To 393.70078
Without 0.1 micron option	-99999.999 To 99999.999	-3937.0078 To 3937.0078

Solution/Work around: Perform validation of input parameters before calling the function

### 4.3 Defects Fixed in this Release

#### 4.3.1 Data API

##### 4.3.1.1 CATC Class

Function:

Int32 GetToolNo(Int32 intPotNo)

Symptom: It returns tool number on pot having DUMMY POT attributte

Function:

Int32 GetToolKind(Int32 intPotNo)

Symptom: It returns incorrect tool kind

THINC-API	Version: S5015-008-09
Release Notes For Lathe	Date: 04/30/2009

Function:

Int32 GetToolSize(Int32 intPotNo)

Symptom: It returns incorrect tool size