SOFTWARE

OSP WIDGET MANAGER

USERS MANUAL

(1st Edition)
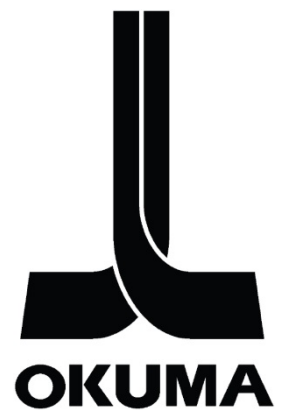
OKUMA

SAFETY PRECAUTIONS

Okuma machines are equipped with safety equipment to protect persons and machinery from any danger that might happen unexpectedly. However operators are requested to operate the equipment in such a way that no person will use the equipment in the following unsafe manner and cause an accident without knowing the following safety precautions.
This instruction manual and the warning sings attached to the machine cover only those hazards which Okuma can predict.
Be aware that they do not cover all possible hazards.

1. Precautions Relating to Installation

(1) Please note the following precautions regarding the primary power supply.

- Do not draw the primary power supply from a distribution panel that also supplies a major noise source (for example, an electric welder or electric discharge machine) since this could cause malfunction of the CNC unit.

- If possible, connect the control enclosure to a ground independently. If there is no choice but to use a common ground, the other equipment must not generate a large amount of noise (such as an electric welder or electric discharge machine).

(2) Installation Environment
Observe the following points when installing the CNC unit and the control enclosure.

- Make sure that the CNC unit will not be subject to direct sunlight.

- Make sure that the control enclosure will not be splashed with chips, water, or oil.

- Make sure that the control enclosure and the operation panel will not be subject to excessive vibration or shock.

- Permissible ambient temperature range of the control enclosure is 5 to 40°C (41 to 104°F).

- The permissible ambient humidity range for the control enclosure is relative humidity 50% or less at 40°C (104°F) (no condensation).

- The maximum altitude at which the control enclosure can be used is 1000m (3281 ft.)

2. Points to Check before Turning on the Power

(1) Close all the doors of the control enclosure and operation panel to prevent the entry of water, chips, and dust.

(2) Make absolutely sure that there is nobody near the moving parts of the machine, and that there are no obstacles around the machine, before starting machine operation.

(3) When turning on the power, turn on the main power disconnect switch first, then the CONTROL ON switch on the operation panel.

3. Precautions Related to Manual/Continuous Operation

(1) Follow the instruction manual during operation.

(2) Do not operate the machine with the front cover, check cover, or another protective cover removed.

(3) Close the front cover before starting the machine.

(4) When machining the initial work piece, check for machine operations, run the machine under no load to check for interference among components, cut the work piece in the single block mode, and then start continuous operation.

(5) Ensure your safety before rotating the spindle or moving a machine part.

(6) Do not touch chips or work piece while the spindle is rotating.

(7) Do not stop a rotating part with hand or another means.

(8) Check that the condition of hydraulic chuck jaws as mounted, operating pressure, and maximum permissible revolving speed.

(9) Check the condition and location of the cutting tool as mounted.

(10) Check the tool offset value.

(11) Check the zero offset value.

(12) Check that the SPINDLE OVERRIDE and FEEDRATE OVERRIDE dials on the NC operation panel are set to 100%.

(13) When moving the turret, check the software limits for X- and Z-axes or the locations of limit switch dogs to prevent interference with the chuck and tailstock.

(14) Check the location of the turret.

(15) Check the location of the tailstock.

(16) Cut work pieces with a transmitted power and torque within the permissible range.

(17) Check each work piece firmly.

(18) Check that the coolant nozzle is properly located.

4. On Finishing Work

(1) On finishing work, clean the vicinity of the machine.

(2) Return the turret to the predetermined retraction position.

(3) Always turn off the power of the machine before leaving it.

(4) To turn off the power, turn off the CONTROL ON switch on the operation panel first, then the main power disconnect switch.

5. Precautions Applicable during Maintenance Inspection and When Trouble Occurs

In order to prevent unforeseen accidents, damage to the machine, etc., it is essential to observe the following points when performing maintenance inspections or during checking when trouble has occurred.

(1) When trouble occurs, press the emergency stop button on the operation panel to stop the machine.

(2) Consult the person responsible for maintenance to determine what corrective measures need to be taken.

(3) If two or more persons must work together, establish signals so that they can communicate to confirm safety before proceeding to each new step.

(4) Use only the specified replacement parts and fuses.

(5) Always turn the power off before starting inspection or changing parts.

(6) When parts are removed during inspection or repair work, always replace them as they were and secure them properly with their screws, etc.

(7) When carrying out inspections in which measuring instruments are used - for example voltage checks - make sure the instrument is properly calibrated.

(8) Do not keep combustible materials or metals inside the control enclosure or terminal box.

(9) Check that cables and wires are free of damage: damaged cables and wires will cause current leakage and electric shocks.

(10) Maintenance inside the control enclosure

   a) Switch the main power disconnect switch OFF before opening the control enclosure door.

   b) Even when the main power disconnect switch is OFF, there may some residual charge in the MCS drive unit (servo/spindle), and for this reason only service personnel are permitted to perform any work on this unit. Even then, they must observe the following precautions.

   - MCS drive unit (servo/spindle)
     The residual voltage discharges two minutes after the main switch is turned OFF.

   c) The control enclosure contains the NC unit, and the NC unit has a printed circuit board whose memory stores the machining programs, parameters, etc. In order to ensure that the contents of this memory will be retained even when the power is switched off, the memory is supplied with power by a battery. Depending on how the printed circuit boards are handled, the contents of the memory may be destroyed and for this reason only service personnel should handle these boards.

(11) Periodic inspection of the control enclosure

    a. Cleaning the Cooling Unit

        The cooling unit in the door of the control enclosure serves to prevent excessive temperature rise inside the control enclosure and increase the reliability of the NC unit. Inspect the following points every three months.

- Is the fan motor inside the cooling unit working?
  The motor is normal if there is a strong draft from the unit.

- Is the external air inlet blocked?
  If it is blocked, clean it with compressed air

6. General Precautions

(1) Keep the vicinity of the machine clean and tidy.

(2) Wear appropriate clothing while working, and follow the instructions of someone with sufficient training.

(3) Make sure that your clothes and hair cannot become entangled in the machine. Machine operators must wear safety equipment such as safety shoes and goggles.

(4) Machine operators must read the instruction manual carefully and make sure of the correct procedure before operating machine.

(5) Memorize the position of the emergency stop button so that you can press it immediately at any time and from any position.

(6) Do not access the inside of the control panel, transformer, motor, etc., since they contain high voltage terminals and other components which are extremely dangerous.

(7) If two or more persons must work together, establish signals so that they can communicate to confirm safety before proceeding to each new step.

PREFACE

This User's Manual is intended for both users and developers. Before attempting to use the application or develop compatible software, please thoroughly read the relevant sections of this manual to ensure correct use

Okuma America Corporation was the first machine tool manufacturer to embrace an open architecture CNC control. We are confident that the ability for third parties to write their own software for use on the Okuma OSP control will result in an abundance of applications being created. OSP Widget Manager was created as means to organize and standardize the presentation of some of those applications.

This remarkable ability, enabling anyone's software to run on the machine control and interface with the machine directly is something feared by other machine tool builders. They have various reasons, but perhaps chief among them is the loss of control. They actively seek to places limits on the consumers of their machines. Okuma has taken the opposite approach.

Okuma strives to make it easy for our customers and partners to expand the capabilities of our machine tools to meet the exact needs of our customers. It is my sincere hope that this application will serve to further that goal.

REVISION HISTORY

Date	Version	Description	Author
10/10/2014	S5086-005-00	OSP Widget Manager Initial release	Scott Solmer

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

This manual provides instruction for installing and uninstalling, operating, and troubleshooting of the application. Additionally, information is provided to enable developers to write applications that conform to the requirements necessary for interoperability.

1.2 Definitions, Acronyms, and Abbreviations

From this point on, the "OSP Widget Manager" application may be referred to simply as "OWM". For additional definitions of terminology, refer to the glossary in section 10.

1.3 Overview

When Okuma first came out with the large format nineteen inch touch panel displays, it got our engineers thinking. How can we make the best use of this additional space? It is nice to have a view of the desktop for quick access to shortcuts, but what about running applications? The space available is quite limited. Sure, developers can design their applications to fit in the new space, but once the user has one or two of the applications running, the space will be filled. Okuma Widget Manager™ was designed to solve this problem by giving the user a quick and natural feeling interface to scroll through multiple running apps.

2. REQUIRED HARDWARE AND SOFTWARE

2.1 Hardware

Although this software is intended solely for use on Okuma machines with the large format 19 inch touch panels, it will also run on any Windows PC with a horizontal screen resolution of 1280 pixels or greater. This was intentional so that developers may use their PCs as a test environment.

2.2 Software

- *Windows®*
  OWM has been tested and will run on Windows XP SP3 and Windows 7.

- *THINC API*
  There is no THINC API requirement.

- *THINC Startup Service*
  To auto-launch OSP Widget Manager, version 3.2.1.0 or greater of the THINC Startup Service is required.

- *.NET*
  Microsoft® .NET Framework version 4.0 or greater is required for operation. It is bundled with the installer for OSP Widget Manager and will be installed, if required, during the installation of OSP Widget Manager.

- *PDF*

  Adobe® Acrobat® Reader® is required to view this Install / User Manual. It is to be bundled with the OSP Widget Manager installer and may be installed if required.

3. OSP WIDGET MANAGER INSTALLER

3.1 Install

To install OSP Widget Manager, run the setup file "Setup_OSP Widget Manager_v1.0.0.exe" from the instillation media or the download directory if obtained from the Okuma App Store. Follow the prompts of the install wizard as seen in **Figure 3-1**.

Figure 3-1

Once installation is completed successfully, OWM can be started by using any method in section 4.1.

3.2 Uninstall

To uninstall OWM, find the application in the Windows 'Add / Remove Programs' window (XP) or the 'Programs and Features' window (Windows 7) as seen in **Figure 3-2**. Select 'OSP Widget Manager' from the list and double-click it, or click the uninstall button.
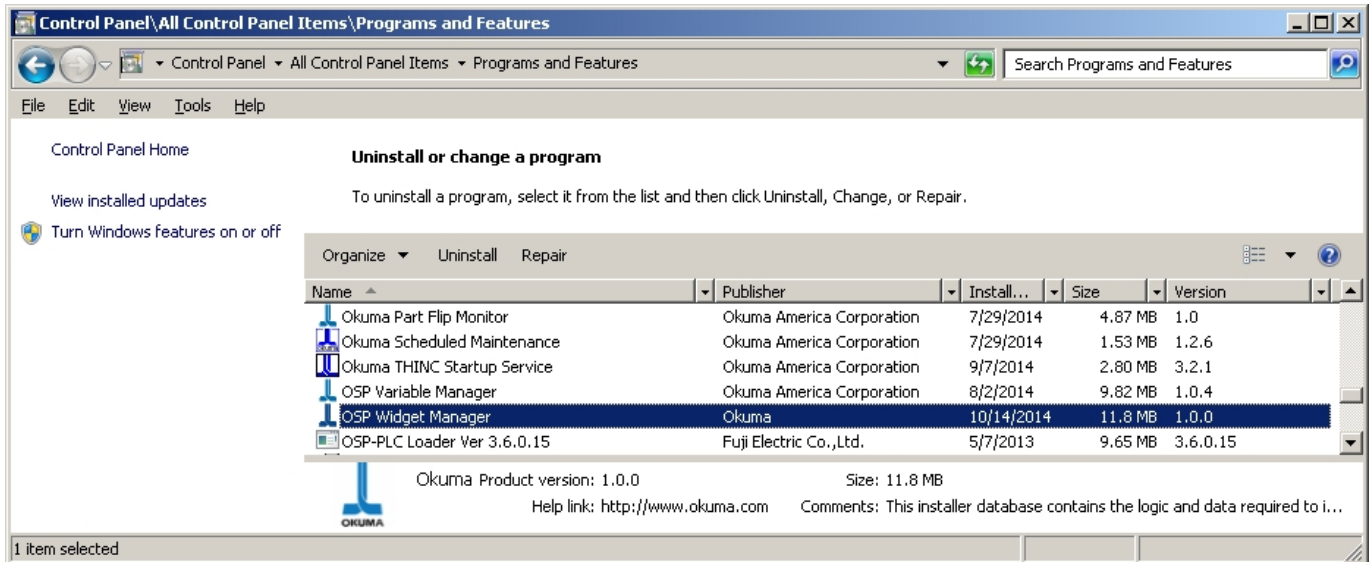
Figure 3-2

After clicking 'Yes' to an uninstall confirmation dialog, uninstallation will proceed automatically.

4. STARTING AND STOPPING THE APPLICATION

4.1 Start the Application

OSP Widget Manager can be started at least 4 different ways.

- ***Automatic Startup***

     The OWM Installer registers with the THINC Startup Service / Okuma Startup System upon install, and if either if these applications is installed, OWM will start automatically when the control is powered ON, and the THINC API is available. Please refer to the documentation for the THINC Startup Service or Okuma Startup System respectively, for details.

- ***Start Menu***

     By default, the installation directory is 'D:\Program Files\Okuma\OSP Widget Manager\'. OWM may be started by running the executable file "OWM.exe" from this directory as seen below in **Figure 4-1**.
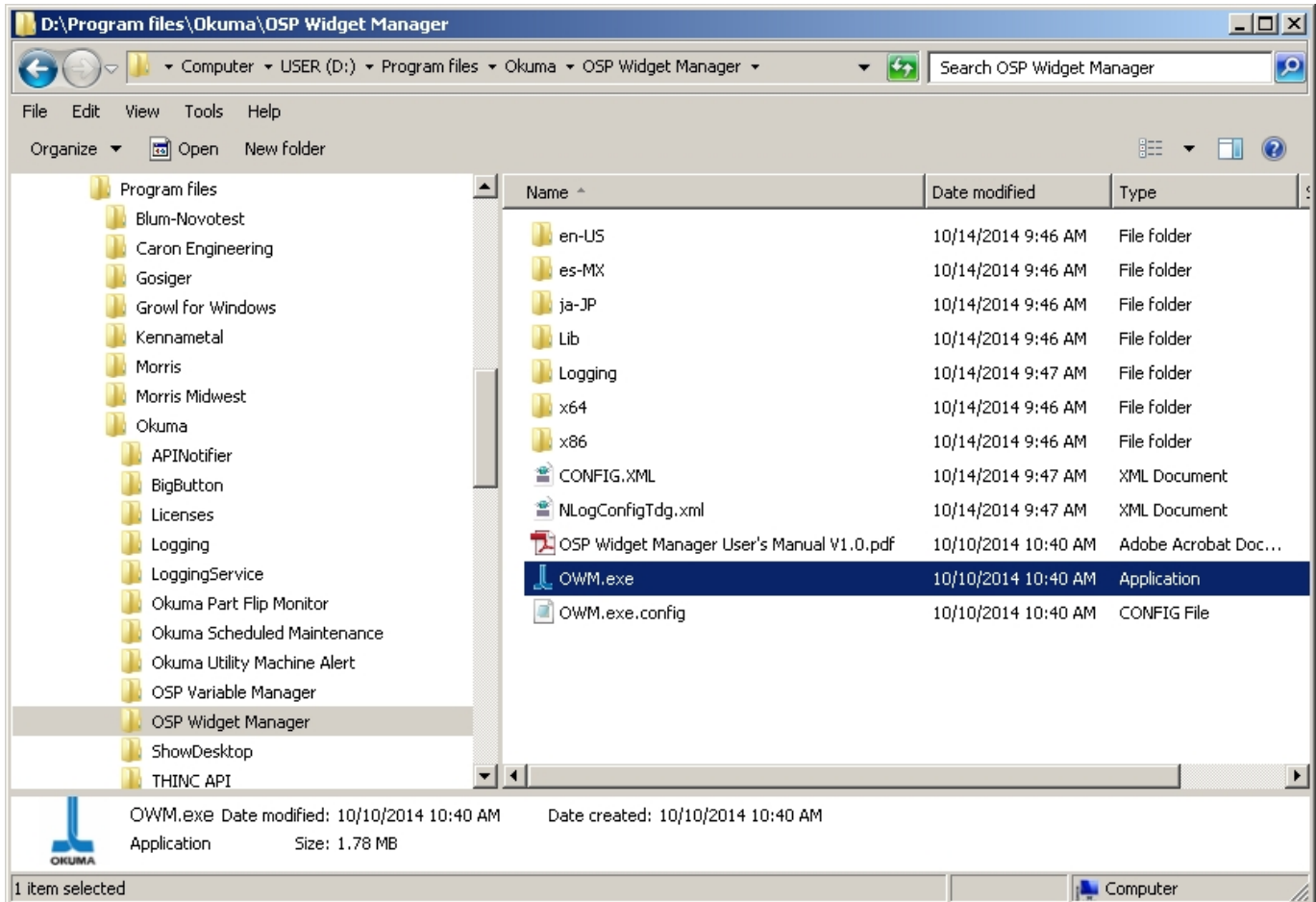
Figure 4-1

- ***Desktop Icon / Shortcut***
  Run the application by double clicking the desktop icon. Users may create their own shortcuts by right-clicking the desktop, selecting New, and Shortcut. Browse to the application's install directory and select the program "OWM.exe".

4.2 Stop the Application

There are at least three ways to stop the application.

- ***Main Tab***
  Press the "Exit" button in the Main tab of the OSP Widget Manager window.
  (Refer to **Figure 4-2**.)

- ***System Tray Menu***
  Click the Okuma icon in the notification area (system tray) and choose "Exit".
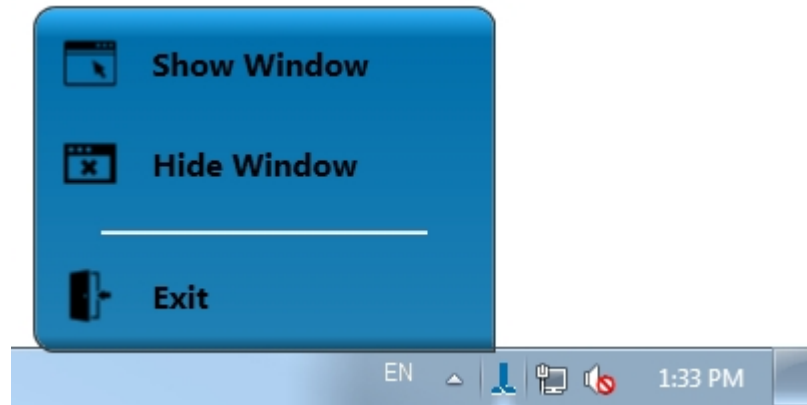
Figure 4-2

- *Shortcut / Command Line*

Run the application using the argument " **-exit**" after the executable name. Alternatively, create a shortcut with the same argument. Refer to **Figure 5-4**.

5. OPERATION

5.1 Scroll Modes

Due to technical reasons, it is currently **not possible** to scroll by making "gestures" which **begin** over widgets (in between the arrow buttons). However, by **beginning** interactions on the arrow buttons of OSP Widget Manager, various functions are available as described below.

(Drag) Figure 5-1   **(Index) Figure 5-2**   **(Scroll to End) Figure 5-3**

5.1.1 Drag

To drag, click and hold one of the arrow buttons and then move the pointer in the direction of the arrow. While dragging, the button will provide visual feedback in the form of a green highlight. Refer to **Figure 5-1**.

5.1.2 Flick

To trigger a flick action, perform a quick drag and release. If the speed at which dragging was taking place at the time of release was greater than a certain threshold, the action will be interpreted as a "flick". The result will be a scrolling motion that continues even after releasing.

5.1.3 Index

A quick tap / click and release will result in an index action. The result of this action is to scroll to the nearest widget's left corner. If the display is currently resting on the corner of a widget, it will scroll to the next widget. The index action will provide visual feedback in the form of a blue highlight. Refer to **Figure 5-2**.

5.1.4 Scroll to End

Double tapping / double clicking one of the arrow keys will result in a scroll to end action, whereby the display is scrolled to the far end (first or last widget, depending on the direction). The scroll to end action will provide visual feedback in the form of a red highlight.   Refer to **Figure 5-3**.

5.2 Hide The Application (minimize)

- *System Tray Menu*

  Press the "Hide Window" button in the system tray menu. (Refer to **Figure 4-2**)

- *Shortcut / Command Line*

  Run the application using the argument " **-hide**" after the executable name. Alternatively, create a shortcut with the same argument. Refer to **Figure 5-4**.

Figure 5-4

5.3 Show The Application (restore)

- *System Tray Menu*

  Press the "Show Window" button in the system tray menu. (Refer to **Figure 4-2**.)

- *Shortcut / Command Line*
  Run the application using the argument " **-show**" after the executable name.
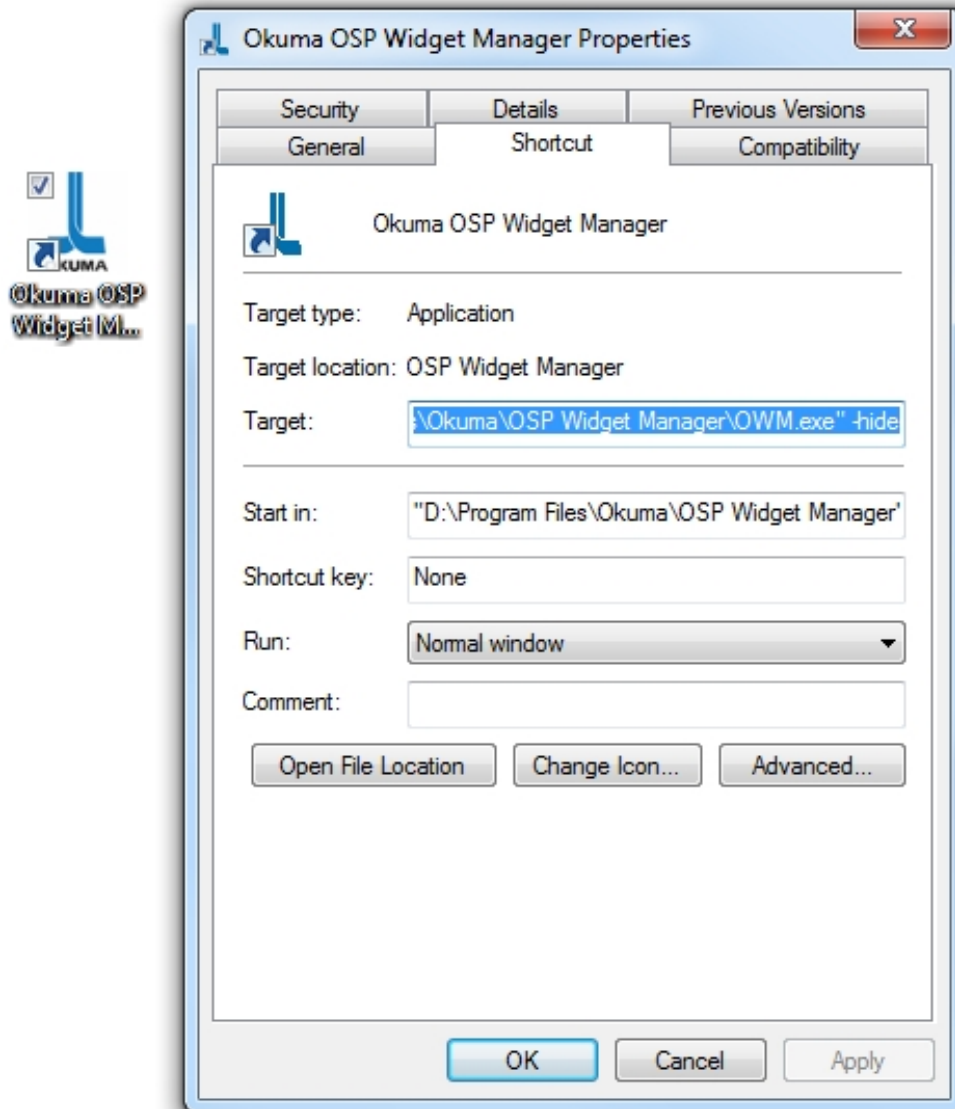Alternatively, create a shortcut with the same argument. Refer to **Figure 5-4**.

5.4 Add Widgets to OSP Widget Manager

No action is required by the user to get a widget into OSP Widget Manager other than running the desired program. OWM will automatically identify widget windows and pull them into the application.

Programs such as Notepad, Word, and Excel may be integrated into OWM by

opening a document whose name contains a widget identifier string. For additional details please refer to section 9.1 – Widget Identifier.

6. CONFIGURATION

6.1 Language

Changing the display language is accomplished by choosing a language from the drop-down menu of the OSP Widget Manager main window. (Refer to **Figure 6-1**) The interface text will change immediately, whereas only subsequent logging activity will appear in the new language.
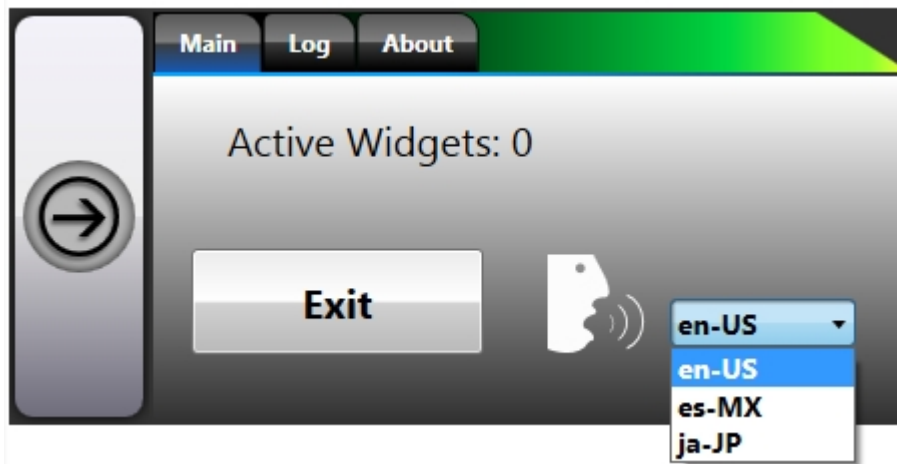

Figure 6-1

Language codes follow the format "Language – Country" using ISO 639-1 and 3166-2 respectively. At the time of writing, the following languages are supported:

- en-US   English – United States
- es-MX   Spanish – Mexico
- ja-JP   Japanese – Japan

6.2 Advanced Configuration

6.2.1 Enable Configuration Menu

In most cases it should not be necessary to make any changes to the configuration. However, the configuration settings for OWM can be accessed by opening the file "CONFIG.XML", which is located in the application directory. The contents of this file are shown in **Figure 6-2**.

Note that before OWM is run for the first time, CONFIG.XML may not exist. It is created the first time the program is run, and will be recreated if it is missing.

It is not recommended to change any values in this file manually, with the exception of "SHOW_CONFIG". This setting may be changed to "true" to give access to an in-application menu.

```xml
CONFIG.XML* ×
    <?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Settings>
    <Localization Language="en-US" />
    <LoggingOptions Verbose="false"
                    EnableWindowsEventLog="false"
                    EnableTextLog="true"
                    EnableDatabaseLog="false"
                    EnableCsvLog="false"
                    EnableTdgCompactDataLog="false"
                    SuppressLoggingErrors="false" />
    <ScrollVars DRAG_THRESHOLD="10"
                SCROLLING_UPDATES_LAYOUT="false"
                SOFT_EDGE_FLICKS="false"
                FLICK_FRAMES="50"
                FLICK_TICK="20"
                SCROLLFAR_TICK="50"
                INDEX_TICK="50"
                DRAG_TICK="50"
                BANDING_TICK="50" />
    <Constants HeightXUU="208"
               WIDGET_IDENTIFIER="11A11A2B1B2B3"
               VELO_CALC_POINT_LIFE="80"
               DARK_MAGIC_POLLING_INTERVAL="1000"
               SHOW_STATUS_TAB="false"
               MAX_LOG_ITEMS="100"
               DEBUG_MODE="false"
               SHOW_CONFIG="false" />
</Settings>
```

Figure 6-2

If 'SHOW_CONFIG` is set to ="true" a new button will appear in the Main window that opens a configuration window. (Refer to **Figure 6-3**). Changing this setting requires a restart of the application to take effect. Please note again, that it is recommended to leave all settings at their default values.
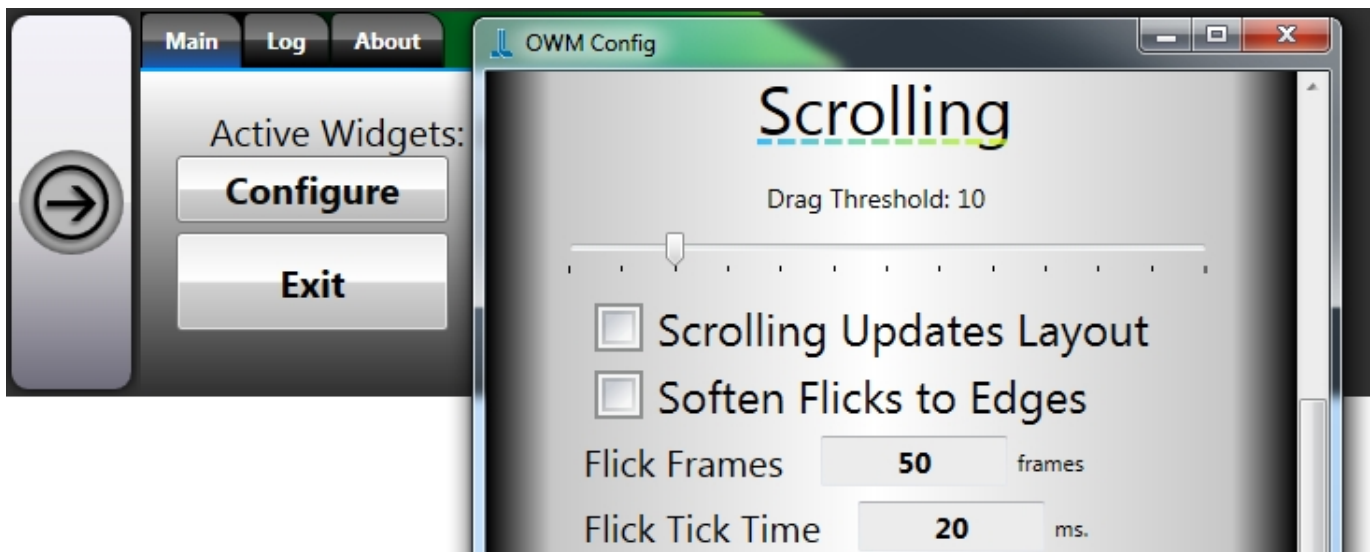

Figure 6-3

6.2.2 Enable Capture of Widgets

When checked, OSP Widget Manager will search for running widgets at a regular interval, grab them if they are found, and place them into its interface.

6.2.3 "Demo" Overlay Enabled

This setting shows the message "DEMONSTRATION ONLY" over the widget area while scrolling.

6.2.4 Logging

- **Verbose:**

  Enabling this setting shows all possible logging messages. When unchecked, only errors and warnings, and some informational messages are logged.

- **Enable Windows Event Log:**

  This setting creates Events in Windows Event Viewer.
  (Under *Control Panel, Administrative Tools, Event Viewer*)

- **Enable Text Log:**

  This setting creates a text file "OWM.txt" in the Logging directory.

- **Enable Database Log:**

  This setting creates a SQLite database file "OWM.sl3" in the logging directory.

- **Enable CSV Log:**

  This setting creates a CSV file "OWM.csv" in the Logging directory.

- **Compact Datalog:**

  Enabling this feature of the TDG logging DLL will attempt to compress old log files.

- **Suppress Logging Errors:**

  This function will prevent any exceptions thrown by the TDG logging DLL from propagating up to the main application.

6.2.5 Scrolling

- **Drag Threshold:**

  This is the number of pixels the cursor must move after MOUSE DOWN on one of the scroll buttons before dragging begins.

- **Scrolling Updates Layout:**

  Calls the method UpdateLayout after scrolling; this may or may not affect the appearance of scrolling and / or performance.
  Refer to: http://msdn.microsoft.com/en-us/library/system.windows.uielement.updatelayout%28v=vs.110%29.aspx

- **Soften Flicks to Edges:**

This setting extends the rubber-banding effect when flicked to the edge of the scrollable area.

- **Flick Frames:**
  Determines how long (in number of frames) scrolling animation lasts during flick.

- **Flick Tick Time:**
  This setting determines the amount of time between animation frames during flick.

- **Scroll Far Tick Time:**
  This setting determines the time between animation frames during scroll to end.

- **Index Tick Time:**
  This determines the time between animation frames during scroll to next widget.

- **Drag Tick Time:**
  This setting determines the amount of time between animation frames during drag.

- **Rubberband Tick Time:**
  This setting determines the time between animation frames during rubber-banding.

6.2.6 Constants

- **Window Height:**
  This setting resizes OWM in the vertical direction, from the bottom.

- **Widget Identifier:**
  This setting specified the partial contents of a window title / name that OSP Widget Manager will look for, in order to identify applications as widgets.
  Note: ***DO NOT MODIFY THIS SETTING!***

- **Velo Calc Point Life:**
  In order to calculate flick velocity, OWM gets a list of mouse pointer locations with time stamps from the operating system. This setting determines the cut-off point where the time stamps are too old to be included in the velocity calculation.

- **DM Polling Interval:**
  The frequency with which OWM looks at all running processes on the system in order to identify and grab widgets.

- **Show Status Tab:**
  The status tab displays information about OWM button user interactions and is a remnant of the development process.

- **Max Log Items Display:**
  Determines the number of line items to display in the Log tab before overwriting entries contained there.

Note: this does not affect the logging of information to files using the TDG logging DLL.

- **Debug Mode:**
    This setting will create log entries related to interactions with the arrow buttons. Additionally, a pop-up window with flick velocity will be shown when a flick is triggered.

6.3 Backup Configuration Settings

Copy the file "CONFIG.XML", located in the install directory, and store it in a safe location.

6.4 Restore Configuration Settings

Exit the application and copy the backup "CONFIG.XML" file into the application install directory. If prompted to overwrite an existing file, accept.

6.5 Restoring Default Configuration

Exit the application, delete the file "CONFIG.XML" located in the install directory, and restart the application. A new file will be created with default settings automatically.

7. TROUBLESHOOTING

7.1 Program Not Responding

In the event that OWM should stop responding or "crash", first try restarting the application. If the problem is recurring, please send an email to API@Okuma.com, include the most recent logging file (located in the logging directory under the application's install folder) and describe the events that led to the problem with as much detail as possible.

7.2 Program is Slow or Unresponsive

Under idle conditions, Program Name should use very little system resources. Only during scrolling should the application use any noticeable amount of CPU cycles. If scrolling seems jittery, jumpy, or slow, check the idle CPU resource consumption using the Windows Task Manager's Performance tab (Ctrl-Alt-Del). If the CPU is showing a solid 100% usage, attempt to determine what process is responsible for consuming most of the resources before contacting Okuma.

Approximately 30% CPU consumption is typical during the brief period scrolling is taking place. This is equivalent to scrolling though files using Windows® Explorer. While not scrolling, OWM should consume 0%, and only intermittently show 1 or 2 percent (during DM polling).

7.3 Known Bugs

- There are no known bugs at the time of this writing.

7.4 Other Issues

For all other issues, please contact the Okuma America Software Development group by using the following email address: API@okuma.com

8. METHOD TO START NC IN WINDOWS ONLY MODE

1) Power ON the OSP Control (Power OFF the control first if already running)

2) When the OSP Loading screen shows, touch the screen anywhere to ensure it is selected

3) Press the ESC key to reveal a button labeled "Cancel"

4) Click the Cancel button.
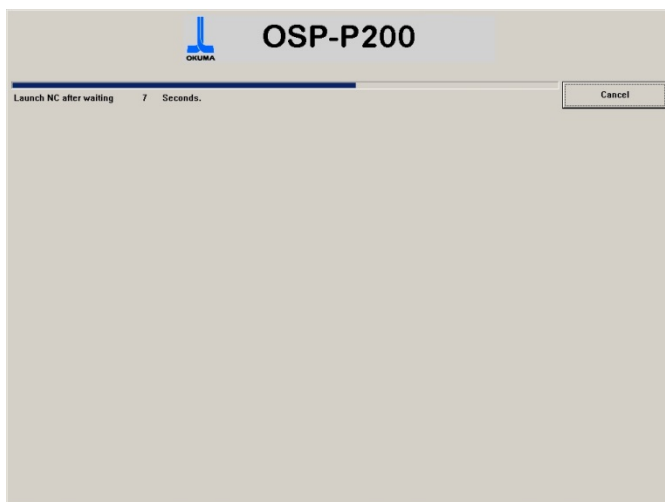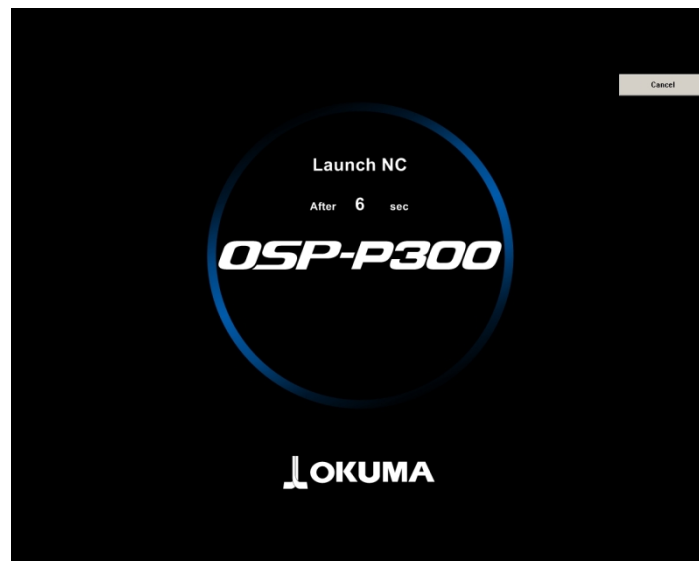
9. DEVELOPERS

The following information is provided for developers seeking to create applications (widgets) compatible with OSP Widget Manager.

9.1 Widget Identifier

OSP Widget Manager is able to find and 'grab' widgets by looking at the title or text of all open windows on the system. If the title contains one of the following strings, it will be identified as a widget:

- 11A11A2B1B2B31U
- 11A11A2B1B2B32U
- 11A11A2B1B2B33U

For applications written using WPF the setting is Window "Title".
For applications written using Windows Forms, the setting is Form "Text".

The 3 different string endings (1U, 2U, and 3U) identify the widget size, as described in section 9.2. The title / text can include other text so long as it *contains* the widget identifier.

9.2 Widget Sizes

Three standard window sizes are allowed, referred to as UU size (unit utilization). The widget display area measures 1140 x 208 pixels. We then divided this by 3 to give the base unit of 380 x 208. Thus, up to three 1UU widgets may be displayed in their entirety, simultaneously.

- Small (1UU): 380W x 208H
- Medium (2UU): 760W x 208H
- Large (3UU): 1140W x 208H

9.3 Widget Window Mode

Widget windows should start in "normal" mode (not minimized, maximized, or hidden). Furthermore they must not have a title bar. In WPF this is accomplished by making the setting WindowStyle="None". Using Windows Forms, set FormBorderStyle to "None".

9.4 Miscellaneous Requirements

The user must not be able to move, resize, or execute window commands such as minimize or maximize. Minimizing, maximizing, and resizing can be prevented through settings. To prevent moving, it might be necessary to disable the Move command in the title bar menu. This menu (typically opened by right-clicking the title bar of an application) might still be able to be opened even when no title bar is present by using the key combination "Alt-Space".

Consider either handling the key combination to prevent the menu from showing or disabling the menu item. Refer to the C# example code in section 11.1.

9.5 Other Recommendations

9.5.1 Multiple Interfaces

Seeing as how it would be undesirable to present the user with a window they cannot move or minimize when OWM is not running; it is recommended to create two separate GUI windows where one is the OWM widget style and the other is a standard window. The program can then decide which one to start based on if OSP Widget Manager is running or not. Refer to C# example code in section 11.2.

9.5.2 Window Border

As a purely optional stylistic choice, developers of widgets can make their applications "feel at home" in OWM by making their interface background color R:50, G:50, B:50, A:100 (#FF323232). For example, refer to **Figure 9-1**.

Figure 9-1

9.5.3 If Widget Window Other Than Main Window

In the case that the widget window is not the application's main window; there are a few precautions to note.

- To pass information between the main application and the widget, **Events** or other thread-safe measures should be used. Do *NOT* use form / window instance referencing.
  For example: *(from main window)* "WidgetWindowInstance.SomeVariable = SomeValue" will fail.

- Once opened, do not allow the widget window to be closed until the entire application is closed. OWM watches the widget *process* in order to know if it has been closed, and thus can be removed from the display. If the process continues to run when the widget window has been closed, OSP Widget Manager will not detect this, and a blank space will be left in the display.

- Do not rely on the parent / child relationship of the widget window to the main application. When OSP Widget Manager 'grabs' the widget window, its parent will become OWM.

10. GLOSSARY

API: Application Programming Interface. In the scope of this document, it is a short form for 'THINC API'

ESC: The Escape key. It is represented by the following symbol on Okuma controls:

GUI: Graphical User Interface

NC: Numerical Control

OSP: "Okuma Sampling Path" intelligent control

OWM: OSP Widget Manager

THINC API: The Intelligent Numerical Control Application Programming Interface, used to access machine information via software applications.

.NET Framework: The Microsoft® common language runtime environment for applications developed with Visual Studio®. This is required to run the Lathe Interface software application.

Widget: The generic name for applications designed to run inside OWM.

WPF: Windows Presentation Foundation, a .NET programming language utilizing XAML

UU: Unit Utilization, referring to standard widget sizes

11. EXAMPLE CODE

11.1 Prevent Window from being moved (Windows Forms)

```
#region Prevent User From Moving Window Snippet
[DllImport("user32.dll")]
private static extern int EnableMenuItem(System.IntPtr hMenu,
int uIDEnableItem, int uEnable);

private const int HTCAPTION = 0x00000002;
private const int MF_BYCOMMAND = 0x00000000;
private const int MF_ENABLED = 0x00000000;
private const int MF_GRAYED = 0x00000001;
private const int MF_DISABLED = 0x00000002;

private const int SC_MOVE = 0xF010;
private const int WM_NCLBUTTONDOWN = 0xA1;
private const int WM_SYSCOMMAND = 0x112;
private const int WM_INITMENUPOPUP = 0x117;
private const int WM_NCLBUTTONDBLCLK = 0xA3;

private bool bMoveable = true;

[System.ComponentModel.Category("Behavior"), System.ComponentModel.Description("Allows the
```

```
form to be moved")]
    protected virtual bool Moveable
    {
        get { return this.bMoveable; }
        set { this.bMoveable = value; }
    }

    protected override void WndProc(ref System.Windows.Forms.Message m)
    {
        if (m.Msg == WM_INITMENUPOPUP)
        {
            //handles popup of system menu
            if (m.LParam.ToInt32() / 65536 != 0) //divide by 65536 to get hiword
            {
                int AbleFlags = MF_ENABLED;
                if (!Moveable)
                {
                    AbleFlags = MF_DISABLED | MF_GRAYED;
                    EnableMenuItem(m.WParam, SC_MOVE, MF_BYCOMMAND | AbleFlags);
                }
            }
        }

        if (!Moveable)
        {
            if (m.Msg == WM_NCLBUTTONDOWN)
            {
                //cancels any attempt to drag the window by it's caption
                if (m.WParam.ToInt32() == HTCAPTION)
                { return; }
            }
            if (m.Msg == WM_SYSCOMMAND)
            {
                //redundant but cancels any clicks on the Move system menu item
                if ((m.WParam.ToInt32() & 0xFFF0) == SC_MOVE)
                { return; }
                if ((m.Msg == WM_NCLBUTTONDBLCLK))  //Try to stop double click on title bar
                { return; }
            }
        }
        //return control to base message handler
        base.WndProc(ref m);
    }

    #endregion end of "Prevent user from moving window" snippet
```

11.2 Multiple User Interfaces

```
using System.Diagnostics;

/// <summary>
/// The main entry point for the application.
/// </summary>
[STAThread]
static void Main()
{
   Process[] WiManMode = Process.GetProcessesByName("OWM");
   if (WiManMode.Length == 0) {
      Application.EnableVisualStyles();
      Application.SetCompatibleTextRenderingDefault(false);
      Application.Run(new MainMenu()); // Standard Window
   }
   else {
      Application.EnableVisualStyles();
      Application.SetCompatibleTextRenderingDefault(false);
      Application.Run(new WidgetMenu()); // Widget Window
   }
}
```

12. DISCLAIMER OF WARRANTY