MPLAB® Harmony Help - MHC & MHGC User's Guides

MPLAB Harmony Integrated Software Framework v1.11

MPLAB Harmony Graphics Composer User's Guide

This section provides user information on using the MPLAB Harmony Graphics Composer.

Introduction

This user's guide provides information on the MPLAB Harmony Graphics Composer (also referred to as the graphics composer), which is included in your installation of MPLAB Harmony.

Description

The MPLAB Harmony Graphics Composer is a graphics user interface design tool that is integrated as part of the MPLAB Harmony Configurator (MHC). This tool allows a user to easily configure and visually design for the MPLAB Harmony Graphics Primitive Library and the MPLAB Harmony Graphics Object Layer.

The overall development flow of Composer consists of:

- Import image and font assets
- Create screens and schemes
- Add objects to screens
- Configure objects
- Generate MHC configuration
- Upload program to device

Glossary of Terms

Throughout this user's guide the following terms are used:

Acronym or Term	Description
Action	A specific task to perform when an event occurs.
Asset	An image, font, or binary data blob that is used by a user interface.
Event	A notification that a specific occurrence has taken place.
Object	An abstract term defining an entity that resides in a user interface screen.
Primitive	An object that represents a Graphics Primitive Library object.
Resolution	The size of the target device screen in pixels.
Screen	A discreet presentation of organized objects.
Tool	An interface used to create objects.
UI	Abbreviation for User Interface.
Widget	An object that represents a Graphics Object Library (GOL) widget.

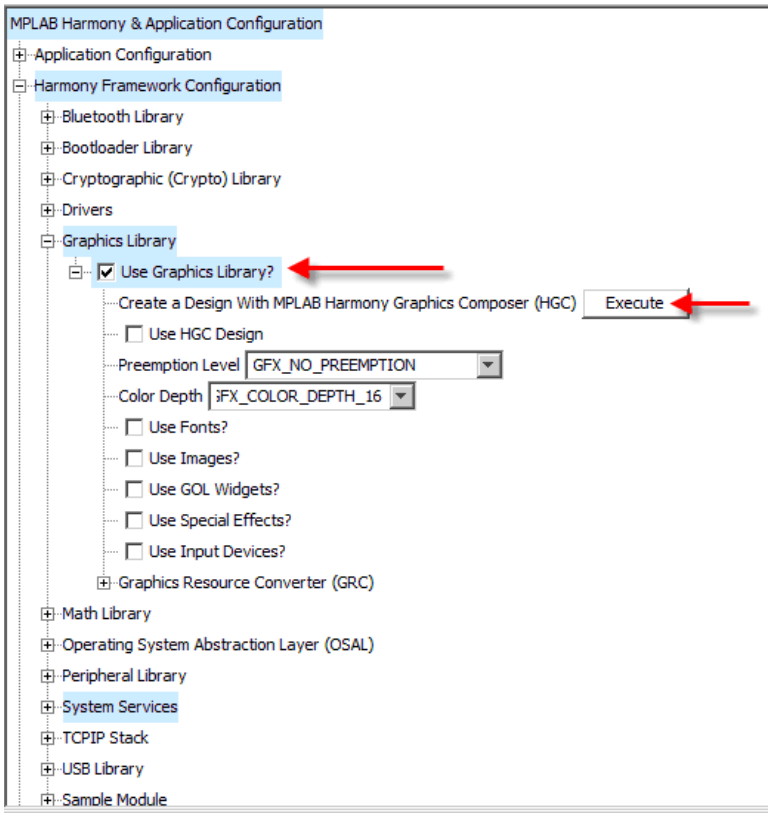
Getting Started

This topic provides information on getting started with the graphics composer.

Description

To begin using the graphics composer, which is part of the MPLAB Harmony Configurator (MHC), you will need to create a new MPLAB Harmony project and select a PIC32 device that is graphics-capable. For example, your project could be named *composer_demo*. Once you've created your project do the following:

1. Open MPLAB Harmony Configurator.
2. In the Harmony Framework Configuration tree expand Graphics Library and select **Use Graphics Library**.
3. Next, click the **Execute** button located next to Create a Design With MPLAB Harmony Graphics Composer.



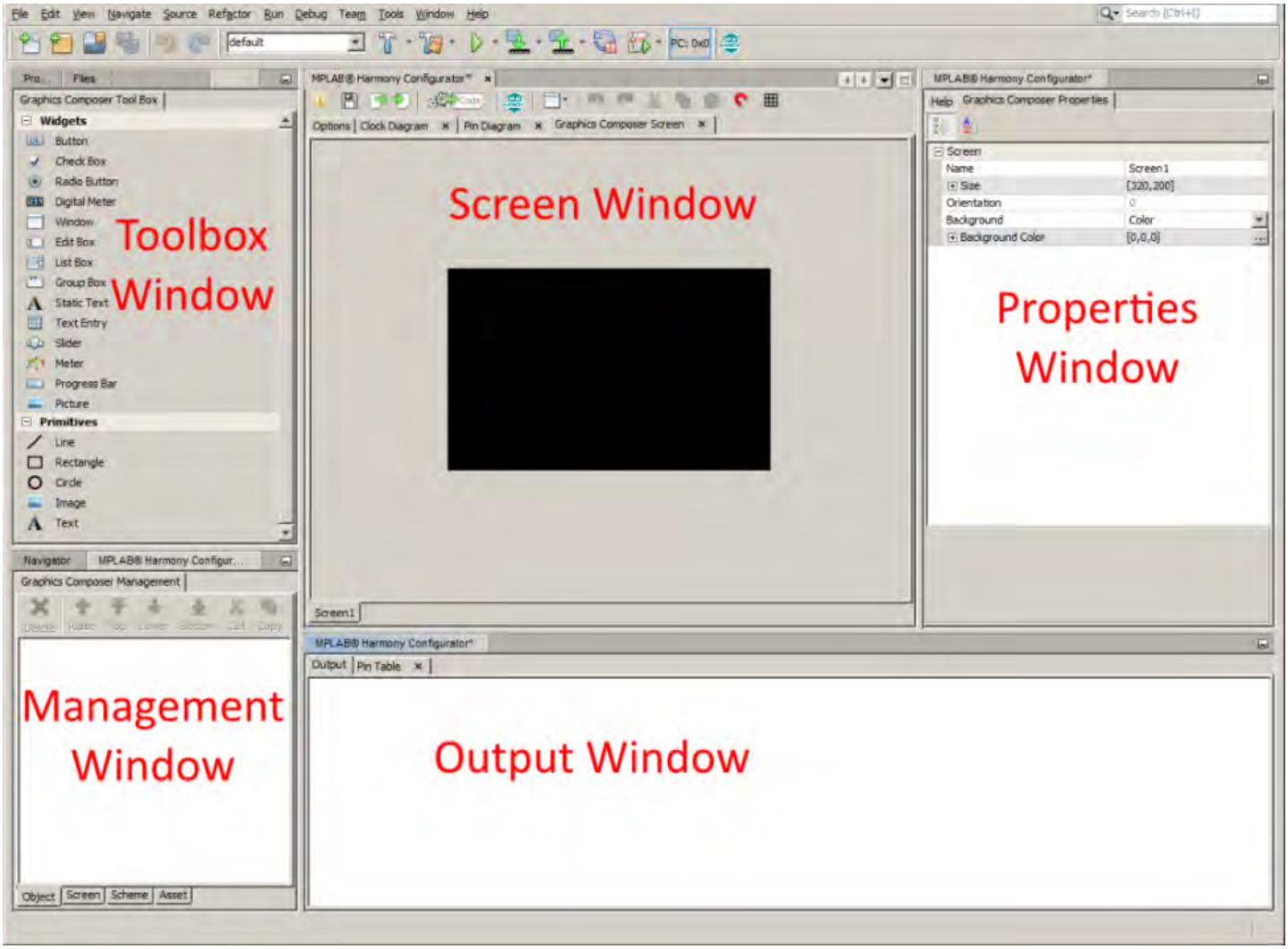
User Interface

This section describes the layout of the MPLAB Harmony Graphics Composer user interface.

Description

User Interface Layout

The following figure shows the initial user interface layout.



Object Toolbox

The Object Toolbox displays all of the available widgets and primitives to the user.

Composer Management Window

This window allows the user to manage objects, screens, schemes, and assets.

Screen Window

The screen window is the Graphics representation of how objects will appear when displayed on the device.

Properties Window

This window provides the user with the means to adjust properties for objects and screens.

Output Window

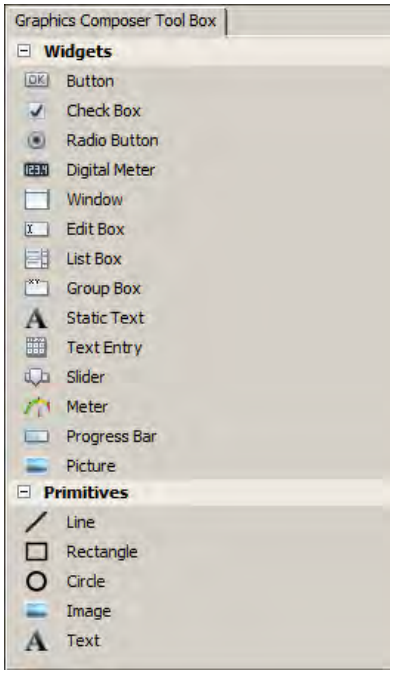
This window displays any output generated during your session.

Object Toolbox

Describes the features of the Object Toolbox.

Description

The Object Toolbox is the interface by which users add widgets and primitives into the screen representation. There are two primary methods for creating new objects: clicking and dragging.



Click Method

The following actions can be performed using the Click method:

- Clicking on an item selects it as active. Users can then move the cursor into the screen window and view a representation of the object about to be added.
- Left clicking confirms the placement of the new object
- Right clicking aborts object creation
- Clicking the active item again will deactivate it

Drag Method

Dragging and dropping a tool item into the Screen Window will also create a new instance of an object. When dragging a tool item, releasing the cursor outside of the Screen Window will cancel the drag operation.

Interactive Object Creation

The Primitives "Line" selection offers an interactive method for creating lines. Activating the Line primitive will open the Line Primitive Create tool. The user will then be prompted to create line points. Lines can be created using two discrete clicks or using a single click and drag operation. When creating the second line point, the <Shift> key can be pressed to lock to the X or Y axis of the first point.

Automatic Code Optimization

MPLAB Harmony Graphics Composer keeps track of the types of widgets that are used and updates the MHC Tree constantly to ensure only the Graphics Library code necessary for your design is included in the project.

Composer Management

This topic describes the features of the Composer Management window.

Description

The Composer Management window provides four tabs for graphics management.

- Object
- Screen
- Scheme
- Asset

Object Tab

Describes the features of the Object tab.

Description

The Object tab of the Composer Management window provides the capability to delete, select, and manage the placement of objects in the active screen.

The follow actions can be performed in the Object tab:

- Left clicking on an object will select it
- Left clicking no objects will clear a current selection
- Shift-Left Clicking will do a group select. Ctrl-Left click will perform a toggle select
- The Delete button will delete the selected objects
- The Raise, Top, Lower, and Bottom buttons control object placement in the list. Objects are drawn from the bottom up and higher objects will cover lower ones.

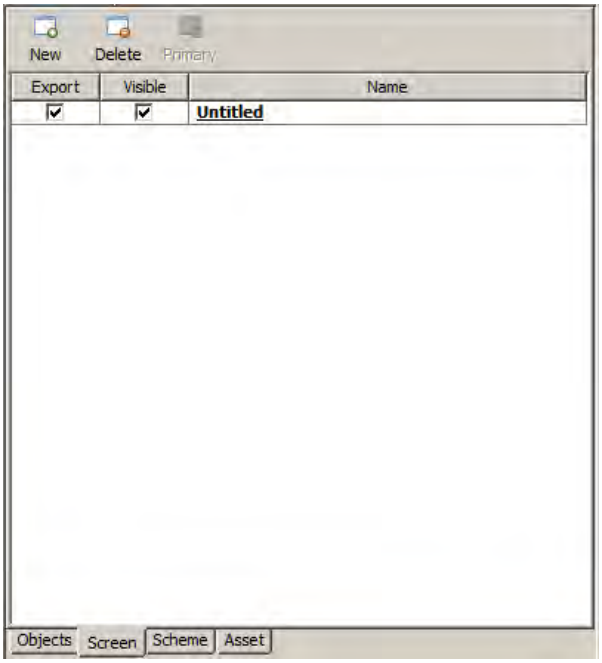
Note: The current Primitive Library implementation in the Graphics Library ignores this type of ordering. Currently, primitives are placed above widgets.

Screen Tab

Describes the features of the Screen tab.

Description

The screen management tab in the management window allows the user to create new screens, delete existing screens, and change some screen options.



Button Descriptions

The following selections are available:

- New – Creates a new screen. Note that screen names must be unique
- Delete – Deletes the selected screen. Screens can be selected by clicking on their row in the table
- Primary – Designates the selected screen as the primary screen. The primary screen is the screen that will be shown first when the UI is activated.

Table Descriptions

- Export – Controls whether the associated screen is exported when converting the project to code
- Visible – Controls whether the associated screen is visible on the screen window tab bar

Screen Status

Screen names may be in **Bold** type or underlined, which represent different screen states.

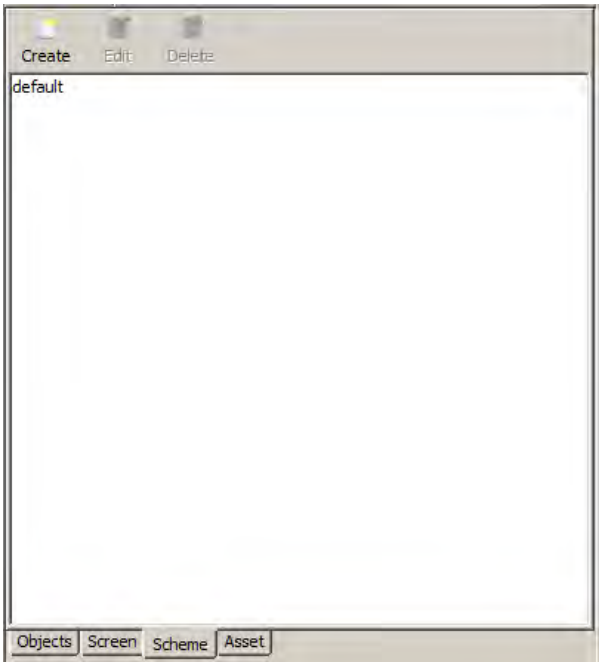
- Bold – The screen with the name in **Bold** type is the currently active screen in the screen window
- Underline – The underlined screen is designated as the primary screen

Scheme Tab

Describes the features of the Scheme tab.

Description

The Scheme tab of the management window allows for the management of display schemes.



Button descriptions

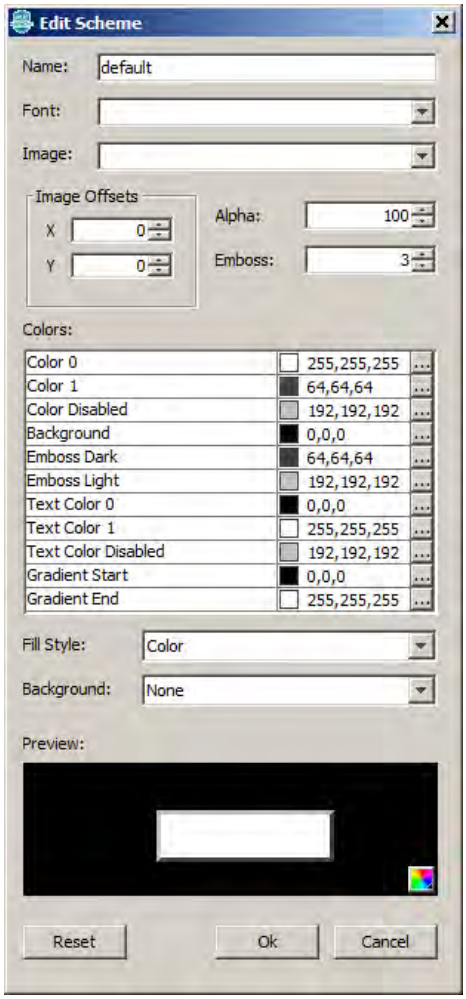
The following features are available:

- Create – Create a new display scheme. Scheme names must be unique.
- Edit – Edit an existing display scheme.
- Delete – Delete an existing display scheme.

Editing a Scheme

To edit an existing scheme, select the scheme from the list and click **Edit**. The Edit Scheme dialog appears, which allows the user to change various options associated with graphics display schemes.

- Font - This drop-down box allows the user to assign a font to this scheme. The box field is currently blank as no fonts have been imported into the graphics composer.
- Background Offsets - The background is to be offset by the specified X and Y coordinates
- Alpha - Defines the Alpha value
- Colors - Colors may be changed by selecting the corresponding ellipsis button
- Fill Style - Sets the fill style
- Background Type - Sets the background type
- Preview - The Preview window shows how the scheme would appear when applied to a button widget. The color box in the lower right corner of the Preview window allows the user to change the Preview window background.



Asset Tab

Describes the features of the Asset tab.

Description

This interface allows the user to import and convert images, fonts, and binary data into assets that the graphics composer will display and output during code generation. Users experienced with the Graphics Resource Converter (GRC) utility will be familiar with these functions.

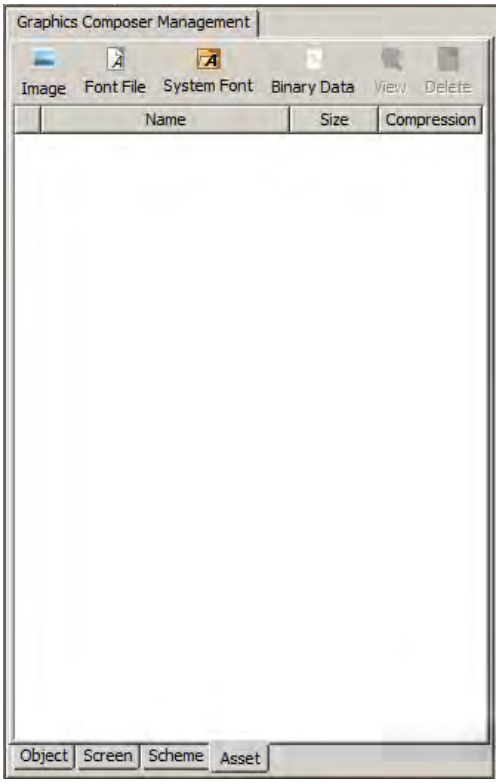
Imported assets are stored in a binary format file named `asset.cache`. This file resides in `firmware\src\system_config\$CONFIGURATION_NAME`. If this file deleted, all imported assets will be unavailable and must be imported again.

Button Descriptions

Note: Beginning in MPLAB Harmony v1.05, the HConfig tree-based Graphics Resource Converter (GRC) interface has been removed, and the Asset Tab is the only available integrated method for importing fonts and images.

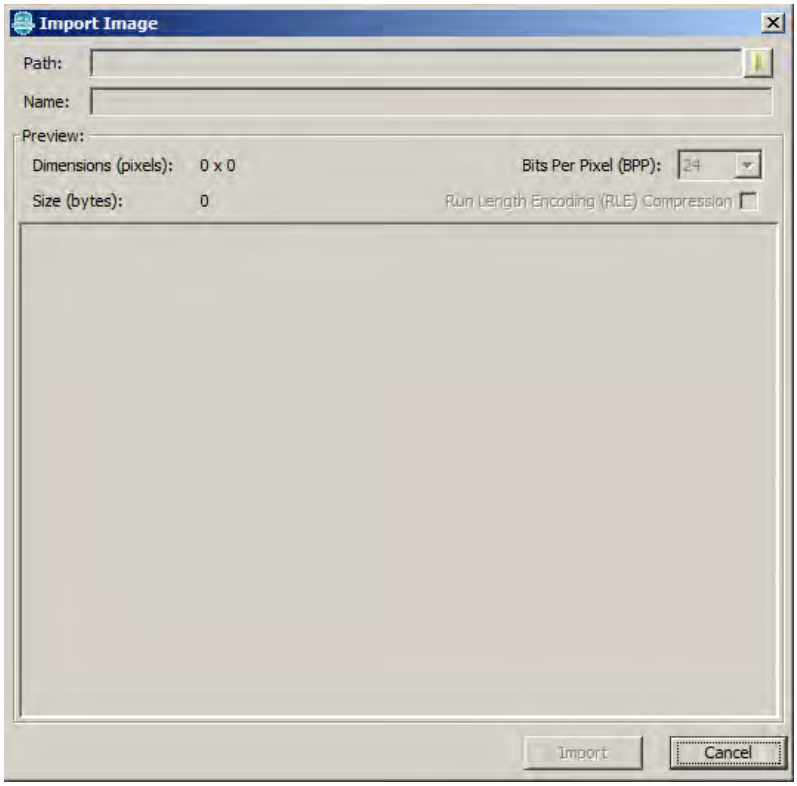
The following selections are available:

- Image – Opens the Import Image dialog
- Font File – Opens the Import Font File dialog
- System Font – Opens the Import System Font dialog
- Binary Data – Opens the Import Binary Data dialog



Importing Images

Open the Import Image dialog by clicking **Image**.

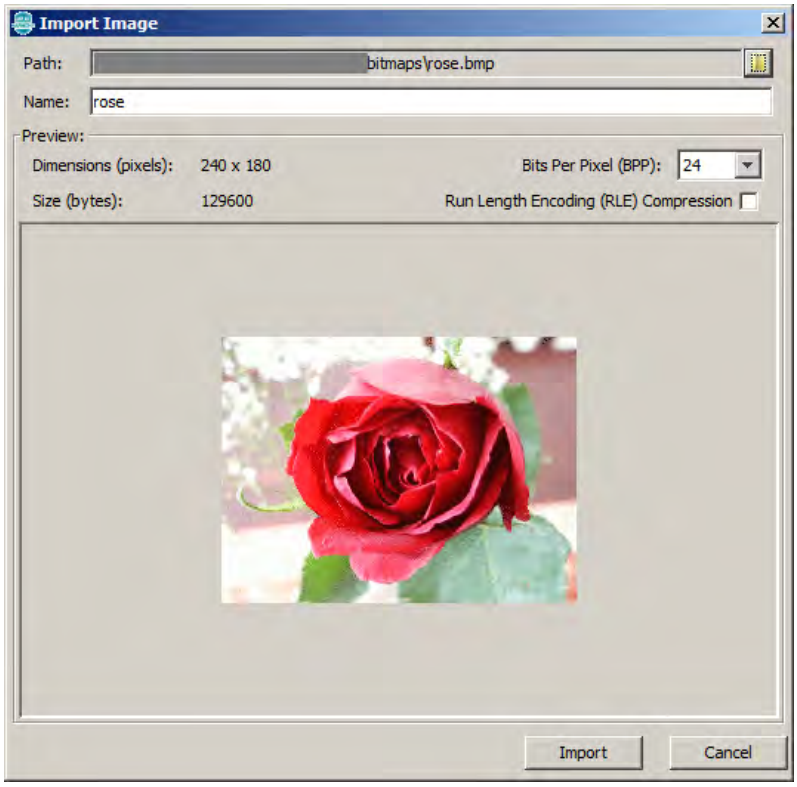


Click the Path: Browse icon and navigate to an image. The graphics composer supports sourcing from all image formats that are natively supported by Java. To be specific, all formats will convert to 16-bpp BMP with the exception of JPEG, which is supported by the JPEG decoder at runtime, and therefore, do not require conversion.

Auto-Configuration

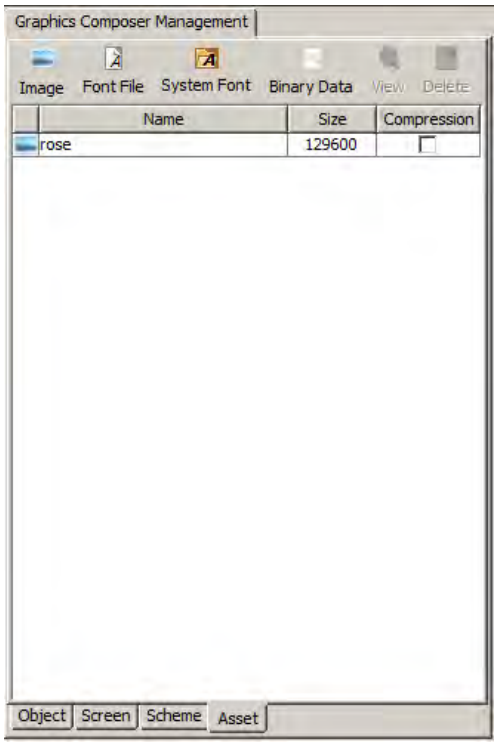
The graphics composer will detect that a JPEG asset has been added and automatically configure the MHC Tree with the JPEG decoder. To inspect or change this in the MHC Tree, see *Harmony Framework Configuration > Graphics Library > Harmony Graphics Library > Use Graphics*

Library? > Use Images? > Enable JPEG Support



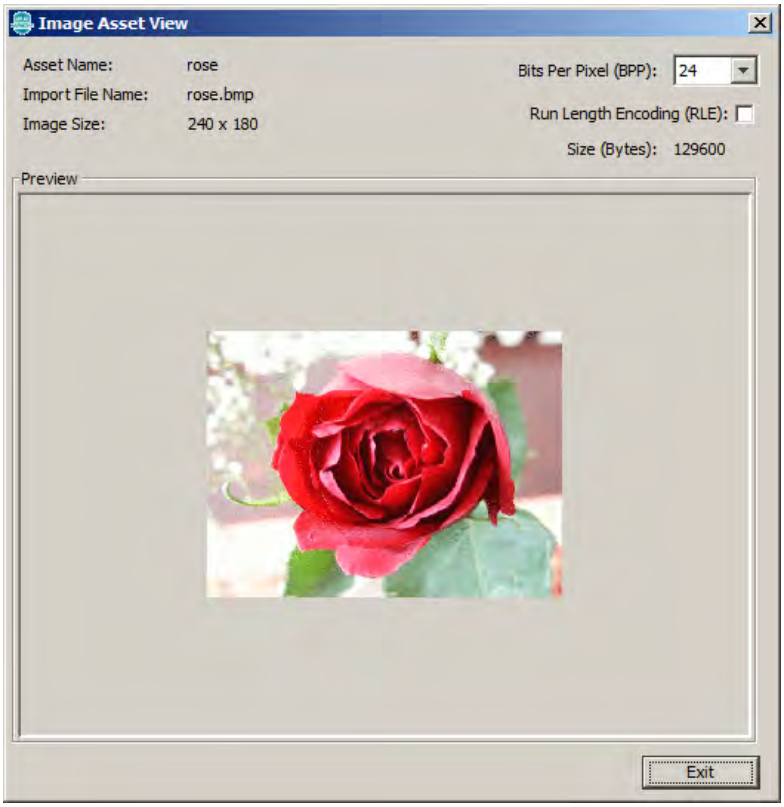
After selecting an image the dialog will display a preview of the image and asset size. The name field shows the asset name and must be unique. The Bits Per Pixel and Compression settings can be changed to see how the image and asset size change.

Note: These values are only for previewing. The current Graphics Library only supports a global BPP setting. The compression setting can be toggled in the asset management table.



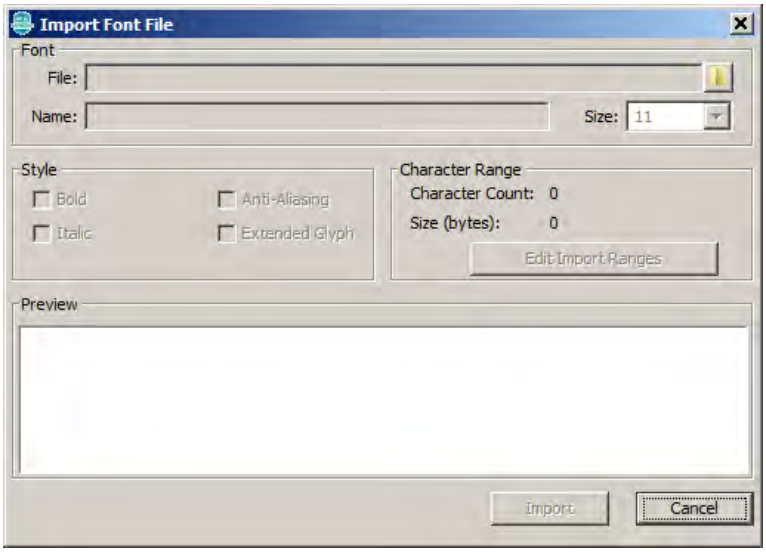
Upon selecting **Import**, the asset table will update to reflect the change. At this point, the asset can be renamed or compression can be enabled.

Selecting the asset and clicking **View** will show the image asset preview dialog.

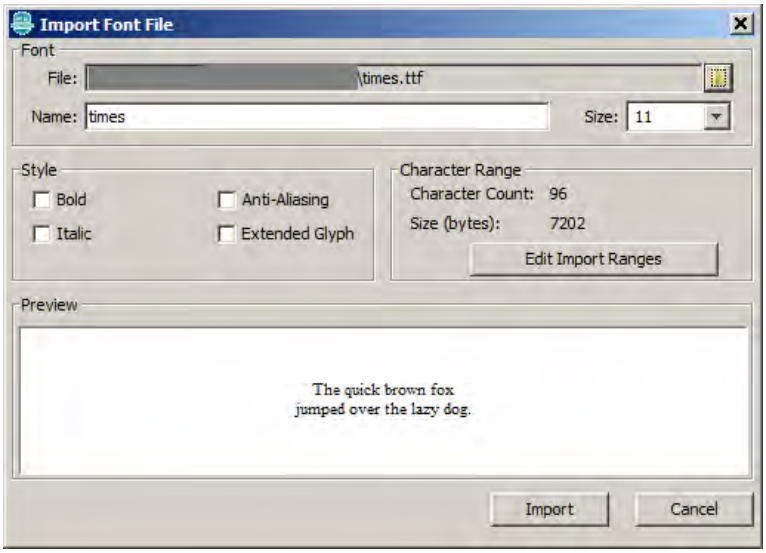


Importing Font Files

Clicking **Font File** opens the Import Font File dialog.



Use the File: Browse icon to locate a font file to import.



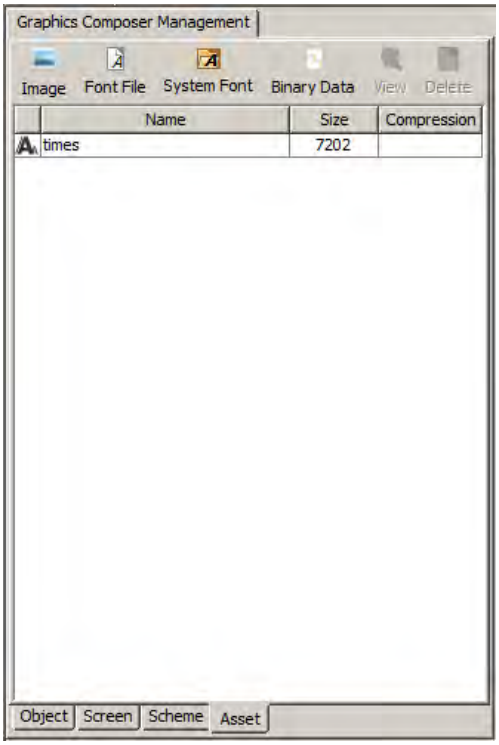
Upon selecting a font, the dialog will enable all of the options and display a preview of the font. Again, the asset name must be unique.

Font Option Descriptions

The following selections are available in Import Font File dialog:

- Bold – Renders the font as **Bold** type
- Italic – *Italicizes* the font
- Anti-aliasing – Enables anti-aliasing for this font in the Graphics Library
- Extended Glyph – Expands the range of imported characters

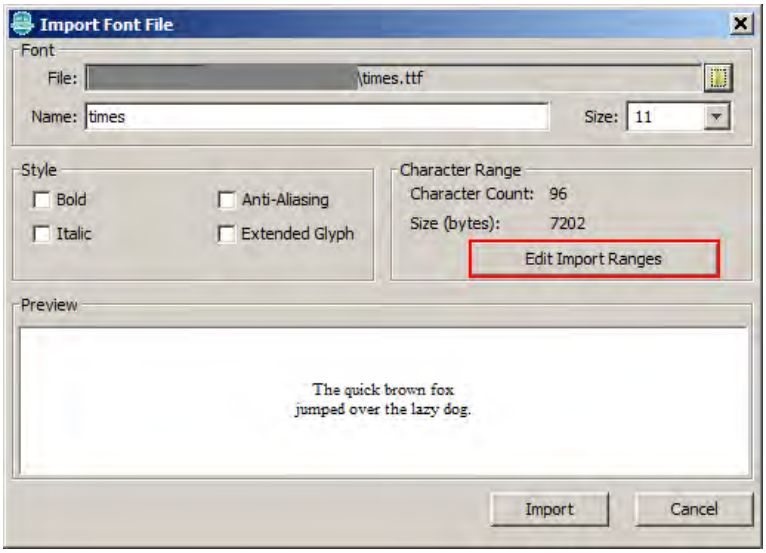
After finalizing your selections, click **Import**.

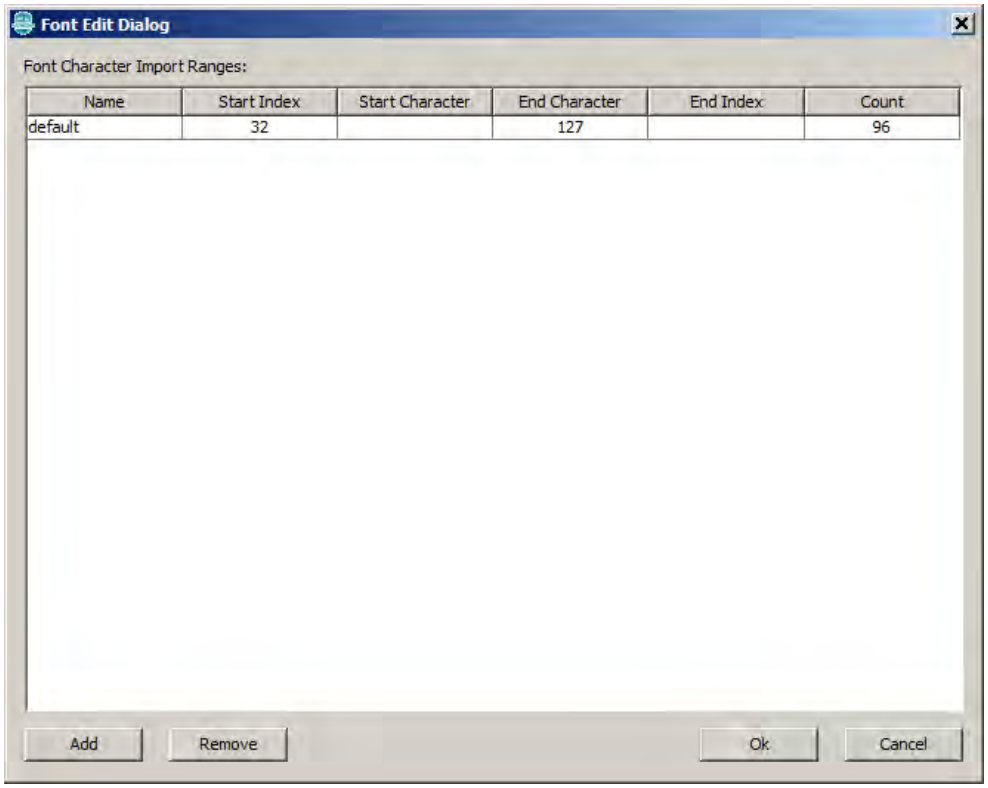


Font Range Selection

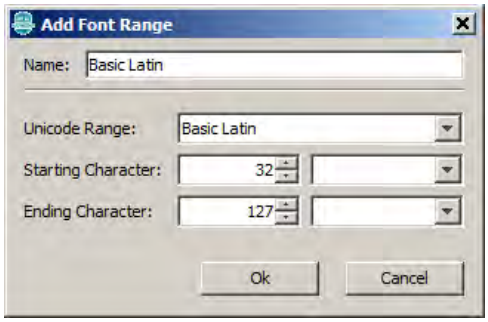
The font range dialog provides the method by which users can select multiple font glyph ranges from an imported font file. Only the selected glyph ranges will be converted into program data.

To open the font range configuration dialog click **Font Range**.





By default, the standard ASCII character range is added for every imported font. Users can either edit this range directly through the table or click **Add** to add a new range.



The Add Font Range dialog allows the user to add a glyph import range to the associated font file.

The process for adding a new range is:

1. Provide a glyph range name (if desired)
2. Select an overall Unicode glyph range

3. Choose a starting and ending glyph for this range.
4. Click **OK**.

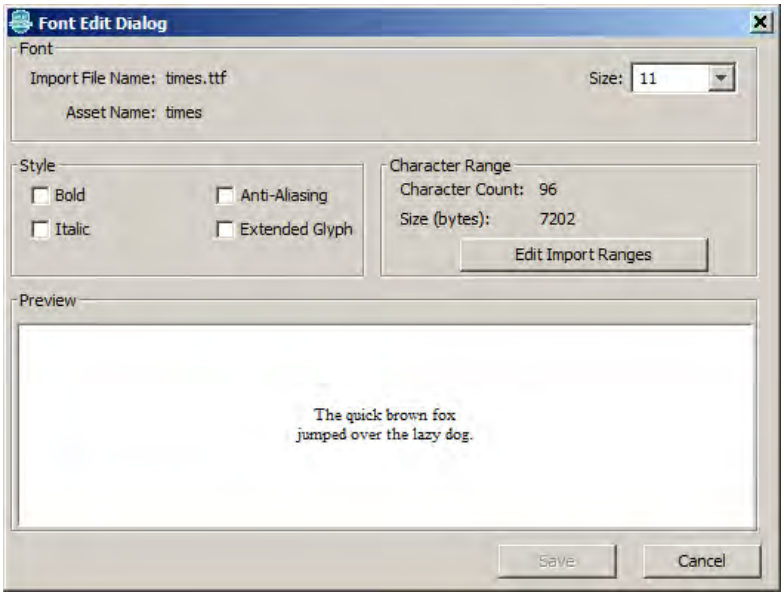
The new range will appear in the font range list.

16-bit Unicode Character Support

The GRC also supports 16-bit Unicode characters. To guarantee 16-bit Unicode support, be sure to set the Font Character Size to GFX_FONT_SIZE_16 in the MPLAB Harmony Configurator options (*Harmony Framework Configuration > Graphics Library > Harmony Graphics Library > Use Graphics Library? > Use Fonts? > Font Character Size*).

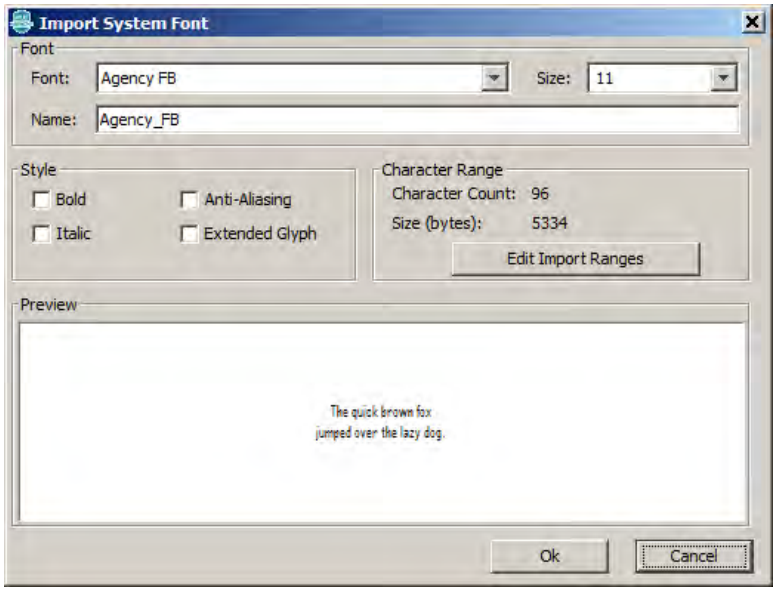
Editing a Font Asset

Font assets can be changed after import. Select the desired font to be changed and click **View** to open the Font Edit dialog.



Importing System Fonts

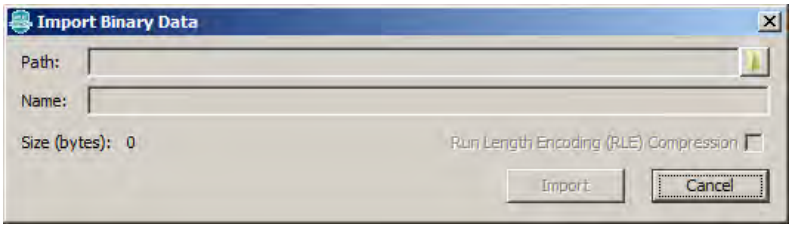
Click **System Font** to open the Import System Font dialog.



Importing system fonts works similarly to font files with the exception that instead of browsing for a physical file, the user selects from a list of installed fonts.

Importing Binary Data

To import binary data, click **Binary Data**, which opens the Import Binary Data Dialog.



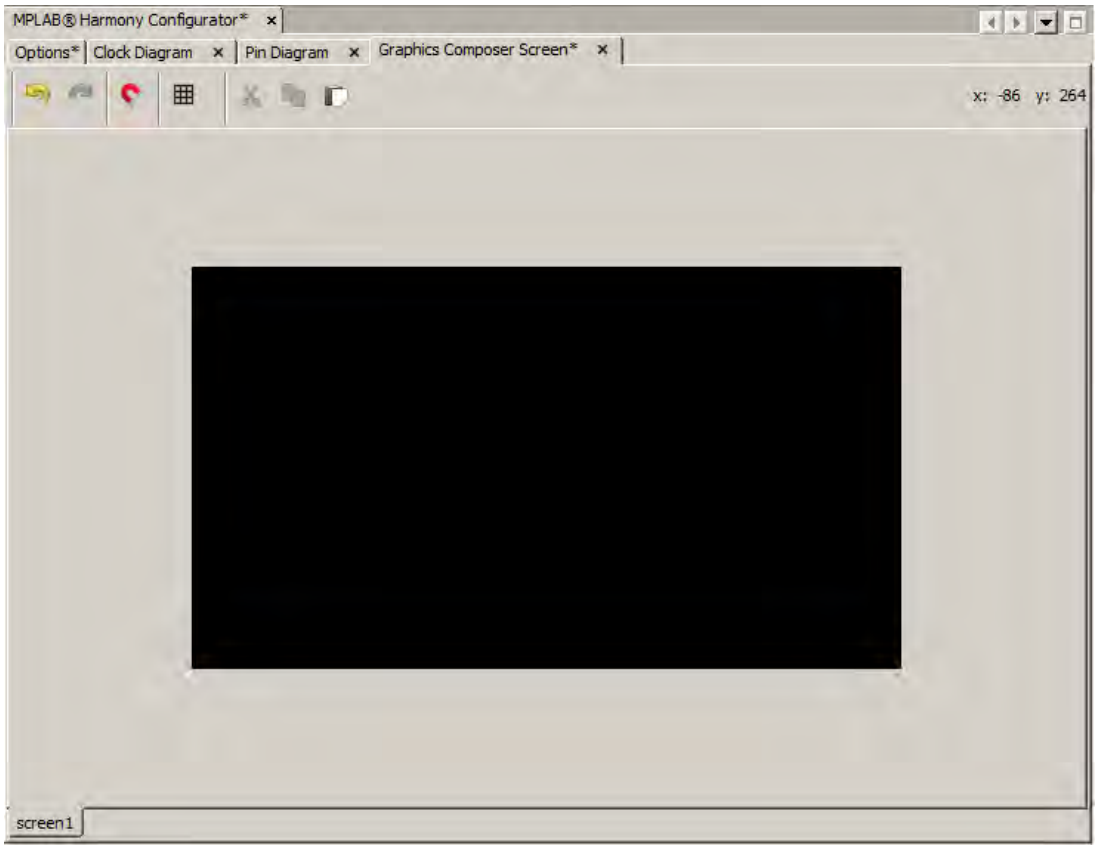
Select a file from using the Path: Browse icon and give it a unique name. The compression flags allows the user to preview if compression provides a size reduction benefit when storing this binary data.

Screen Window

Describes the features of the Screen Window.

Description

The screen window provides an approximate visual representation of the resultant embedded user interface.



Centered in the screen is an area that matches the size of the currently selected display device. This area will automatically resize when the display device changes in HConfig. The top-left corner of the box is at coordinates 0,0. The current cursor coordinates (in screen space) will be displayed in the top-right corner of the Screen Window when the cursor is inside the Screen Window.

The tabs at the bottom show the screens that are currently visible. These can be changed in the Screen tab of the Composer Management window.

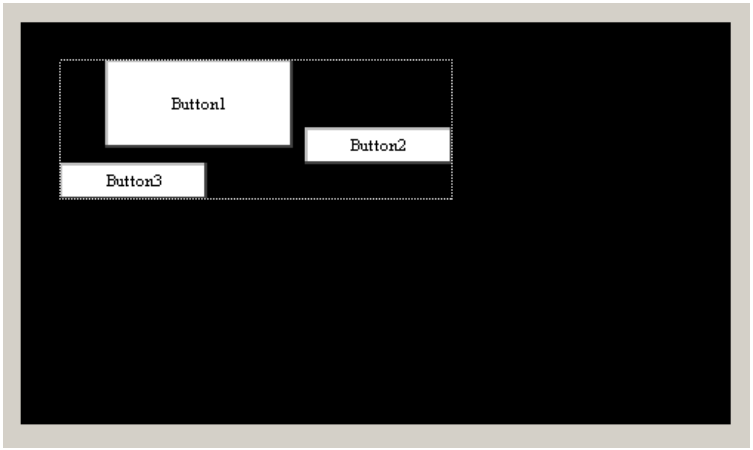
Buttons

The following selections are available:

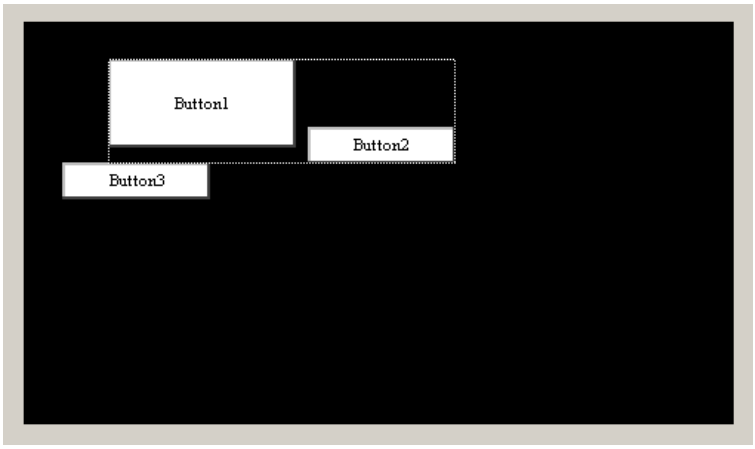
- Magnet – This icon enables line snapping while moving objects or points
- Grid – This icon enables a visual grid that can be snapped to. When selected, the user can adjust the grid size and color.
- Left Arrow – This icon performs an "undo" of the last action
- Right Arrow – This icon performs a "redo" of the last action
- Scissors – This icon performs a "cut" of the currently selected objects
- Pages – This icon copies the currently selected objects
- Clipboard – This icon pastes the currently selected objects to the clipboard

Manipulating Objects

The Screen window provides the ability to graphically configure the objects of a screen. Given a screen with the following layout, left click an object to select it.



The manipulator can be moved by left-click-dragging it. The white circles represent the handles that allow the manipulator to be resized

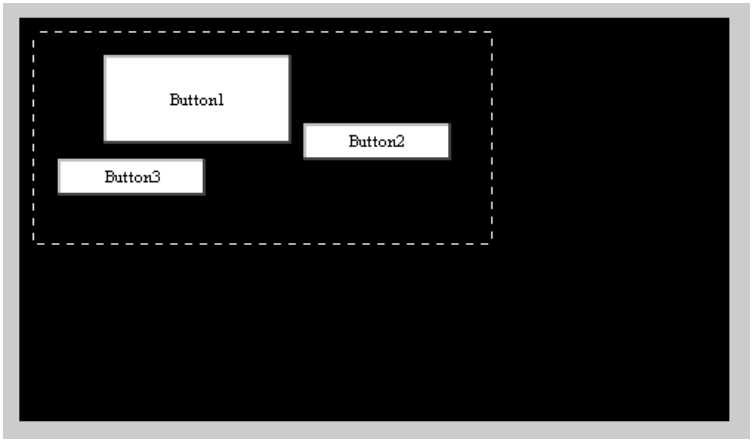


Selecting Multiple Objects

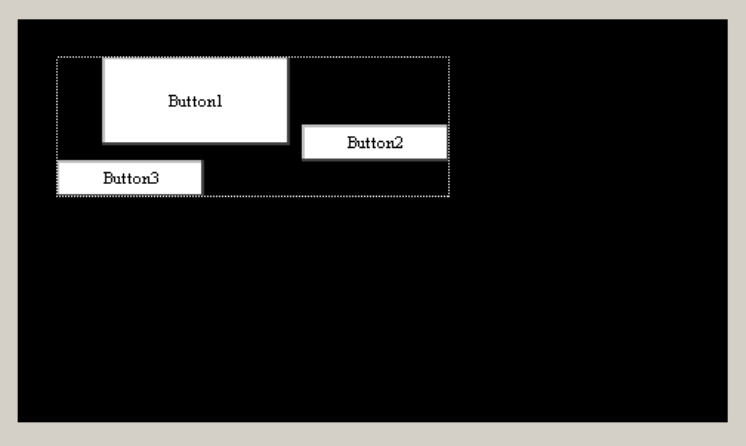
Several methods are available to select objects. Refer to the Object Tab topic for details.

Marquee Select

Left click in the screen and drag the marquee (i.e., dotted) box around the desired objects.

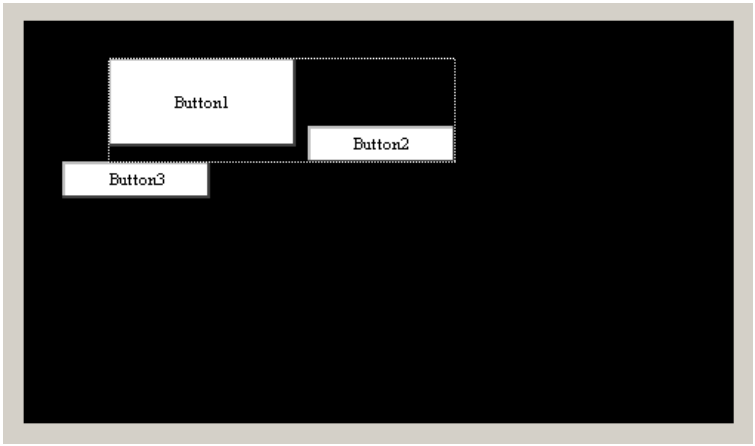


The Object Group Move box will appear. Drag the box to move the selected objects.

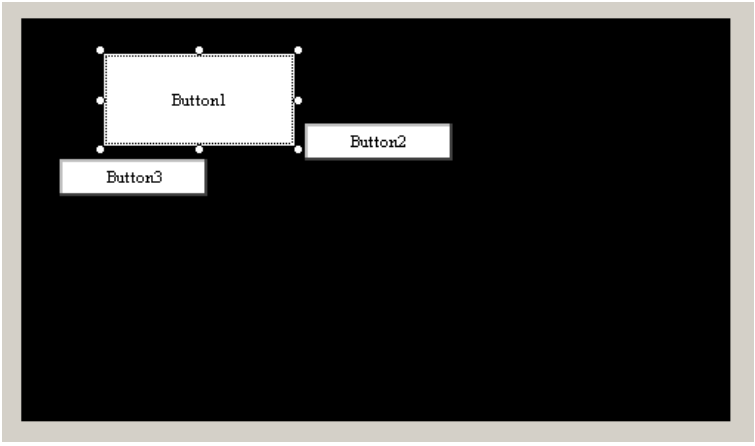


Managing Object Selection

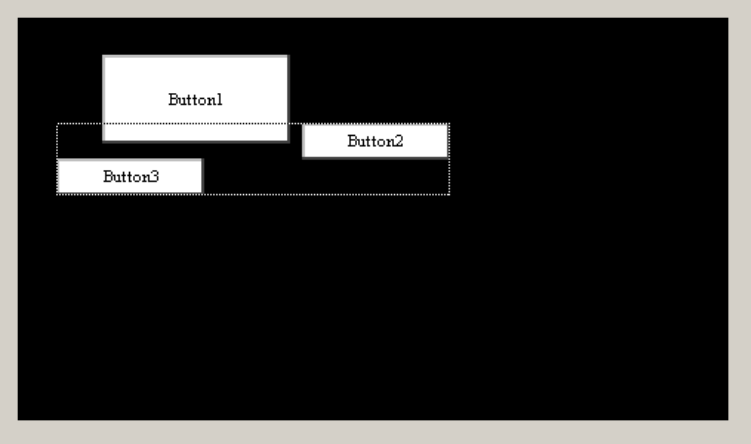
Objects can be added or removed from a group selection. To remove Button3 from the previous selection, press and hold the <Ctrl> key and left click inside Button3. Button3 will be unselected. The object can be added back to the group by pressing and holding the <Shift> key and left clicking the Button3 object. A toggling selection can be performed by pressing and holding <Ctrl>+<Shift> and selecting either with a left click or a marquee box selection.



Before:

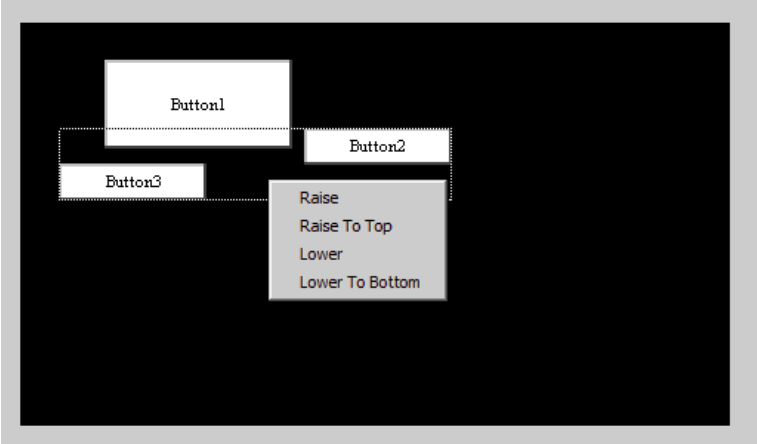


After:



Order Management Through the Screen Window

Object ordering can be managed from the Screen Window as well as the Object tab in the Composer Management Window. Right click an object or a group of objects to display the context menu.

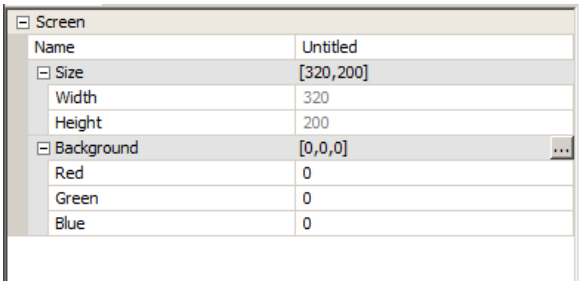


Properties Window

Describes the features of the Properties Window.

Description

The Properties Window displays options for the currently selected object, or the options for the active screen if no objects are selected.

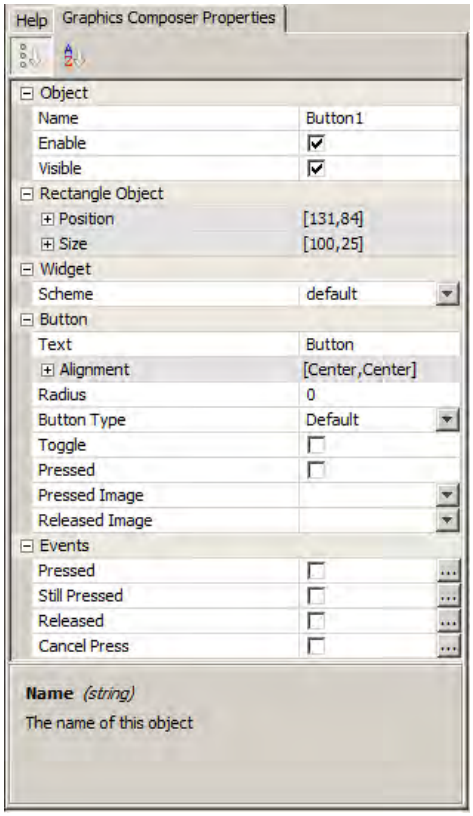


To edit an option, left click the value in the right column and change the value. Some values have an ellipsis that will provide additional options. In the previous case, the ellipsis button will display the Color Picker dialog.

Some properties, like the screen width and height, are locked and cannot be edited. Other properties offer check boxes and combo-type drop-down box choices.

Some properties are grouped together like the Position and Size entries. Individual values of the group can be edited by expanding the group using the plus symbol.

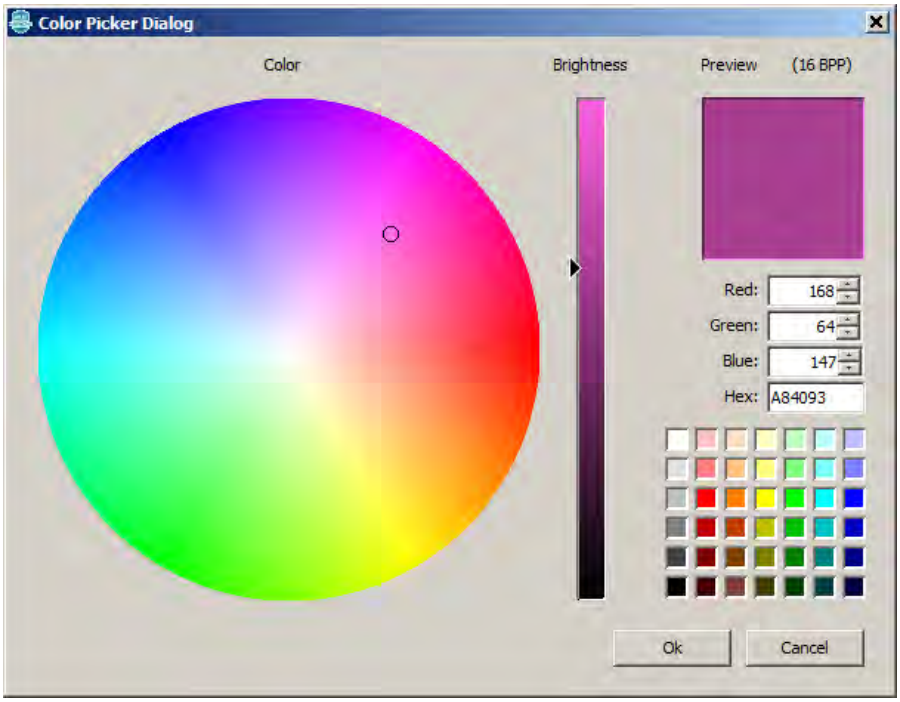
For example, the following figure shows properties for a Button Widget.



Notice that the bottom panel provides help text for each property, which provides the type of data expected and a description of what the property represents. Some properties are configured to reject invalid settings.

Color Picker Dialog

The Color Picker dialog allows the user to easily select a color by providing a color wheel, brightness gauge, and some common predefined color choices. The user can change the individual color values or input a number in Hexadecimal format. The end result is displayed in the top right corner.



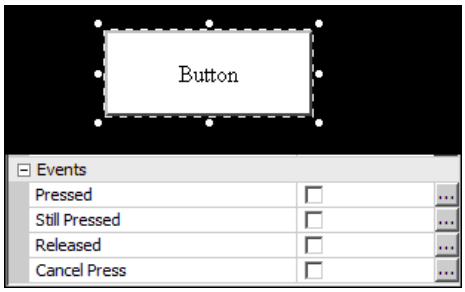
Event Generation

This topic describes using the graphics composer to generate events.

Description

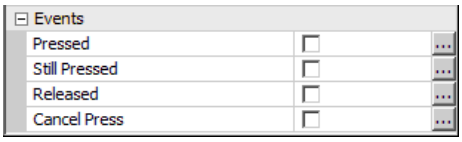
Some objects have events that can be enabled and defined. The graphics composer provides the capability to generate event handler code using a visual interface.

As shown in the following figure, these events are associated with the GOL Button Widget.



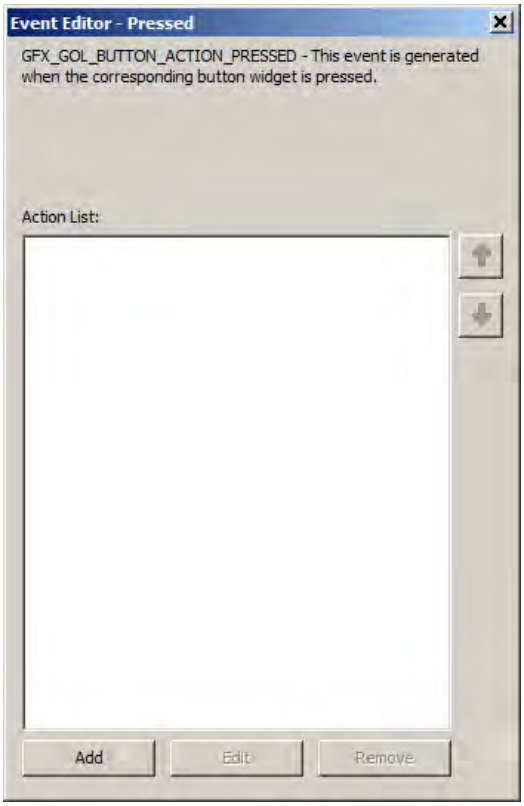
Defining Events

To define events, select the object on the screen for which events are to be defined. If an object that supports events is selected, the properties table will display the events that can be defined for that object. Select the check box to enable the event. If the generate process is run at this point, an empty event handler will be created for that event.

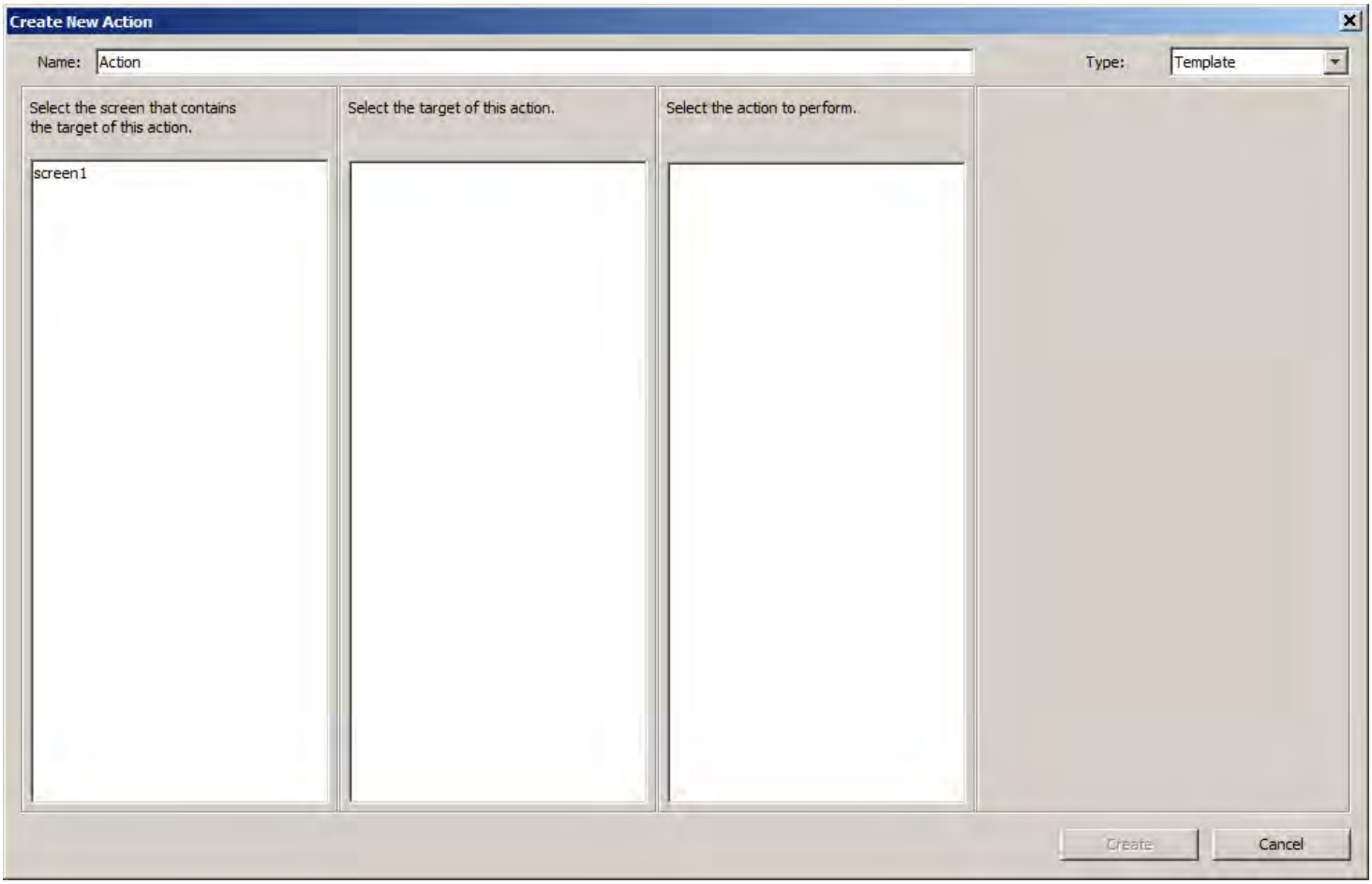


Defining Event Actions

Each event property has a corresponding ellipsis button. Clicking this button opens the Event Editor dialog.



In this case, the Event Editor displays the event state for the "Pressed" event of a GOL Button Widget. To add an action, click **Add**. This action will open the Create Action dialog. Actions can be edited after creation and can be removed using this dialog. Action code is generated in top-down order. The arrows on the right change can the order of the action list to configure action precedence.



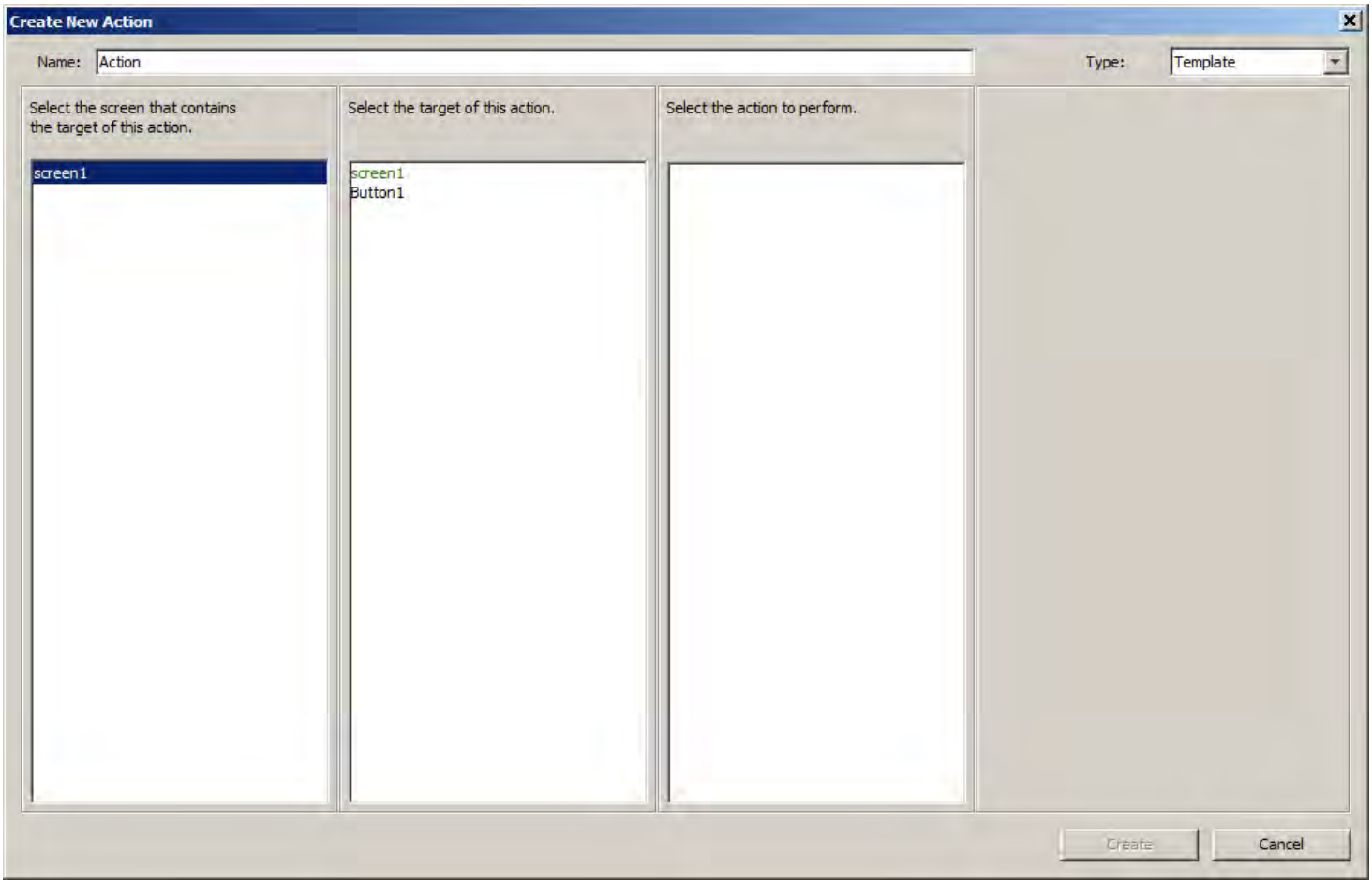
There are two types of actions that can be defined for an event: Template and Custom. Template events allow the user to choose a source screen, an action target, an action, and potentially data associated with an action.

Creating a Template Action

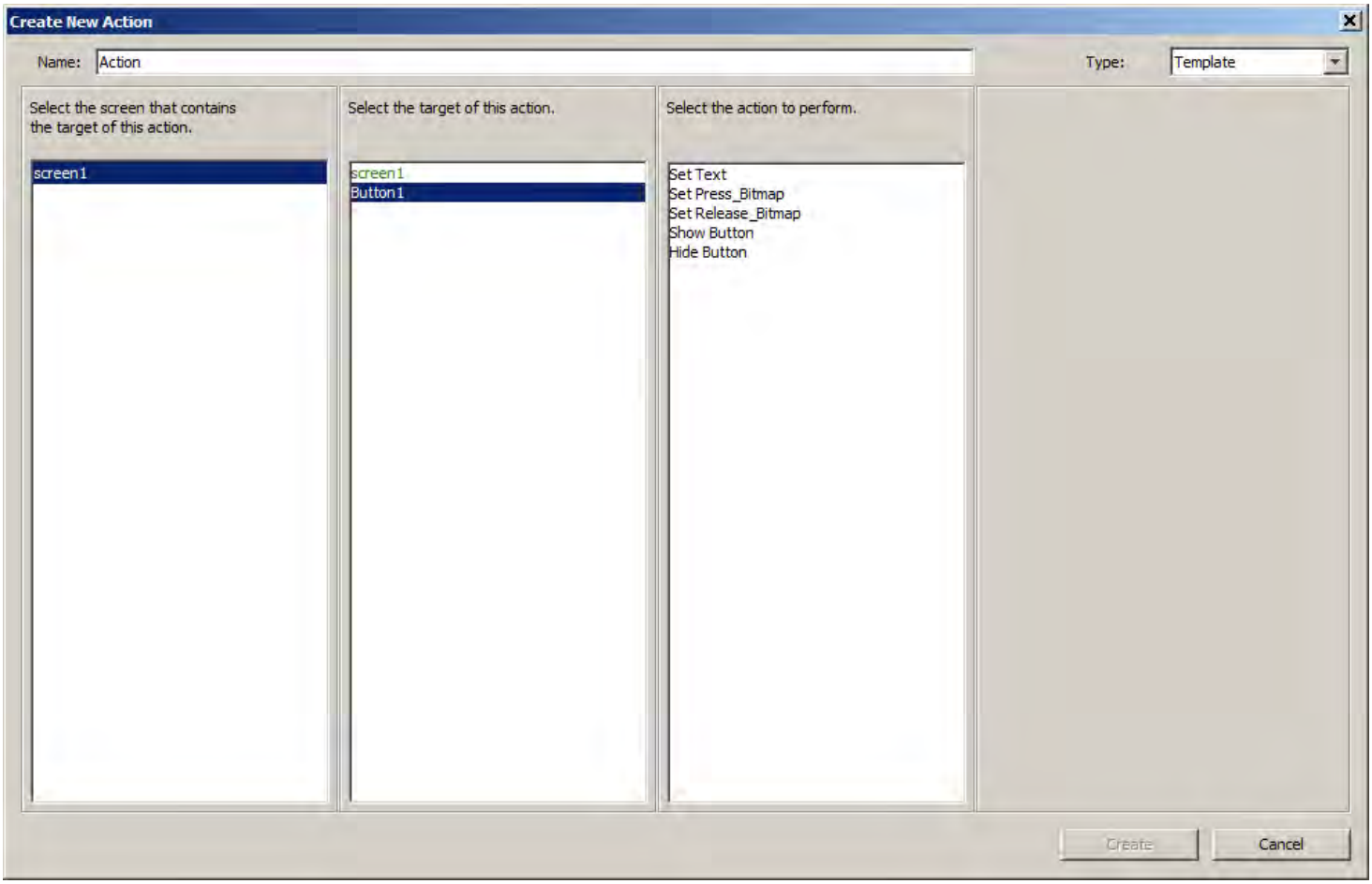
To create a template action, follow these steps:

1. From the first column select the screen that is, or contains, the target of this action.
2. From the second column select the target of this action. Screens are highlighted in green.
3. From the third column select the action to perform on the selected target.
4. From the fourth column input the requested data for the action (not always required).

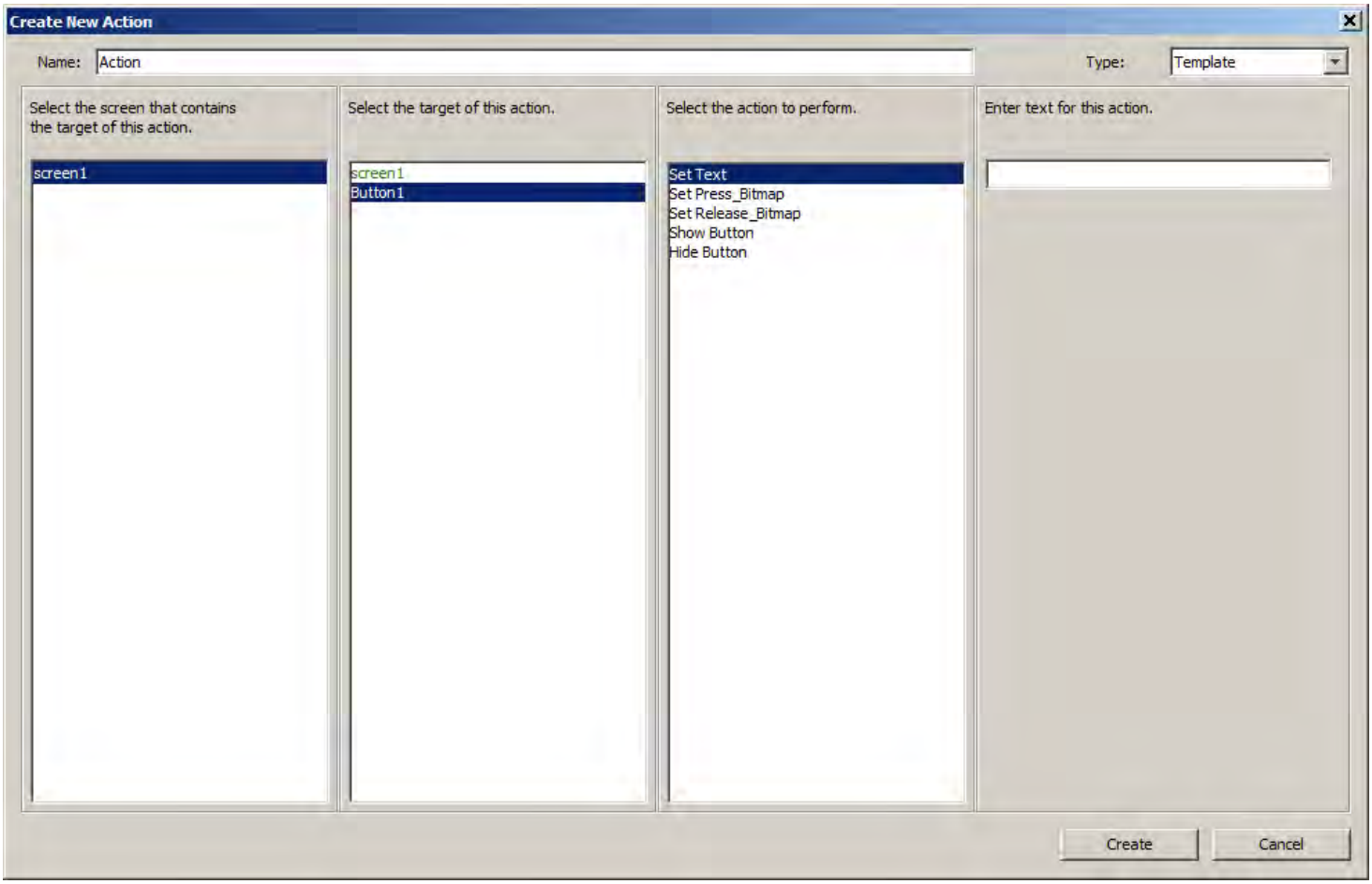
In this example "screen1" will be selected as the source.



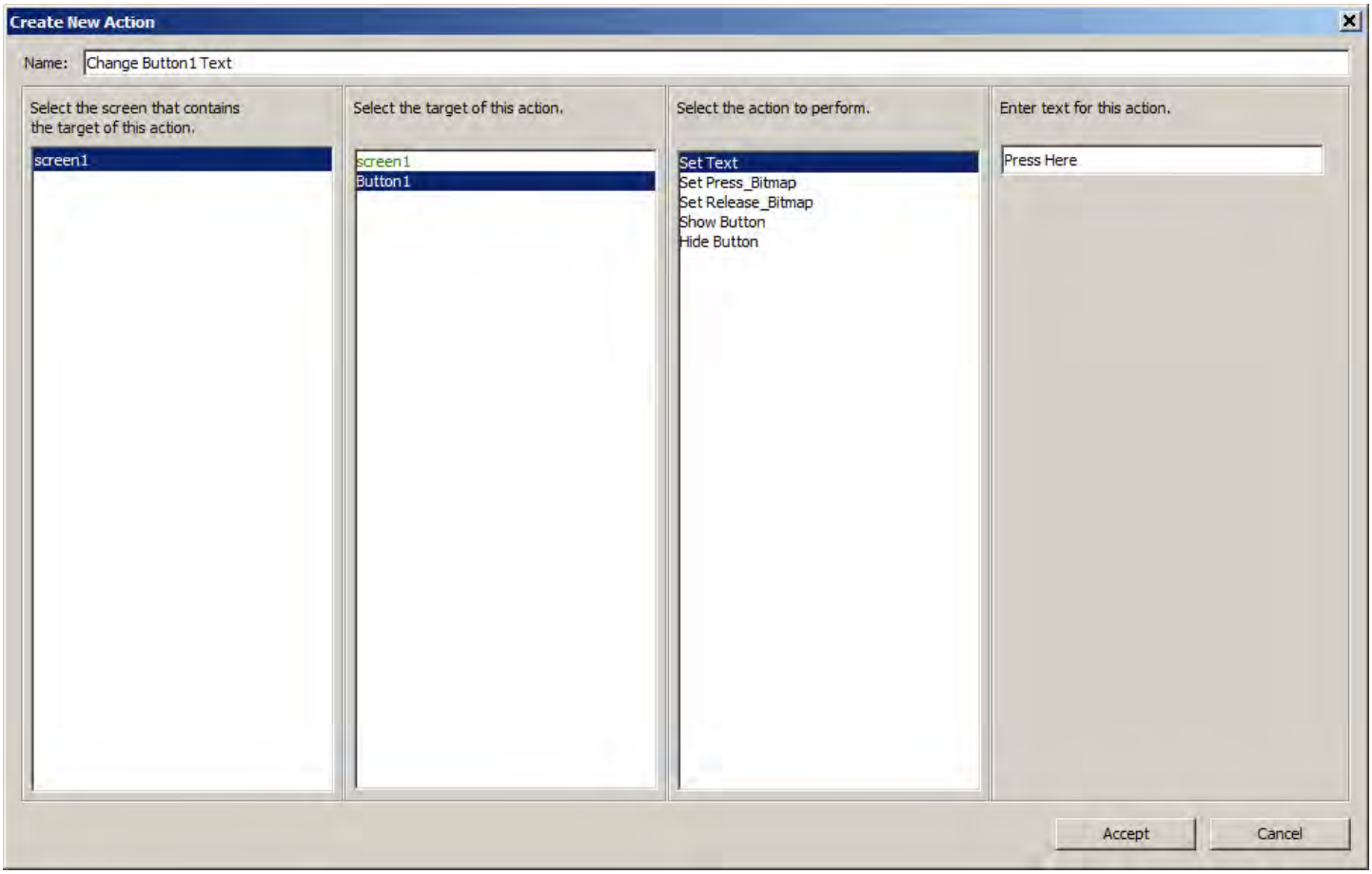
The object named "Button1" will be selected as the target.



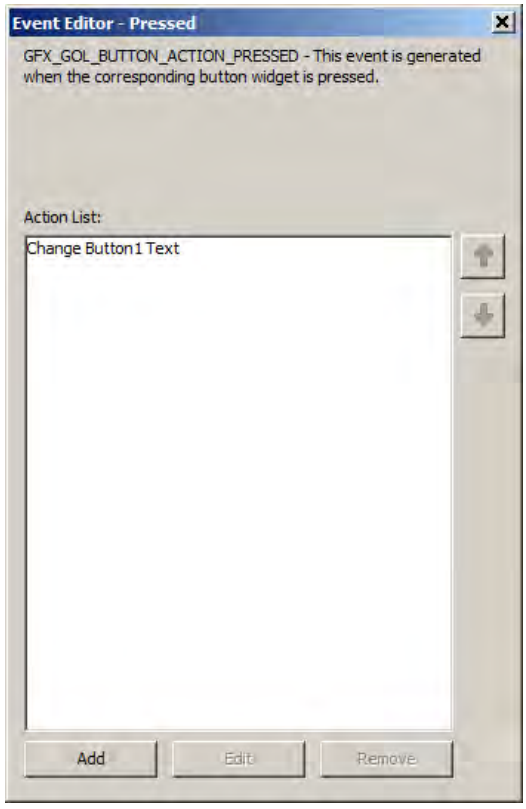
The action that will be selected is "Set Text".



This action requires that the user input the text to assign to "Button 1". The user can also assign a unique name to an action.



Once these steps are complete, the dialog will enable the "Create" button and the action can be finalized. When **Create** is clicked, the dialog will close and the action will be added to the action list for the event.

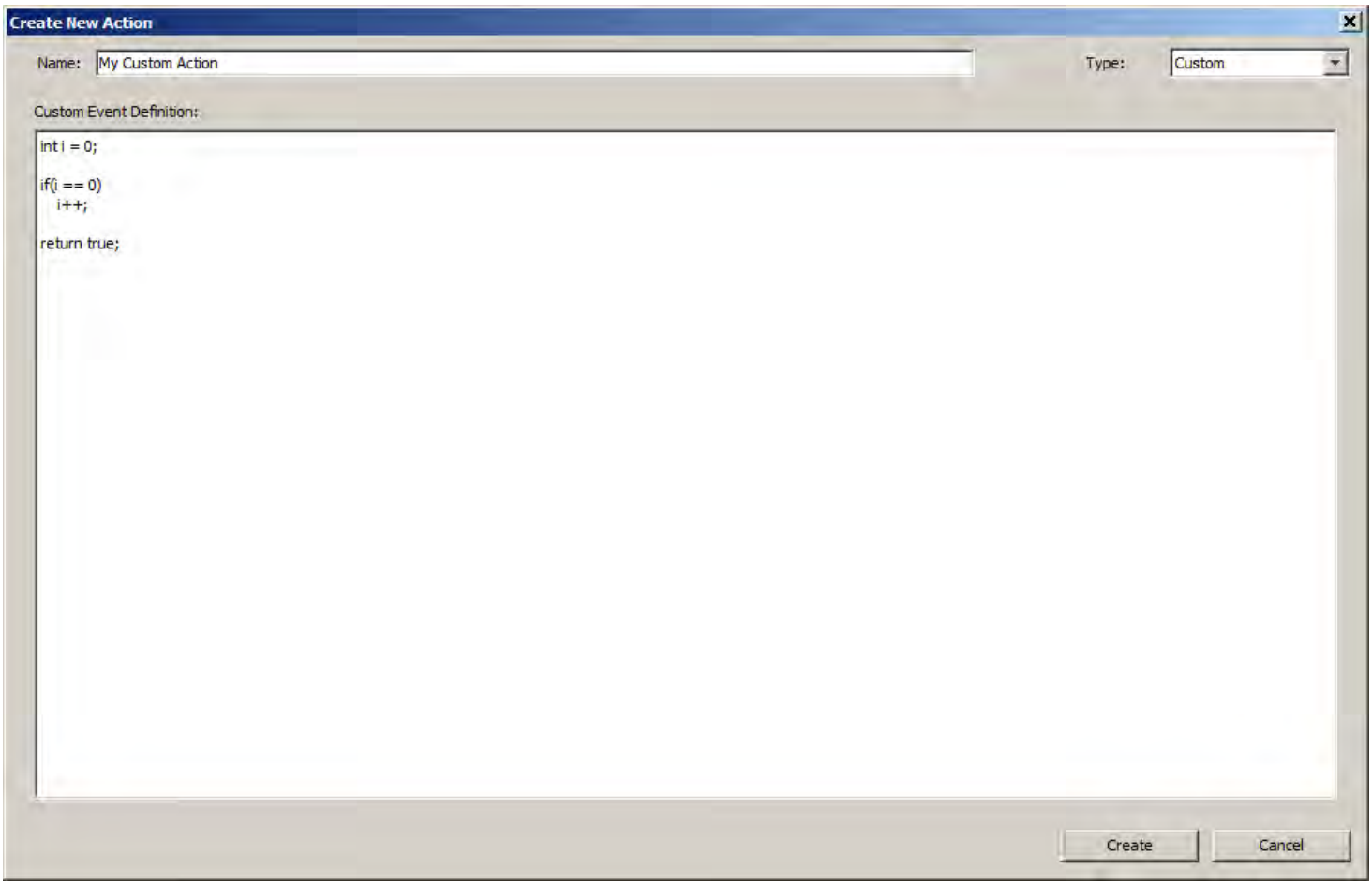


Creating a Custom Action

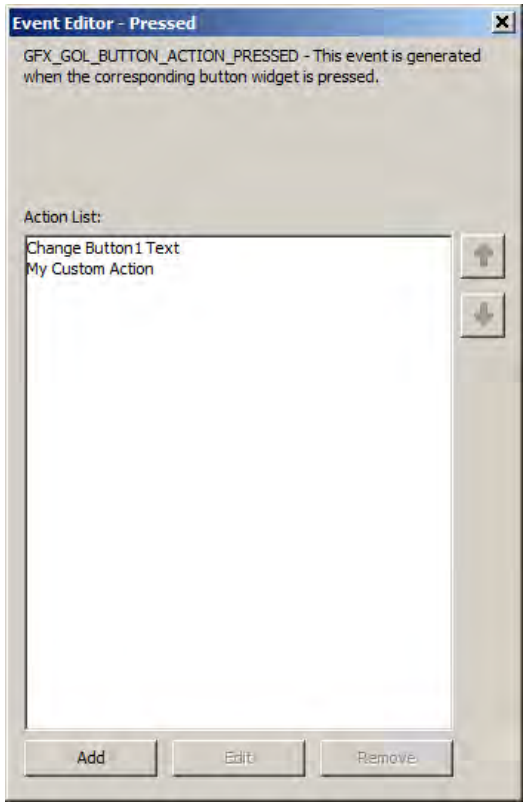
The second type of action is a Custom action. This type allows the user to include custom code and have it inserted into the event handler

function. The graphics composer is not responsible for ensuring that the code input is valid.

To create a custom action click **Create** in the Event Editor dialog and change the type selection in the top right corner to Custom.



Clicking **Create** adds the custom action to the event action list.



The code generator will automatically generate or include the defined actions during generation.

```
Remarks:
    Handles GFX GOL Button events
*/
bool GFX_HGC_MsgButtons(uint16_t objMsg, GFX_GOL_OBJ_HEADER *pObj)
{
    switch (GFX_GOL_ObjectIDGet(pObj))
    {
        case Button1:
            if (objMsg == GFX_GOL_BUTTON_ACTION_PRESSED)
            {
                // Button Pressed Event Code
                // Change Button1 Text
                GFX_GOL_ButtonTextSet(((GFX_GOL_BUTTON*)(GFX_GOL_ObjectFind(GFX_INDEX_0, Button1))), (GFX_XCHAR*)"Press Here");
                GFX_GOL_ObjectStateSet(((GFX_GOL_BUTTON*)(GFX_GOL_ObjectFind(GFX_INDEX_0, Button1))), GFX_GOL_BUTTON_DRAW_STATE);

                // My Custom Action
                int i = 0;

                if(i == 0)
                    i++;

                return true;
            }

            return true;
        default:
            return false; // process by default
    }

    return true;
}
```

Auto-configuration

MPLAB Harmony Composer will automatically enable touch input support in the MHC tree when a event is created and enabled. This setting can be manually overridden in the MHC tree. Refer to *Graphics Library > Harmony Graphics Library > Use Graphics Library > Use Input Devices? > Enable Touchscreen Support* in MHC for details.

Code Generation

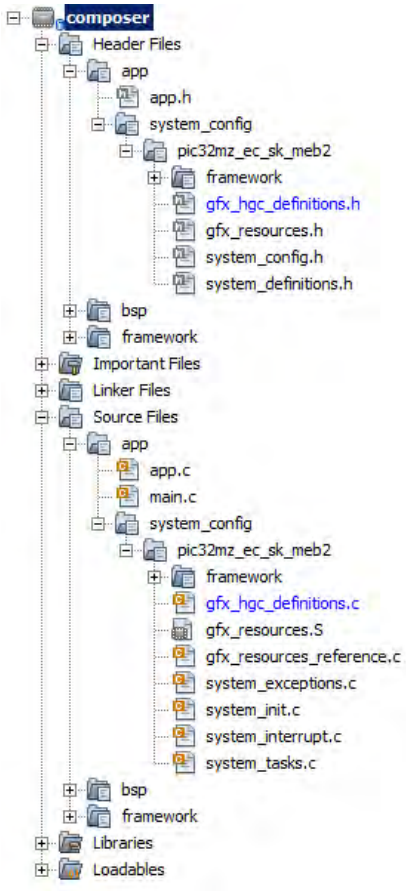
This topic describes using the graphics composer to generate code.

Description

MPLAB Harmony Graphics Composer data is generated the same way as the rest of the project within MHC through the Generate button.

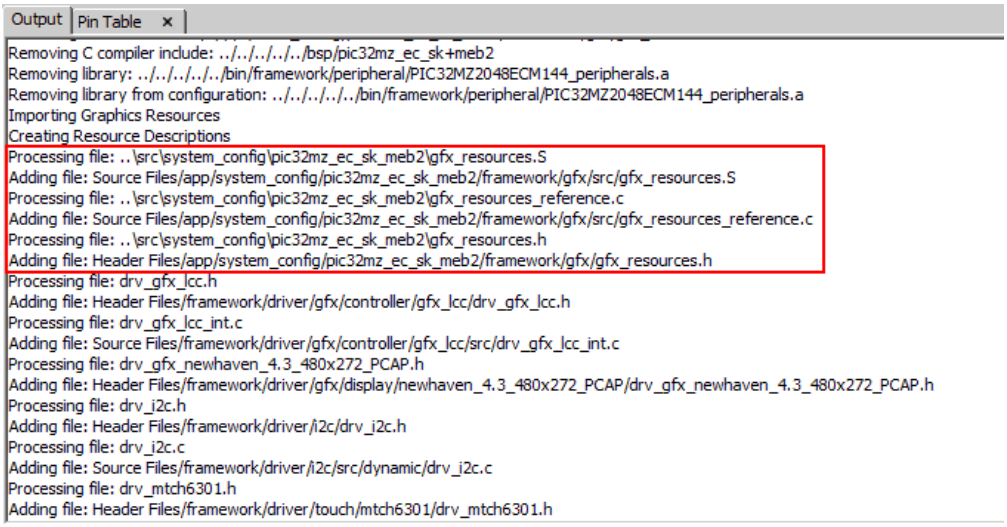
For function calls to the gfx_gol and gol_primitives library, the graphics composer will add the `gfx_hgc_definitions.h` and `gfx_hgc_definitions.c` files to `app/system_config/<configuration_name>/framework/gfx` folder in the header and source file structure of the MPLAB X IDE Project.

For generated asset data, the graphics composer will add the `gfx_resources.h`, `gfx_resources.S`, and `gfx_resources_reference.c` files to the `app/system_config/<configuration_name>/framework/gfx` folder in the header and source file structure of the MPLAB X IDE Project.

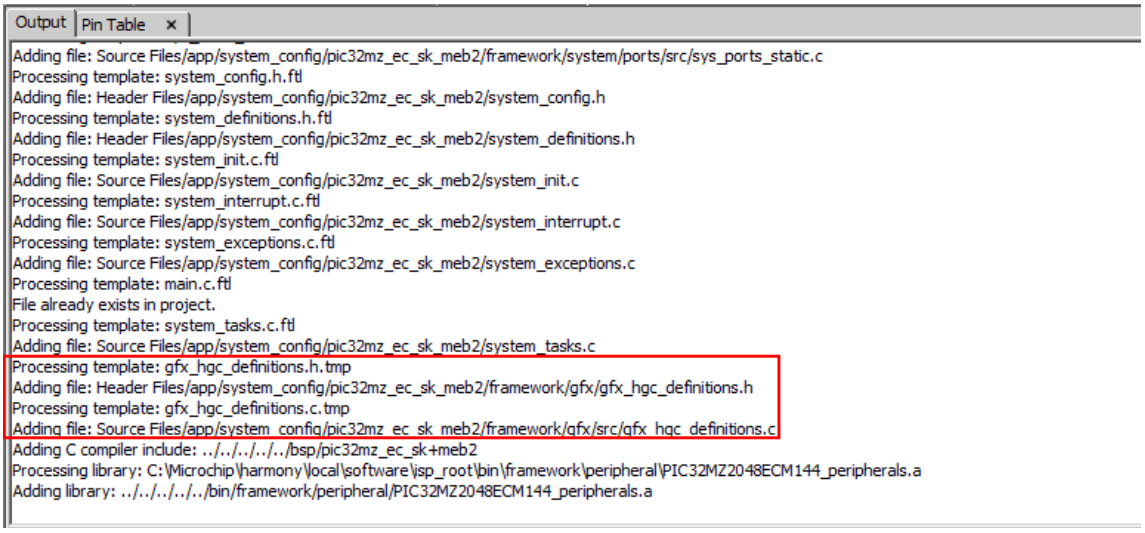


You may want to monitor the progress of the generation in the Output window.

The asset resource is the first to get generated. To confirm accurate file generation, look for output such as that highlighted in the following figure.



For accurate generation of the draw function calls, you should expect output such as that highlighted in the following figure.

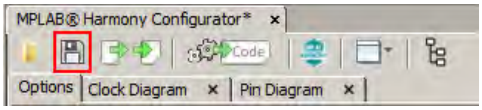


Saving and Loading Data

This topic describes using the graphics composer to save and load data.

Description

The graphics composer saves and loads its data into the `configuration.xml` file of the MHC configuration. This file is always located within `<install-dir>\apps\<feature>\<demonstration_name>\firmware\src\system_config\$CONFIGURATION_NAME`. The saved data is loaded when MHC starts and is saved when the configuration is saved through the Save or Save As dialogs.



Importing and Exporting Data

This topic provides information on importing and exporting graphics composer-related data.

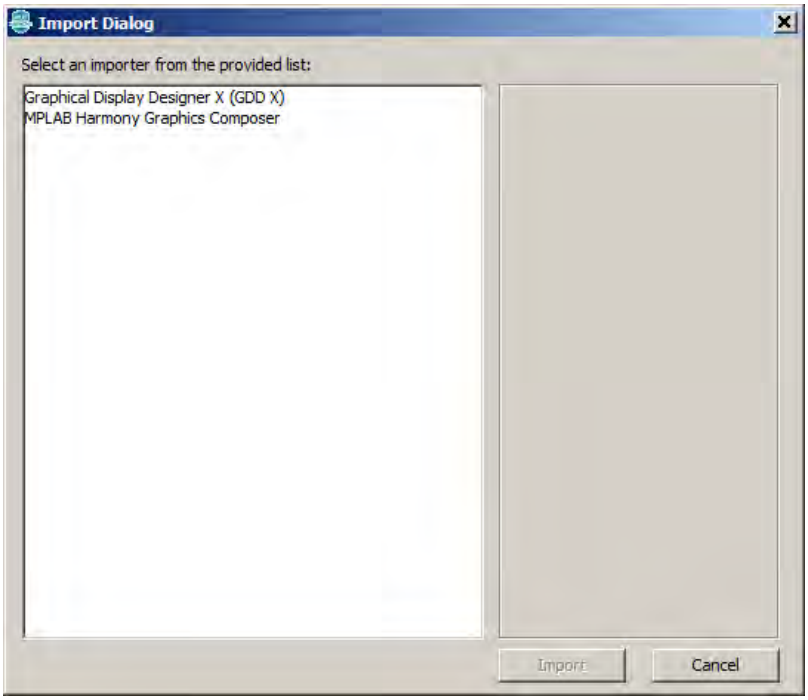
Description

The MPLAB Harmony Graphics Composer provides the capability for users to import and export graphics composer-related data. The user can export the state of an existing graphics composer configuration, import another graphics composer configuration, and import projects from the Graphics Display Designer X (GDD X) utility.

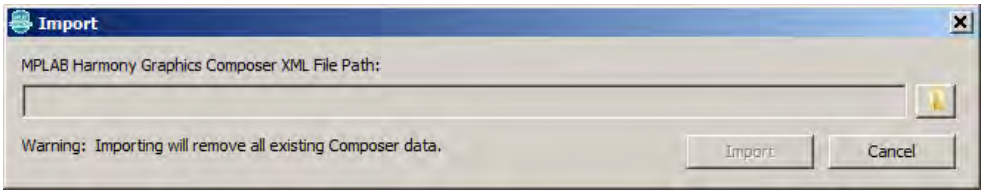
The import and export interfaces are located in the Configuration dialog of the MPLAB Harmony Configurator, which is accessible from the Options tab.

Importing Data

To import data into graphics composer, click **Import** from the main window toolbar. The Import dialog will appear.



The user can choose to import either GDD X or graphics composer data. Upon selecting a format and clicking **Import**, a path dialog will appear and the user can browse to either a graphics composer XML file or a GDD X project file.

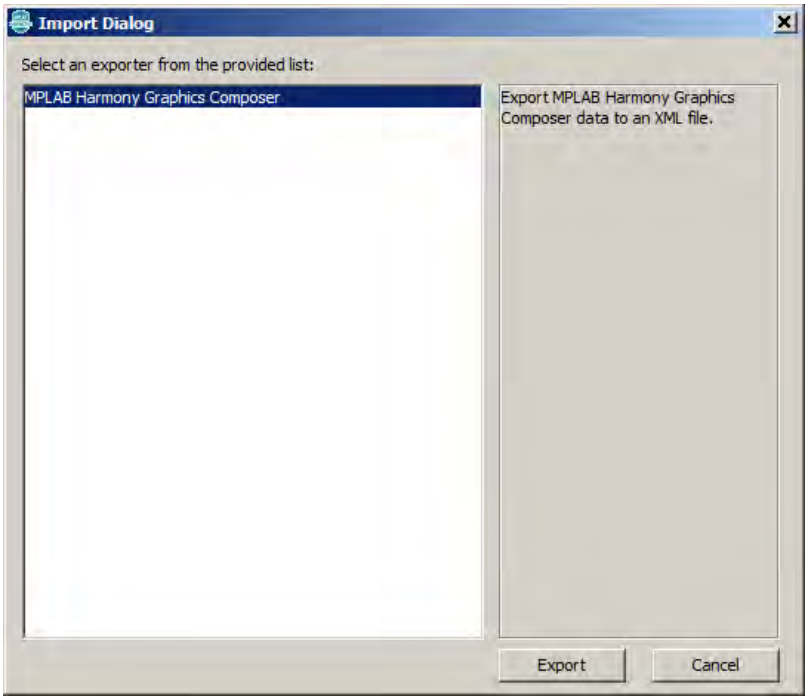


Warning

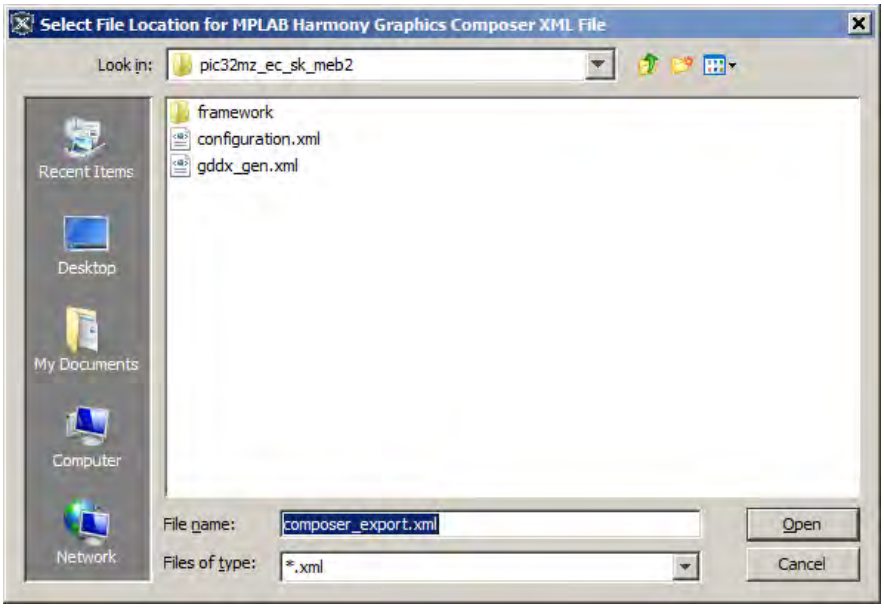
Importing data will remove all currently existing graphics composer data.

Exporting Data

To export a Composer configuration click **Export**.



Select MPLAB Harmony Graphics Composer from the list and click **Export**.



Select the file path where the exported data should be placed and click **Open**. The current graphics composer data will be written into this file.

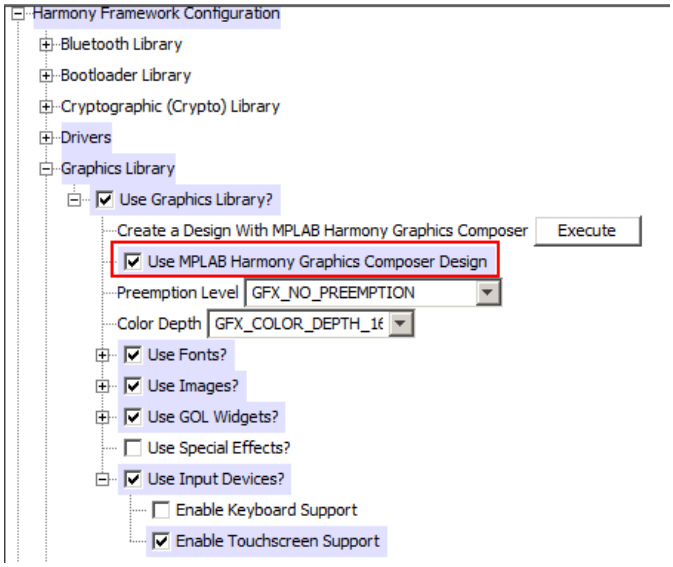
Managing Graphics Composer Features

This topic describes how to manage graphics composer features.

Description

Users can easily enable or disable all graphics composer features using the option **Use MPLAB Harmony Graphics Composer Design**.

If this configuration flag is enabled, the graphics composer will generate its respective state machine code and will also take responsibility for managing many of the Graphics Library options.



MPLAB Harmony Configurator User's Guide

This section provides user information on using the MHC.

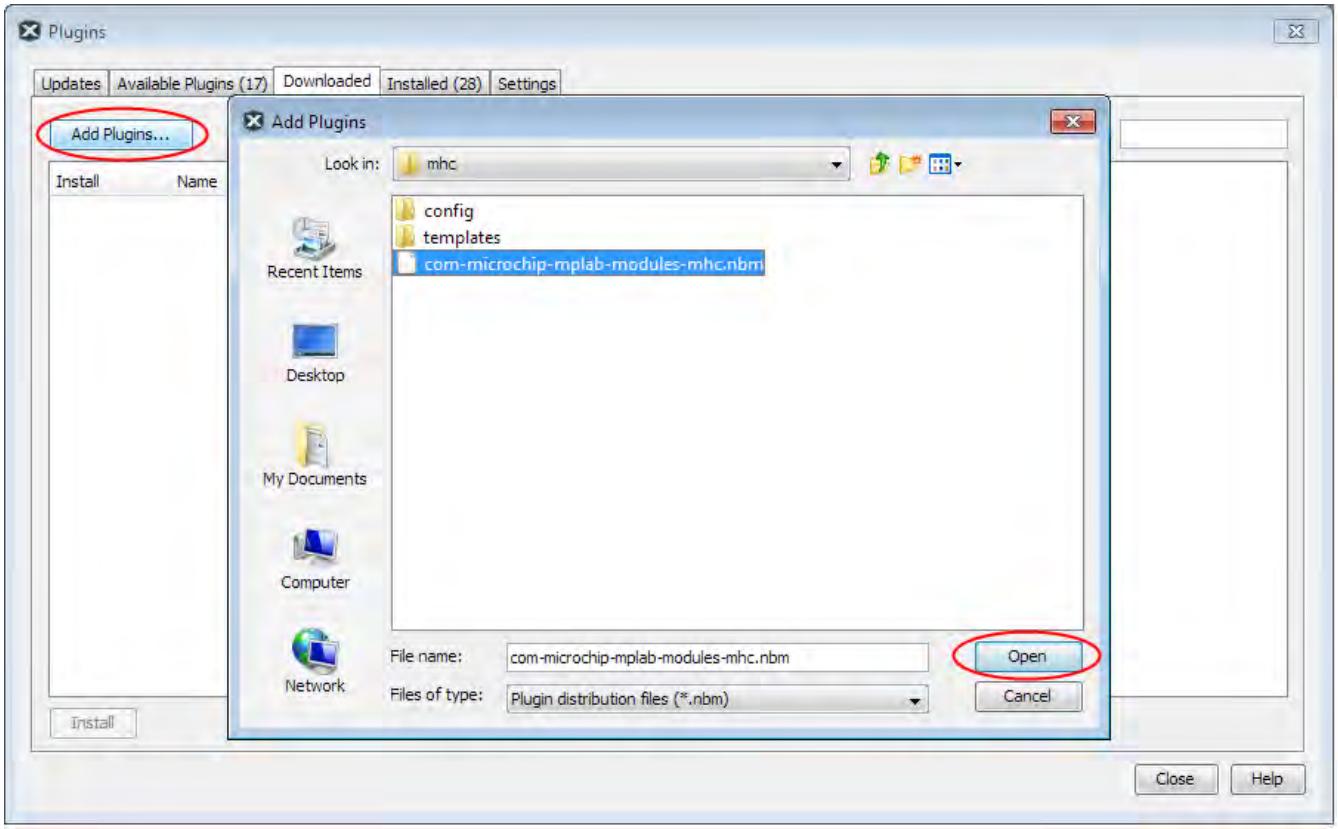
Installing MHC

This topic provides information on installing the MHC plug-in.

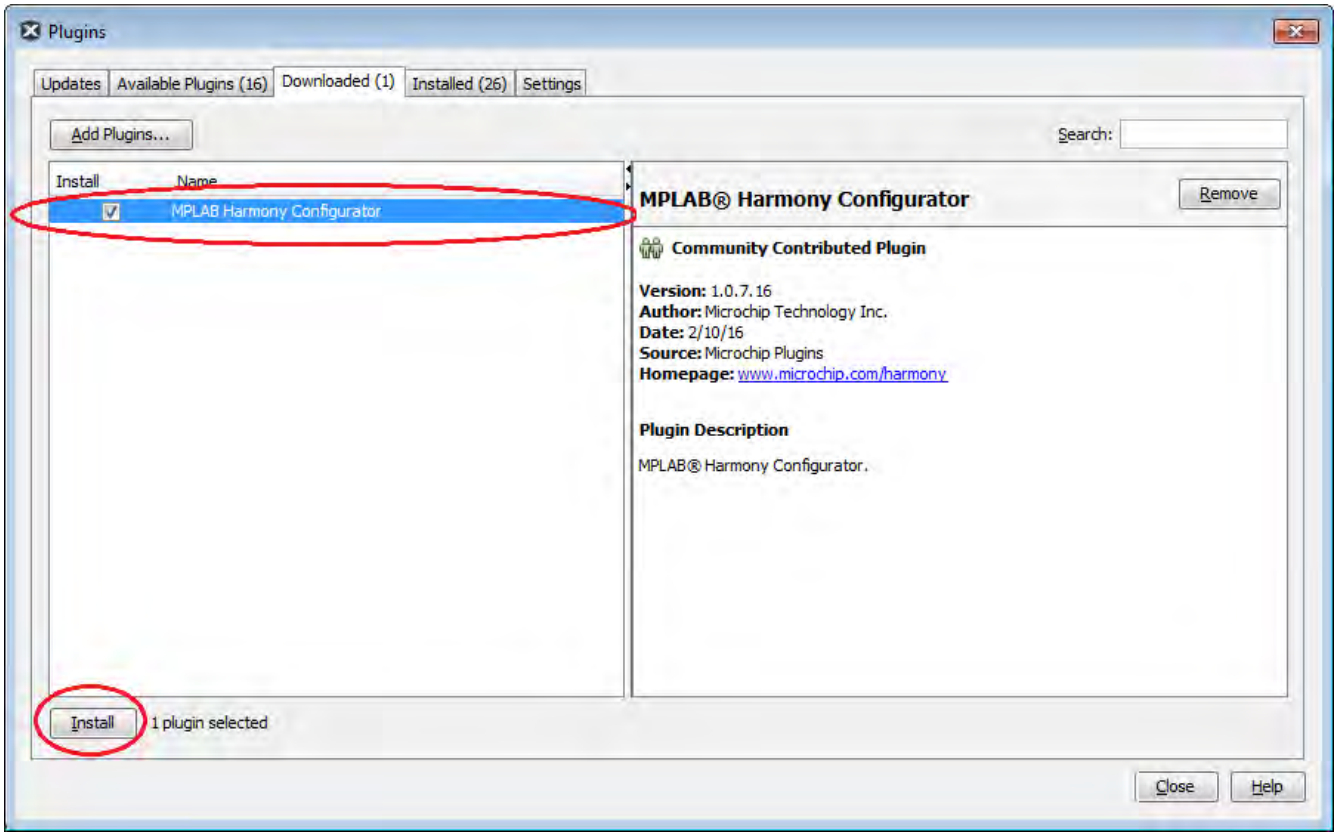
Description

Installing the MHC Plug-in

1. Start MPLAB X IDE and select *Tools > Plugins*.
2. Select the **Downloaded** tab and click **Add Plugins...**.
3. In the Add Plugins dialog, navigate to the MHC `com-microchip-mplab-modules-mhc.nbm` plug-in file, which is located in `<install-dir>/utilities/mhc`, and then click **Open**.



4. Ensure that the Install check box for the plug-in is selected and click **Install**.



5. Follow the prompts from the installation and continue until the installation completes. (Do not be concerned if the version you are installing is *signed* but not *trusted*, simply click **Continue**). Once the installation has finished you can close the **Plugins** dialog.
6. To verify the installation, select *Tools > Plugins* and select the **Installed** tab. The MHC plug-in you installed should be included in the list.

MPLAB Harmony Configurator Interface

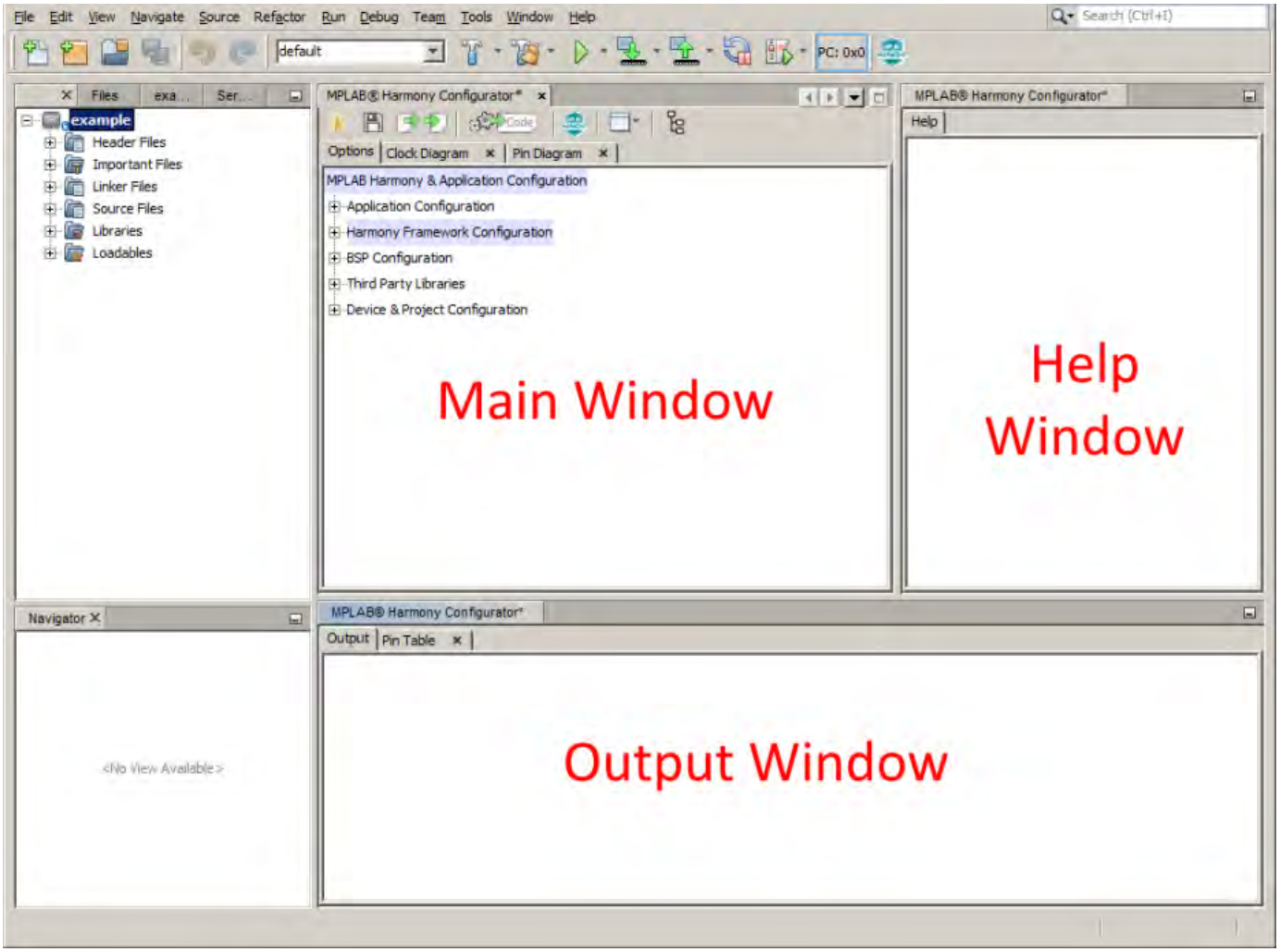
This section describes the MHC interface.

Description

This section provides a basic overview of the MHC user interface. For detailed information on using MHC to create a MPLAB Harmony application, refer to Using MHC to Create a New Application.

Initial Interface Configuration

The following figure shows the initial interface configuration for MHC.

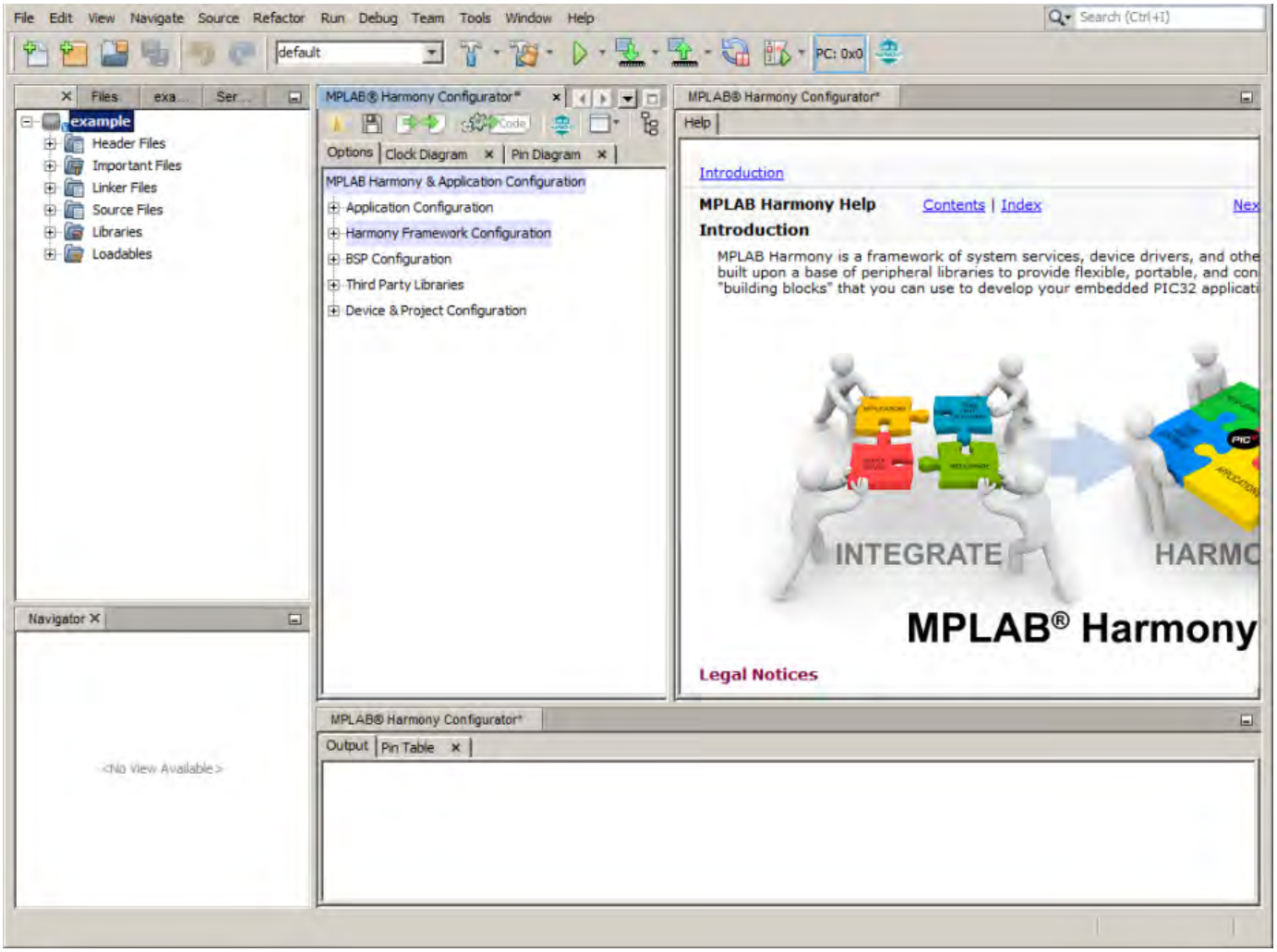


Main Window

This view shows the available configuration options for the selected Microchip device, which is arranged in a hierarchal tree structure. Click the check box to enable a specific component. The options for the enabled component will appear.

Help Window

When a tree component is interacted with, the corresponding help information is displayed in the Help Window.



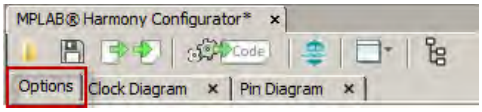
Output Window

The output window displays various log messages about the actions taken by the MPLAB Harmony Configurator.

Main Window Toolbar

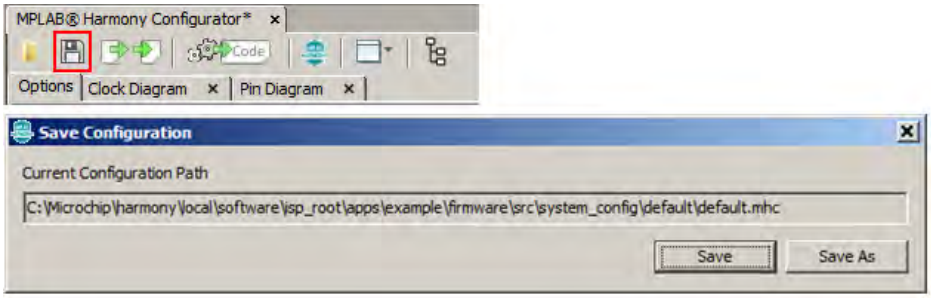
The main window contains a context-sensitive toolbar. This toolbar provides both global and tab-specific functionality.

When viewing the **Options** tab, this toolbar contains the following functionality:

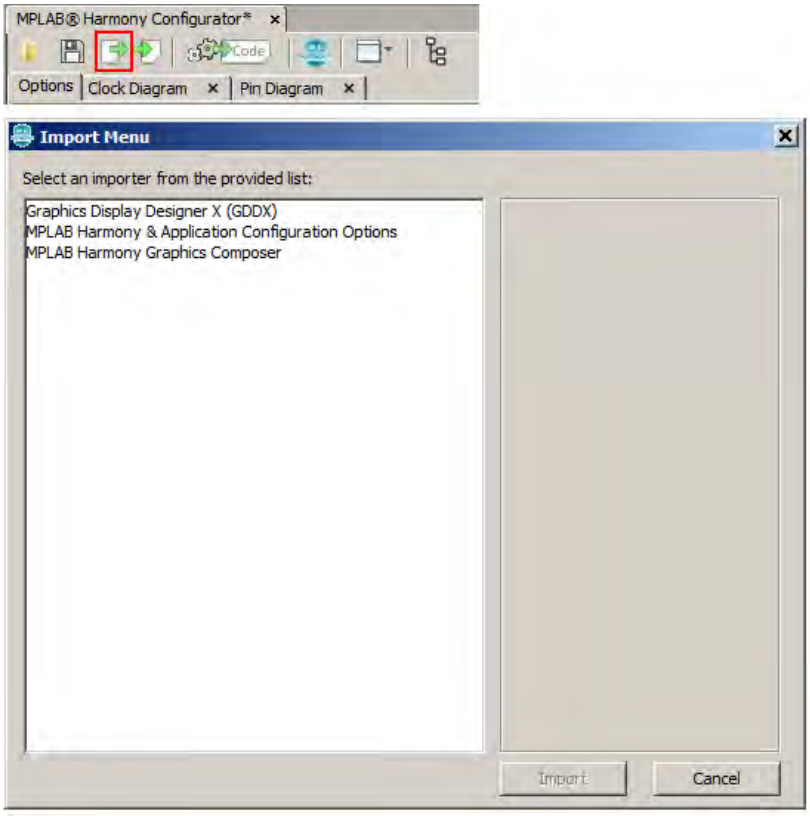


Open: Select the Open icon to open a saved `.mhc` configuration into the current Option tree.

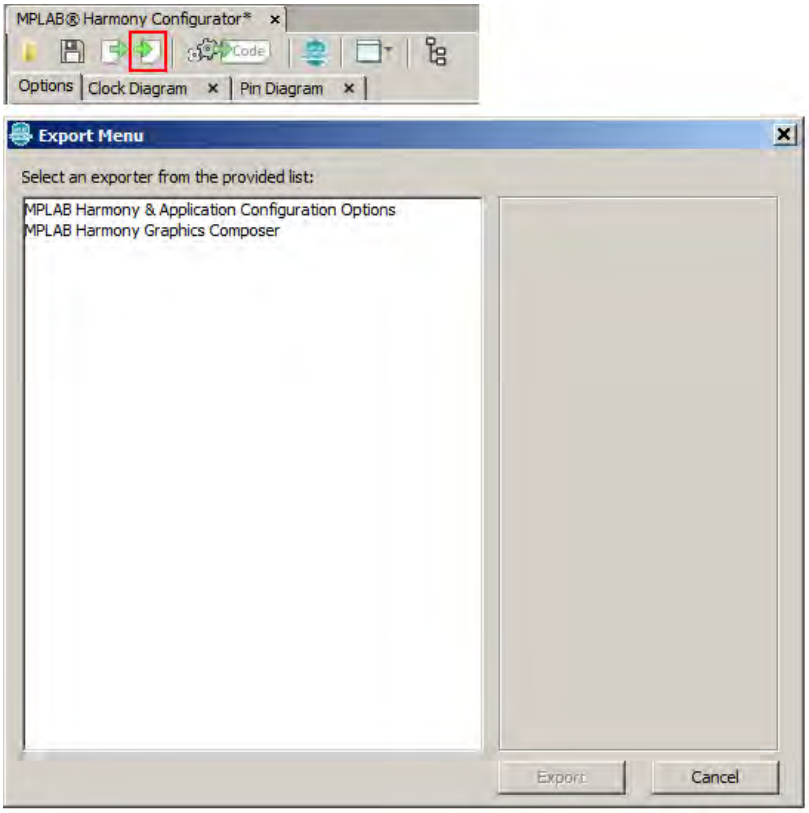
Save: Select the Save icon to save the current Option tree into the last used `.mhc` file or click **Save As** to save to a new file.



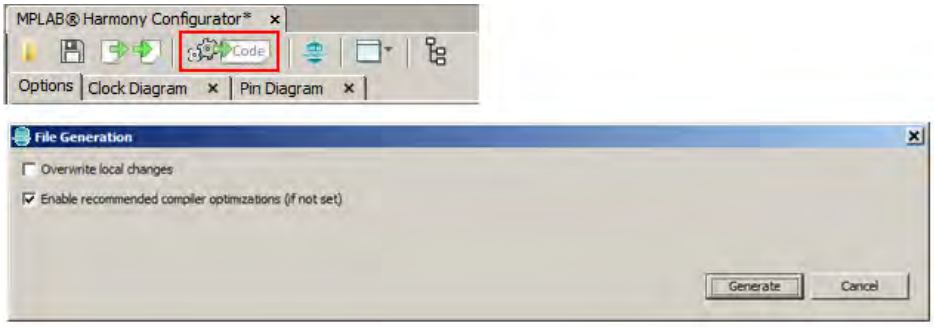
Import: Selecting the Import icon opens the import data dialog. This dialog can be used to import different types of information into the current project.



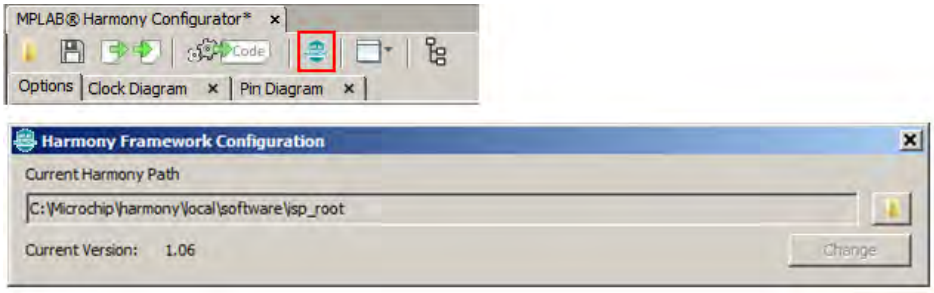
Export: Selecting the Export icon opens the export data dialog. This dialog can be used to export different types of information from the current project.



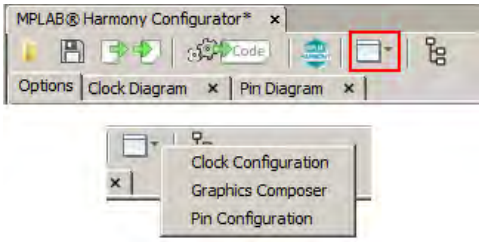
Generate: Selecting the Generate icon opens the project file generation dialog.



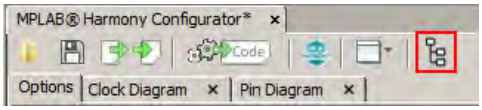
Framework Options: Selecting the Framework Options icon opens the framework configuration dialog.



Application Launcher: Selecting the Application Launcher icon provides the ability to quickly launch applications such as the clock configurator, pin configurator, or the MPLAB Harmony Graphics Composer.

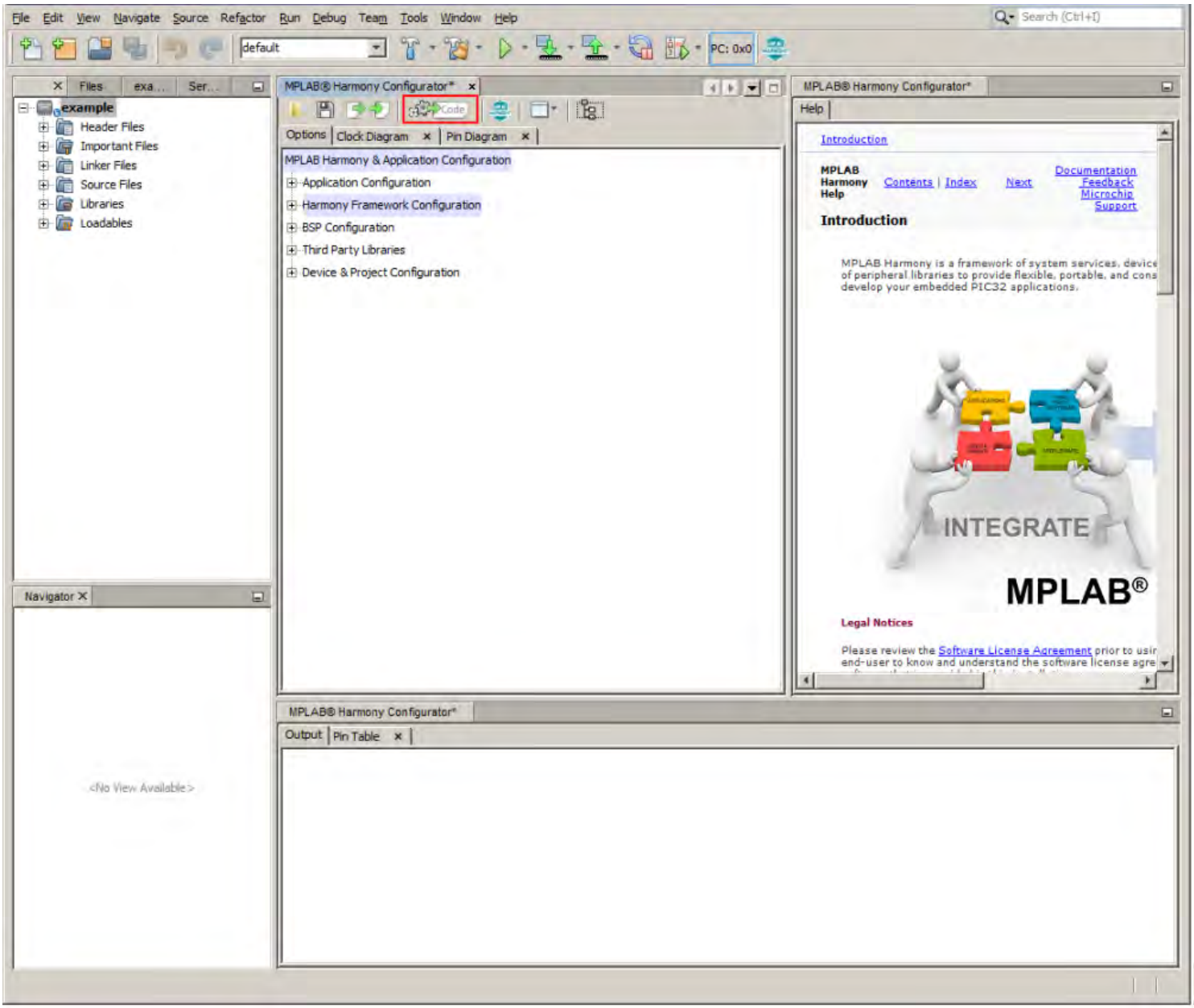


Option Tree View: Selecting the Option Tree View icon toggles the option tree between global and active view.



Project Generation

Once all of the desired options have been selected from the configuration tree, the next step is file generation, which is done by clicking **Generate** in the main window. Various options for generation are displayed in the File Generation dialog.



- Overwrite local changes – Automatically overwrites any local changes made by the user. A merge window will be displayed for all locally changes files if this option is not selected.
- Enable recommended compiler optimizations (if not set) – A compiler optimization level of at least 'O1' is highly recommended for MPLAB Harmony projects. This option will set the compiler optimization level to 'O1' if no optimization level is currently set.
- The Generate button will cause all of the selected components and options to be processed and output as valid code files. These files will be automatically added to the project.

Using MHC to Create a New Application

Provides information on creating a new MHC project.

Introduction

This section provides an introduction to creating your own MPLAB Harmony applications using the MPLAB Harmony Configurator (MHC).

Description

MPLAB Harmony provides a MPLAB Harmony Configurator (MHC) MPLAB X IDE plug-in that can be installed in MPLAB X IDE to help you create your own MPLAB Harmony applications.

To create a new MPLAB Harmony application with MHC, follow these three steps:

- Step 1: Create the New Harmony Project
- Step 2: Add and Configure Required Libraries/Modules
- Step 3: MPLAB Harmony Application Structure and Developing the Application

Note: If you are a Microchip Libraries for Applications (MLA) user, and will be porting your application from the MLA TCP/IP, File System, USB Device, Graphics, or peripheral libraries to the MPLAB Harmony equivalents, refer to Porting to MPLAB Harmony for more information.

Prerequisites

This topic describes the prerequisites for creating your own MPLAB Harmony applications using MHC.

Description

This tutorial assumes that you have already completed these steps before you start:

1. Installed the MPLAB X IDE (http://www.microchip.com/mplabx).
2. Installed MPLAB Harmony (http://www.microchip.com/harmony).
3. Installed the MPLAB XC32 C/C++ Compiler (http://www.microchip.com/xc32).
4. Set up a working PIC32 development platform (http://www.microchip.com/32bit).

You can download the MPLAB X IDE, MPLAB Harmony and the MPLAB XC32 C/C++ Compiler from the links provided. If you do not already have a PIC32 development platform, you can learn more about the PIC32 family and determine which hardware platform best meets your development needs by visiting the 32-bit website listed previously.

This tutorial also assumes that you have some familiarity with the MPLAB X IDE, embedded C-language programming and PIC32 microcontrollers. If you are unsure how to complete some of the steps in this tutorial, please refer to the documentation for the item on which you have questions. You may also seek assistance from your peers on the Microchip discussion forums (http://www.microchip.com/forums) or from the Microchip support staff (www.microchip.com/support).

Once you have everything installed, connected, and up and running you are ready to begin creating your own MPLAB Harmony applications.

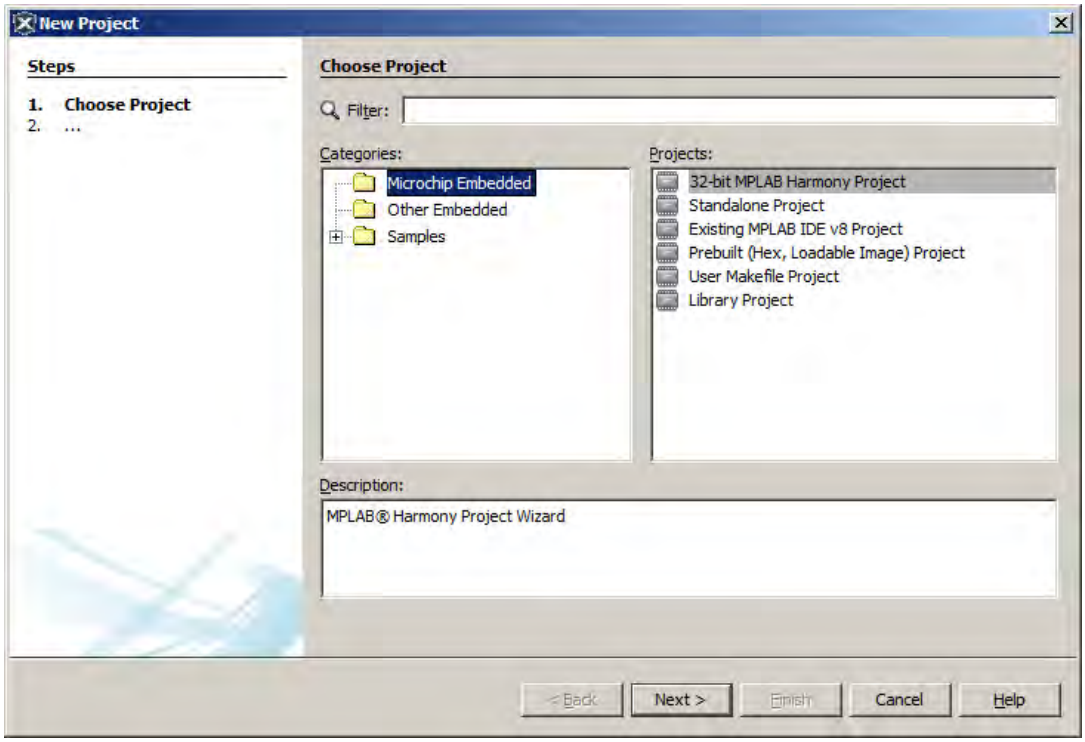
Step 1: Create the New Project

To create a new MPLAB Harmony project, you first need to create a new MPLAB X IDE project and the basic set of source code files and functions that are necessary for a properly formed MPLAB Harmony application.

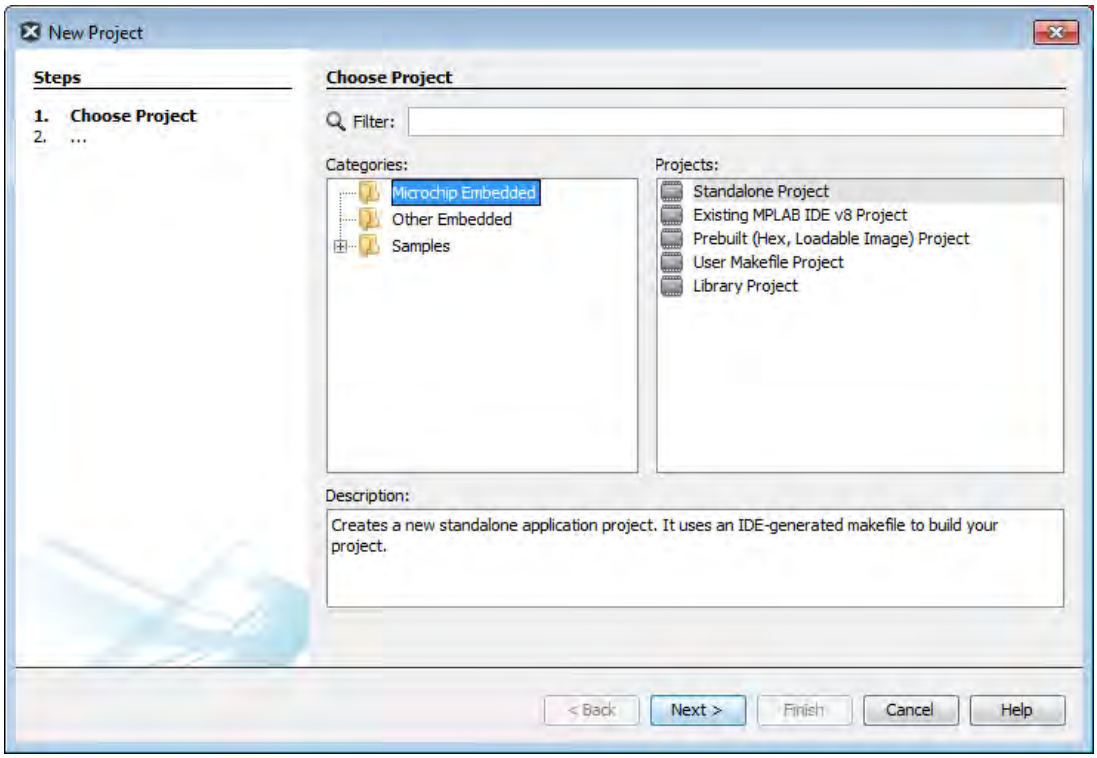
Description

To create a new MHC project:

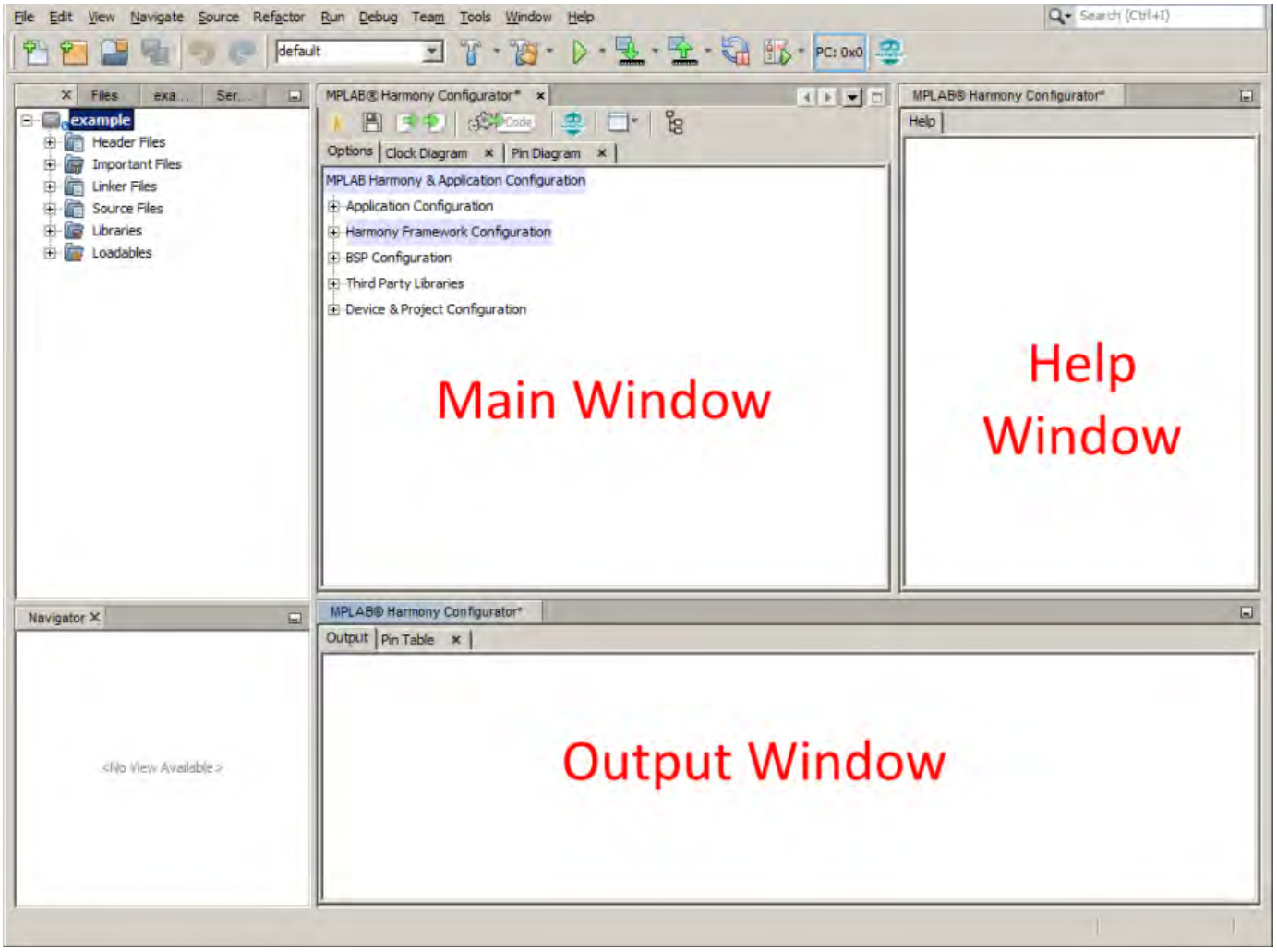
1. Select *File* > *New Project* or click the New Project icon in MPLAB X IDE.
2. In Categories, select **Microchip Embedded** and in Projects select **MPLAB Harmony Project** from the list of available project templates, and then click **Next** to launch the Microchip Harmony Configurator Project Wizard.



3. Specify the following in the New Project dialog:
   - Harmony Path (path to the folder containing Harmony framework: `<install-dir>`)
   - Project Location (the default project path is the `apps` folder within the selected MPLAB Harmony path)
   - Project Name
   - Configuration Name (optional)
   - Target Device (when a valid harmony path is selected, the device selection menu will be filled)



4. A MPLAB Harmony project will be created and the MPLAB Harmony Configurator will open. Refer to MPLAB Harmony Configurator for additional information.



Step 2: Add and Configure the Required Libraries and Modules

This topic describes how to configure the MPLAB Harmony library modules.

Description

1. In the Main window, expand the Device Configuration tree and select the desired device configuration settings.
2. Expand the MPLAB Harmony Project Configuration tree and select and configure the desired libraries.
3. If use of a Board Support Package is desired, expand the BSP Configuration tree and select the desired BSP.
4. When complete, generate and save the configuration.
5. Develop your application logic using the selected libraries.

At this point, you should be able to build, debug, and step through the application. Effectively, you have a running MPLAB Harmony system; however, it is not yet ready to do anything. Next, you will develop your application state machine logic and make sure the system does what you want it to do.

Step 3: MPLAB Harmony Application Structure and Developing the Application

This topic describes the steps necessary to maintain the state machines.

Description

main.c

The main.c file contains calls to the SYS_Initialize function, which initializes MPLAB Harmony modules, as well as applications. It also contains the main task execution, which calls tasks for all selected MPLAB Harmony modules, as well as the application task function, APP_Tasks.

app.c

The `app.c` file contains the APP_Initialize function that is used to place an application into its initial state. It will be called from the SYS_Initialize

function. The APP_Task function, which is also contained in the `app.c` file, implements the application state machine logic. Add application code to this task as desired.

Refer to the example applications located in the `<install-dir>/apps/` folder within your MPLAB Harmony installation for example applications for various MPLAB Harmony modules. Related documentation is available in the *Applications Help* > *Examples* section.

Porting a Legacy PLIB to MPLAB Harmony

Provides an example on how to port a legacy (i.e., prior to MPLAB Harmony) USART Peripheral Library (PLIB) demonstration application to a MPLAB Harmony application using the MPLAB Harmony Configurator (MHC).

Description

A detailed procedure for porting the legacy UART PLIB Interrupt demonstration application (`<compiler-install-dir>/examples/plib_examples/uart/uart_interrupt`) to MPLAB Harmony is provided in the Framework Help > Peripheral Library Help > Peripheral Library Porting Example .

In this example, the following assumptions are made:

- The PIC32MX795F512L device will be used; however, the process described in this section is applicable for other PIC32 devices with appropriate changes
- The Explorer 16 Development Board is the hardware used in this example
- For the v1.33 MPLAB XC32 C/C++ Compiler, the `examples` folder is not present. To view the legacy USART PLIB example, refer to v1.31 or earlier of the MPLAB XC32 C/C++ compiler.

Configuring the Oscillator Module Using the MHC Clock Configurator

Provides information configuring the Oscillator module using the MHC Clock configurator

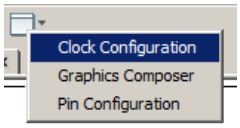
Description

The MHC Clock Configurator is a component of the MPLAB Harmony Configurator (MHC) MPLAB X IDE plug-in. Its function is to provide a graphical user interface to configure the Oscillator module.

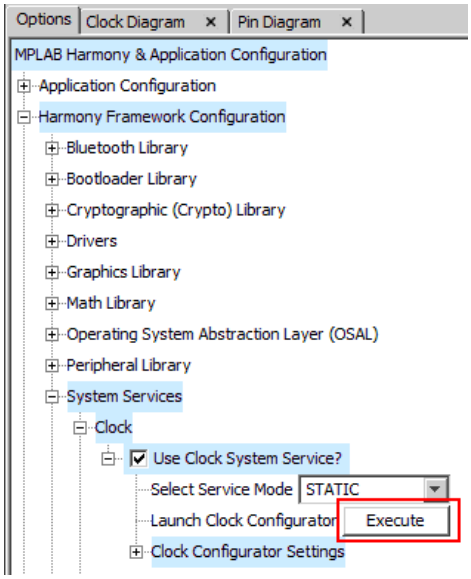
While simulating the normal operation of the Oscillator module, the MHC Clock Configurator contains interactive controls, dynamic output, and visual warnings to help guide the user in establishing the desired system clock configuration.

The MHC Clock Configurator is launched automatically when the MHC is launched. It is in the form of a tab panel in MPLAB X IDE. Clicking the MPLAB Harmony Clock Configuration tab will open the MHC Clock Configurator.

The clock configurator screen can also be accessed using the main window toolbar application launch feature. Simply click the application launch icon and select **Clock Configuration**.



Another way to access the MHC Clock Configurator is via the Clock System Service section in MHC Harmony & Application Configuration tree view. Pressing the Execute button at the Launch Clock Configurator topic will either bring the tab panel into focus or launch the MHC Clock Configurator, if the tab panel was closed.



Note: The MHC Clock Configurator is one option to configure the Oscillator Module. Another option is to configure directly via the MPLAB Harmony & Application Configuration tree structure. The majority of the settings captured in the MHC Clock Configurator exist under the Clock Configurator Settings node in the Clock System Service, while the remainder are in the Device Configuration section.

Clock Configuration for PIC32MZ Family Devices

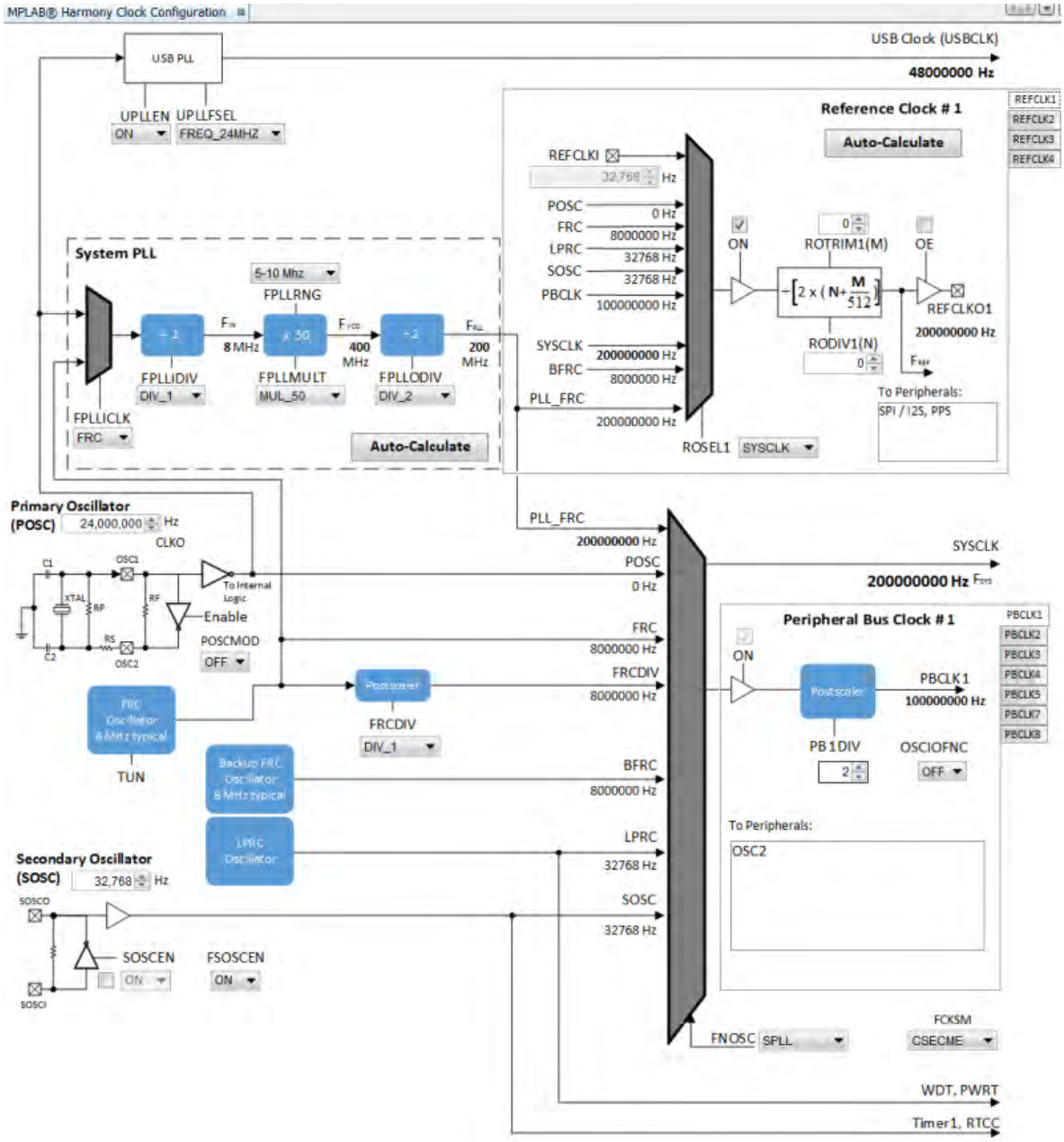
Provides configuration information for PIC32MZ family devices.

Description

The MHC Clock Configurator's support of configuring the Oscillator Module of a PIC32MZ family device is divided into the following sub-sections:

- Configuring System Clock Frequency
- Configuring the Peripheral Bus Clocks
- Configuring the Reference Clocks
- Using the SPLL Divider Auto-Calculate Feature

For details regarding the operation of the Oscillator module, refer to the **"Oscillator"** chapter in the *"PIC32MZ Embedded Connectivity (EC) Family Data Sheet"* (DS60001191). This document is available for download from the Microchip website (www.microchip.com).



Configuring the System Clock Frequency

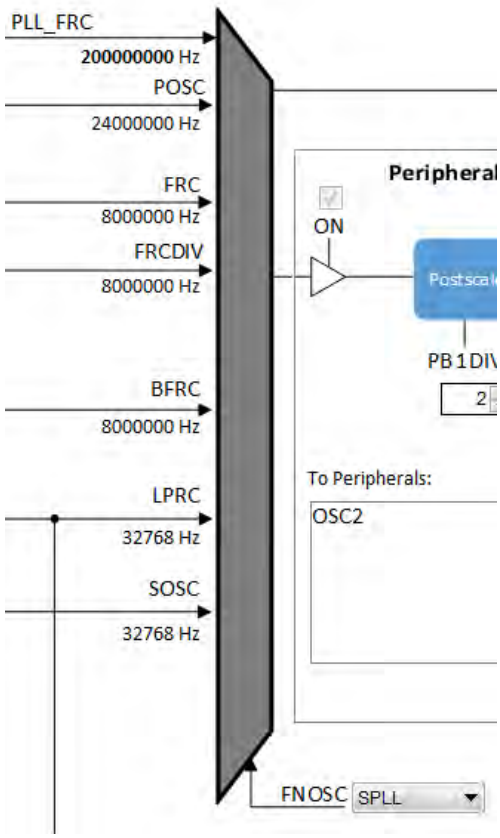
Provides information on configuring the system clock frequency for PIC32MZ family devices.

Description

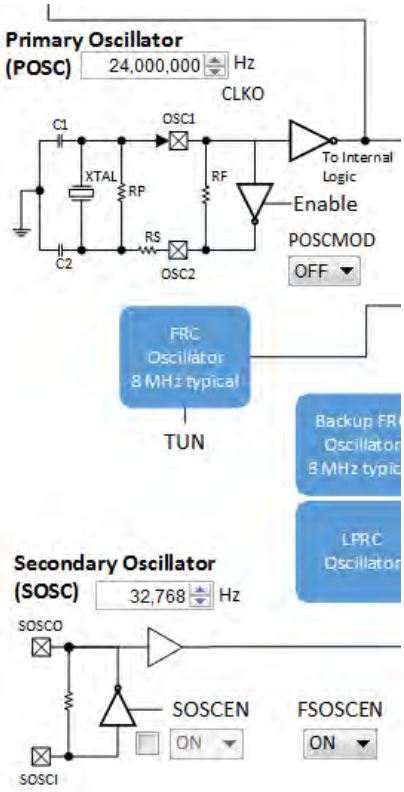
There are a total of five external and internal oscillator options as clock source:

- Internal Fast RC (FRC) Oscillator divided by the FRCDIV bits in the OSCCON register
- Internal Low-Power RC (LPRC) Oscillator
- Secondary Oscillator (SOSC)
- Primary Oscillator (POSC) (POSCMOD: HS or EC)
- System PLL (SPLL)

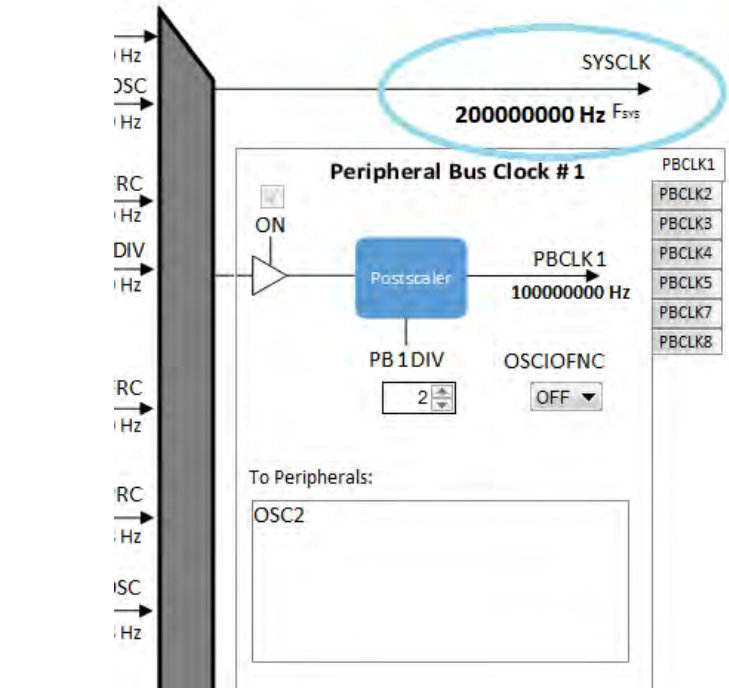
The device configuration bit FNOSC is represented as a drop-down with the above selections in the MHC Clock Configuration. The current selection is represented in **bold**.

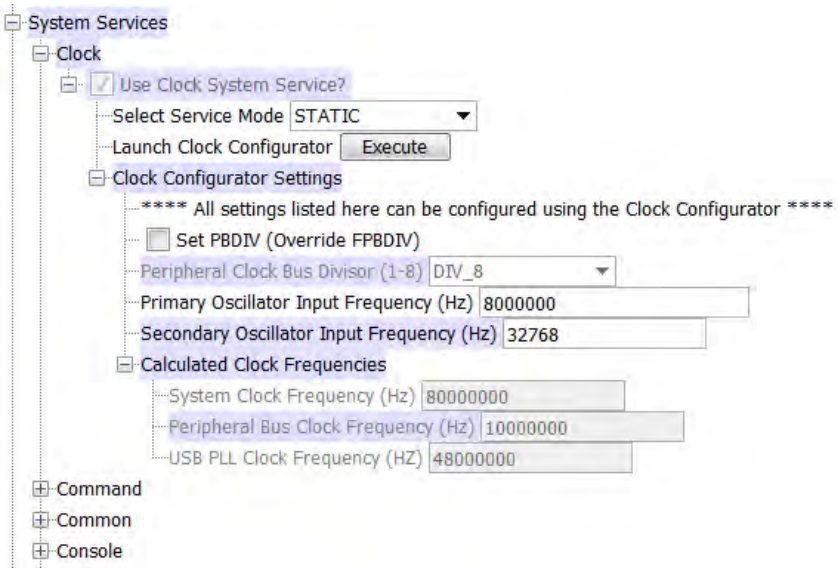


The Primary Oscillator (POSC) and Secondary Oscillator (SOSC) are customizable external clock sources. For the POSC, the device configuration bit, POSCMOD, needs to be set to EC or HS. If FNOSC is set to SOSC, the device configuration bit, FSOSCEN, should be set to ON. SOSCEN is set post-initialization. There is an option to override FSOSCEN with SOSCEN.

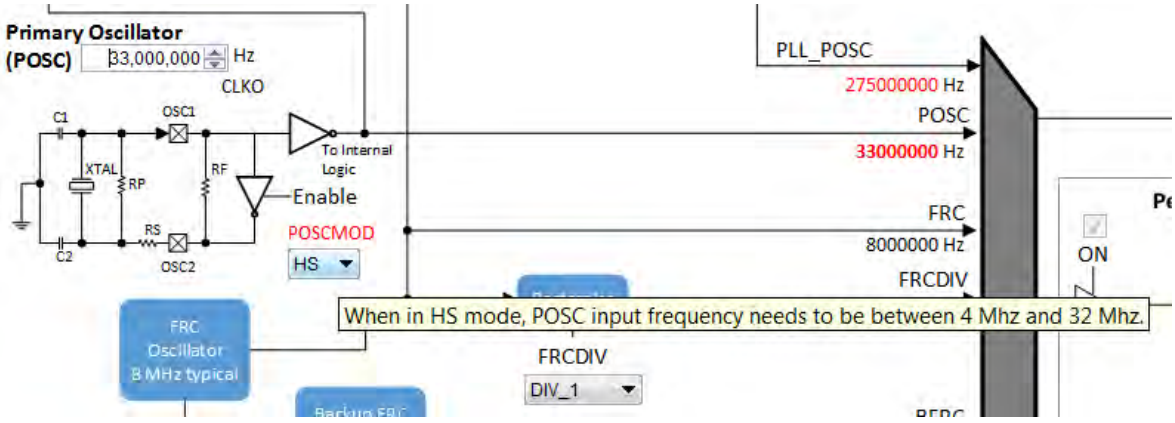


The output system clock frequency (SYSCLK) is displayed on the left side. This value (in Hz) corresponds to System Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

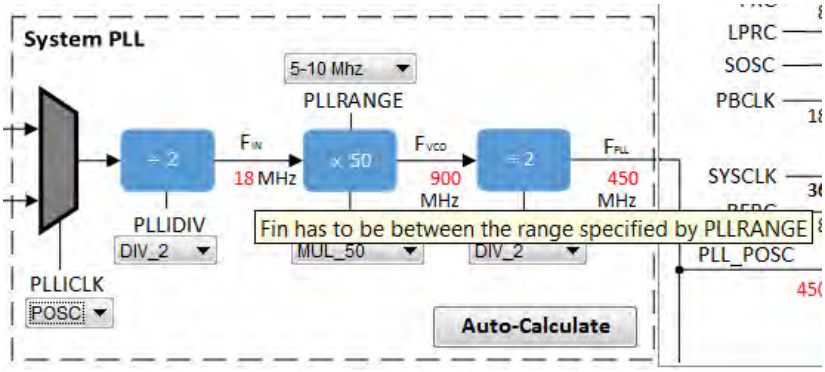




Certain frequency values may be displayed in red when the input value does not meet specification and may cripple performance of the device. An example is shown in the following figure, when the HS Oscillator Mode is selected for POSCMOD and the POSC input frequency set is outside of the 4 MHz - 32 MHz range. A dynamic help tip will also appear if the user hovers over the POSCMOD control or any of the red text.



Another example is the SPLL, where FPLL (60 MHz – 120 MHz), FVCO (80 MHz – 240 MHz), and FIN (range specified by PLLRANGE) will appear as red text, including an explanation tool tip, if they fall outside of their respective required ranges.

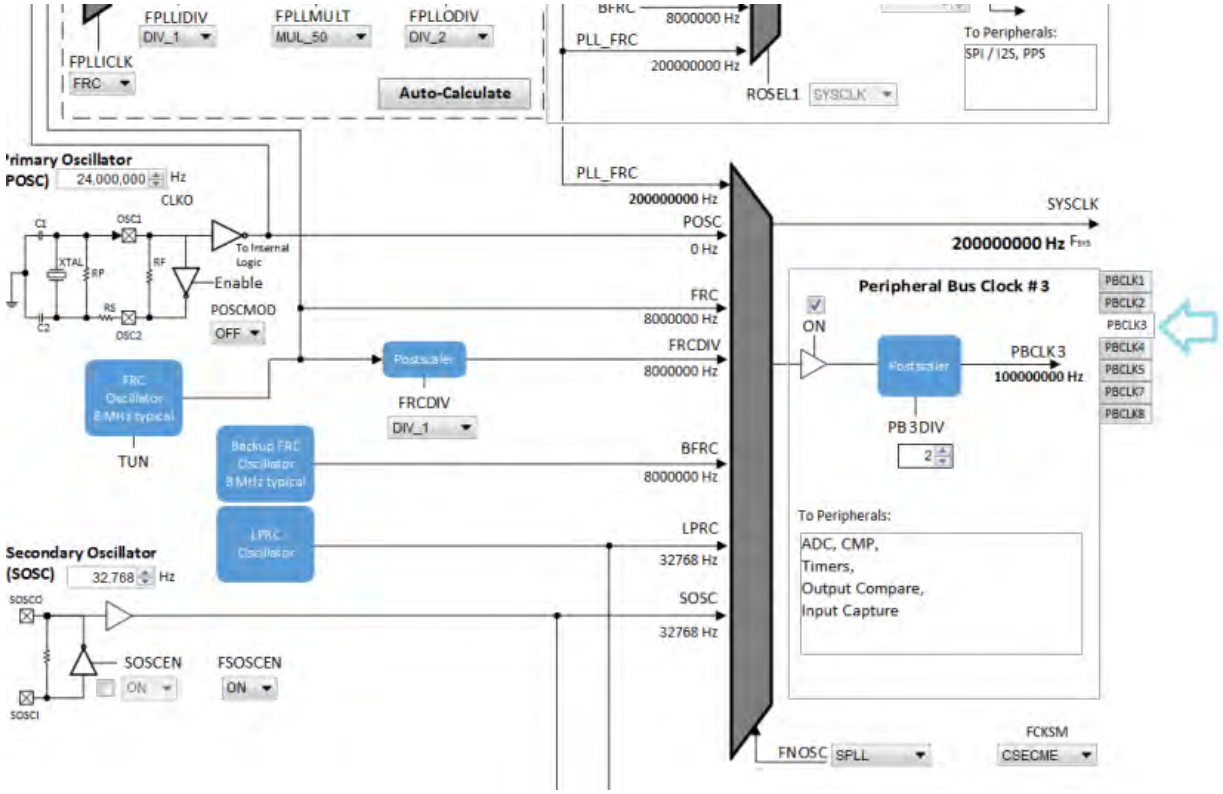


Configuring the Peripheral Bus Clocks

Provides information on configuring the peripheral bus clocks for PIC32MZ family devices.

Description

Each of the eight Peripheral Bus Clocks on the PIC32MZ family devices can be configured by using the tabs on the left.



The output frequency is in **bold**. The “To Peripherals” window provides a reminder of which peripherals each clock is driving.

This value (in Hz) corresponds to Peripheral Bus Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

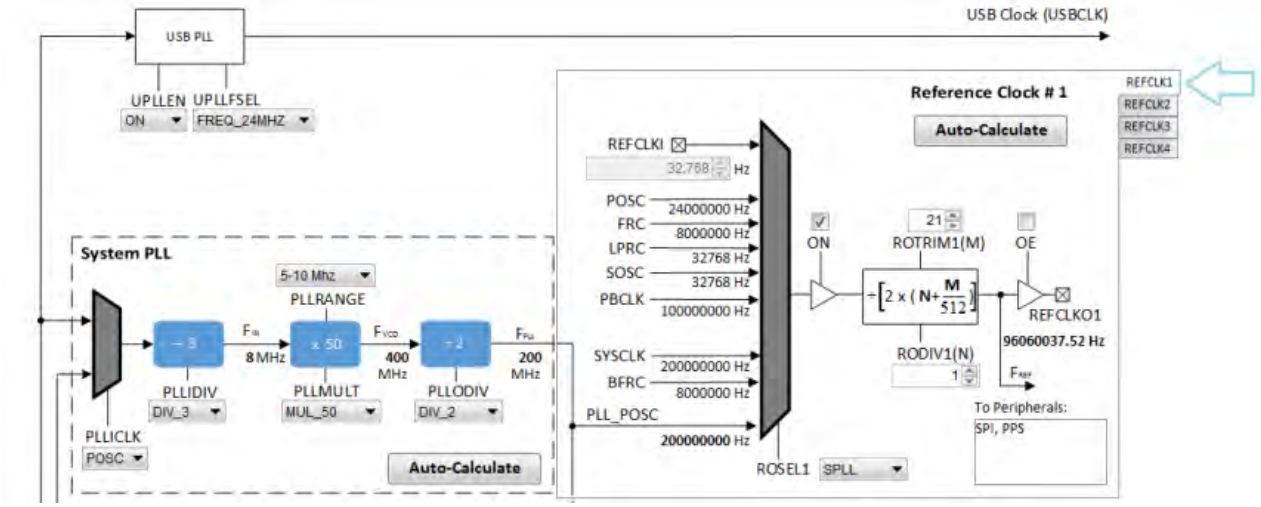
Note: It is important to know the acceptable clock range for the peripherals. The Clock Configurator will NOT provide a warning if the output peripheral clock frequency falls outside of the specified range of the peripheral.

Configuring the Reference Clocks

Provides information on configuring the reference clocks for PIC32MZ family devices.

Description

Each of the four Reference Clocks on the MZ Family of device can be configured by using the tabs on the left.



The clock input source (ROSELx), divider (RODIVx), trim value (ROTRIMx) are independently configurable. The output frequency (REFCLKOx) is in **bold**.

This value (in Hz) corresponds to Reference Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in the MHC Harmony & Application Configuration tree view.

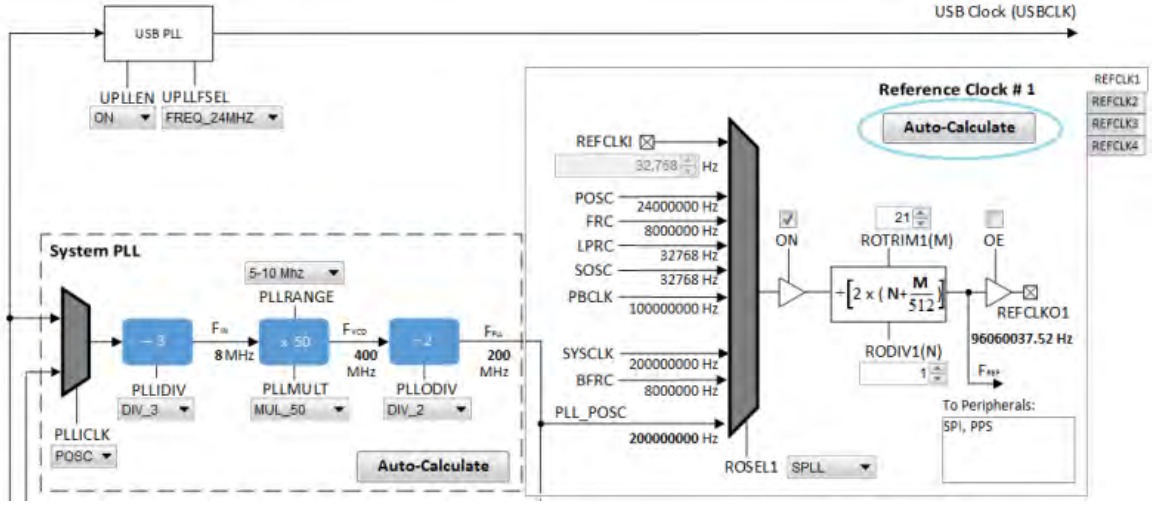
Using the Reference Clock Auto-Calculate Feature

Provides information on the reference clock auto-calculate feature for PIC32MZ family devices.

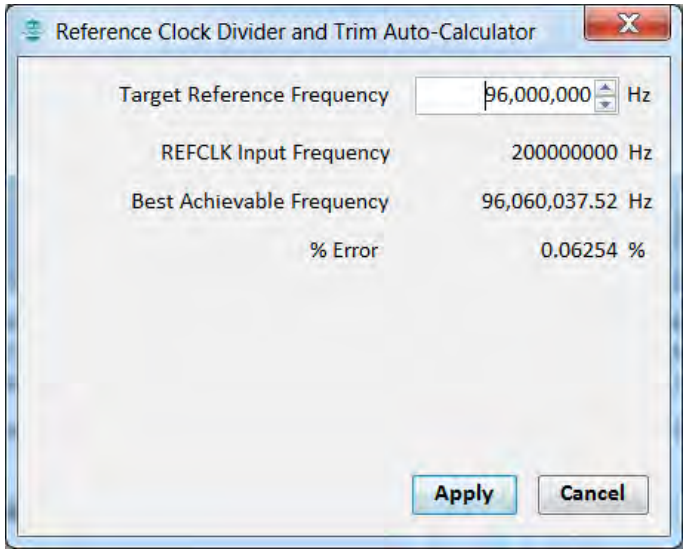
Description

The MHC Clock Configurator is equipped with the ability to help the user establish the closest possible match to a user-desired target reference clock frequency. The Auto-Calculate feature is designed to determine the divider and trim values in the each of the four reference clocks based on a user requested clock output frequency.

The feature can be accessed via the Auto-Calculate button in the Reference Clock section of the Clock Configurator.

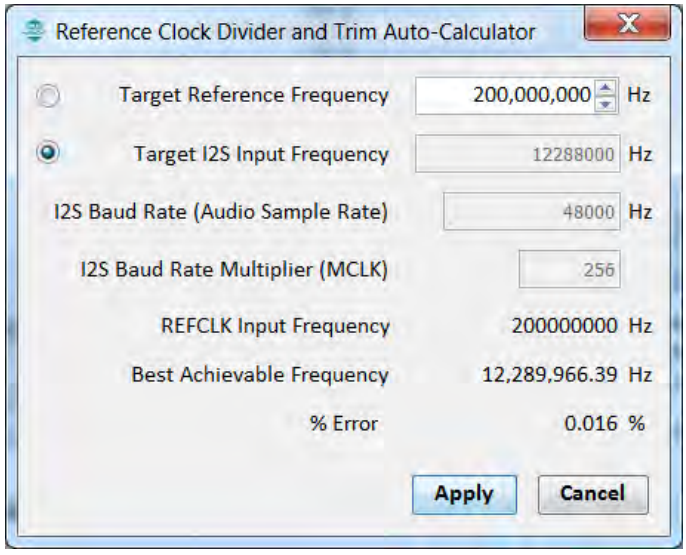


Clicking the **Auto-Calculate** button opens the Auto-Calculate dialog.



Enter the desired target reference frequency (remember to press the <Enter> key), and the dialog window will display the best achievable frequency that can be provided by the Reference Clock Divider (RODIVx) and Trim (ROTRIMx) combination, as well as the percentage discrepancy from the desired value, if any. The REFCLK Input Frequency is determined based on selection at ROSELx.

If the I2S driver is selected as part of the configuration, the Reference Clock Divider and Trim Auto-Calculator dialog opens automatically reconfigured with the option to use the target I2S input frequency as the target reference frequency.



Clicking the **Apply** button will cause the MHC Clock Configurator to update the Reference Clock divider and trim to establish the closest achievable frequency.

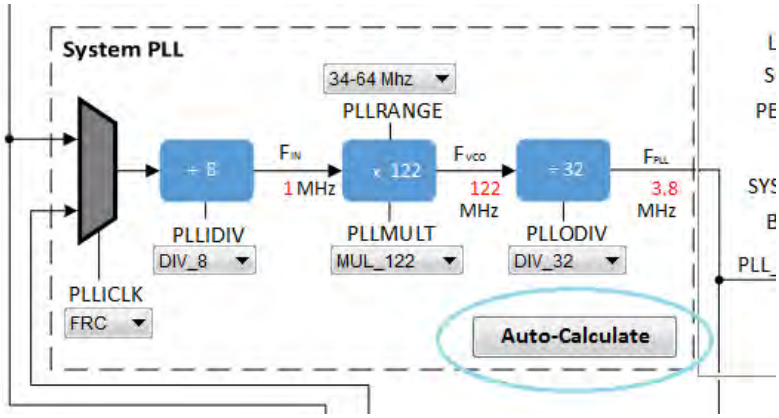
Using the SPLL Divider Auto-Calculate Feature

Provides information on the SPLL auto-calculate feature for PIC32MZ family devices.

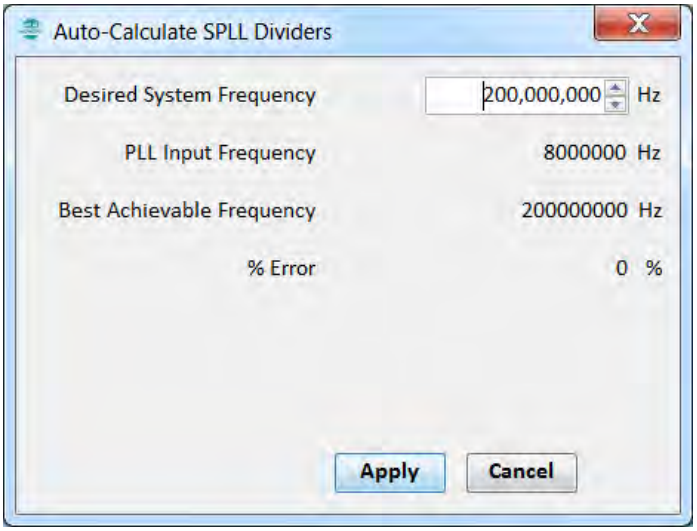
Description

The MHC Clock Configurator is equipped with the ability to help the user establish closest possible match to a user-desired target system clock frequency. The Auto-Calculate feature is designed to determine the divider and multiplier values in the SPLL-based on a user requested system clock frequency.

The feature can be accessed via the Auto-Calculate button in the SPLL section of the Clock Configurator.



Clicking the **Auto-Calculate** button opens the Auto-Calculate dialog.



Enter the desired system clock frequency (remember to press the key ENTER), and the dialog window will display the best achievable frequency that can be provided by the SPLL divider/multiplier combination, as well as the percentage discrepancy from the desired value, if any. The PLL Input Frequency is determined based on selection at PLLICLK (FRC or POSC).

Clicking the **Apply** button will cause the MHC Clock Configurator to update the SPLL dividers and multiplier to establish the closest achievable frequency.

Note: The Auto-Calculate feature will also update the PLLRANGE setting to satisfy the necessary FIN frequency.

Clock Configuration for PIC32MX Family Devices

Provides configuration information for PIC32MX family devices.

Description

The MHC Clock Configurator's support of configuring the Oscillator Module of a MX Family Device is divided into the follow sub-sections:

- Configuring the System Clock Frequency
- Configuring the Peripheral Bus Clock
- Configuring the Reference Clock
- Configuring the USB PLL
- Using the SPLL Divider Auto-Calculate Feature

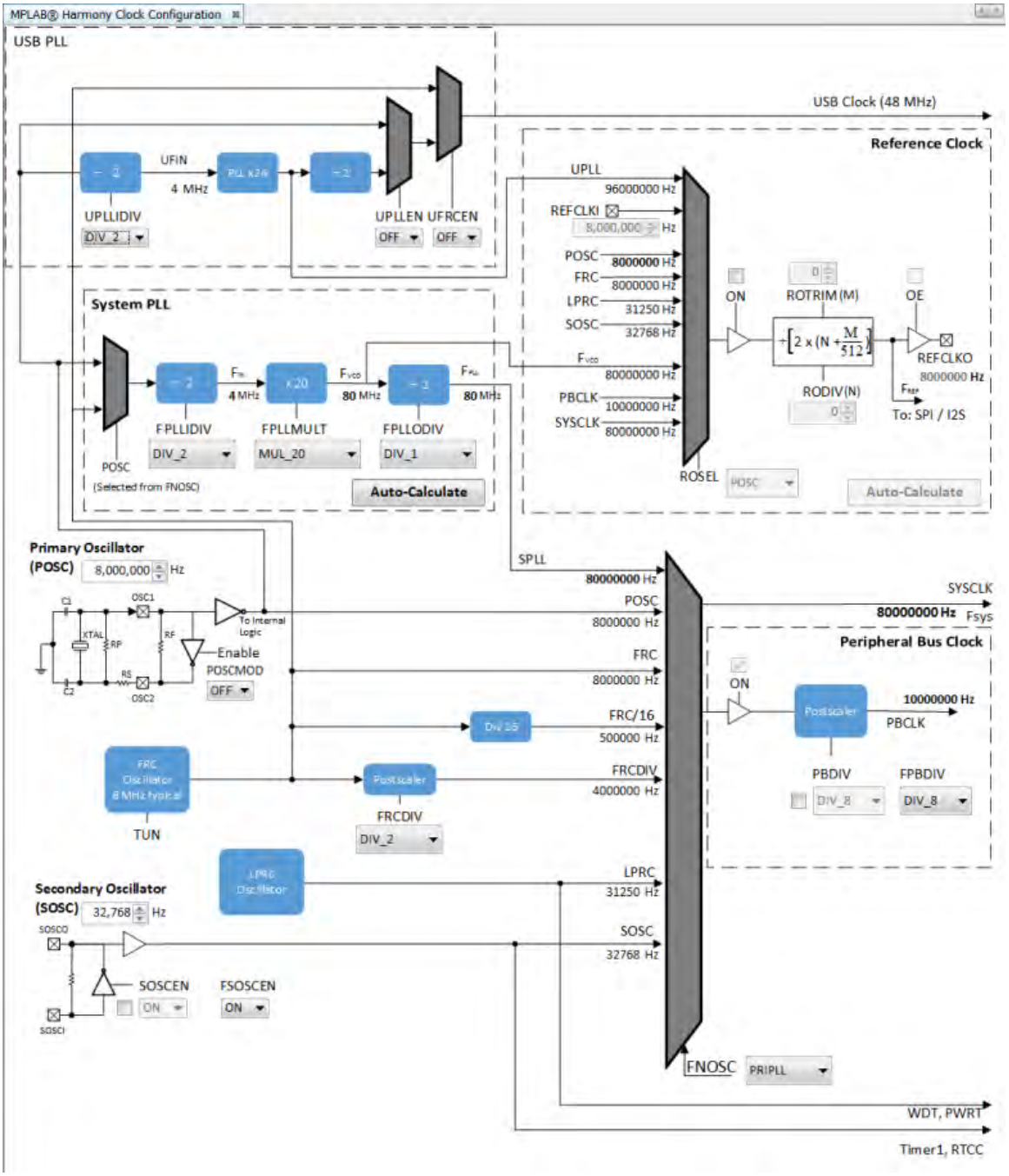
For details regarding the operation of the Oscillator module, refer to the **"Oscillator"** chapter in the specific PIC32MX device data sheet:

- PIC32MX1XX/2XX (DS60001168)
- PIC32MX1XX/2XX/5XX 64/100-pin Family (DS60001290)
- PIC32MX320/340/360/420/440/460 (DS60001143)
- PIC32MX330/350/370/430/450/470 (DS60001185)
- PIC32MX5XX/6XX/7XX (DS60001156)

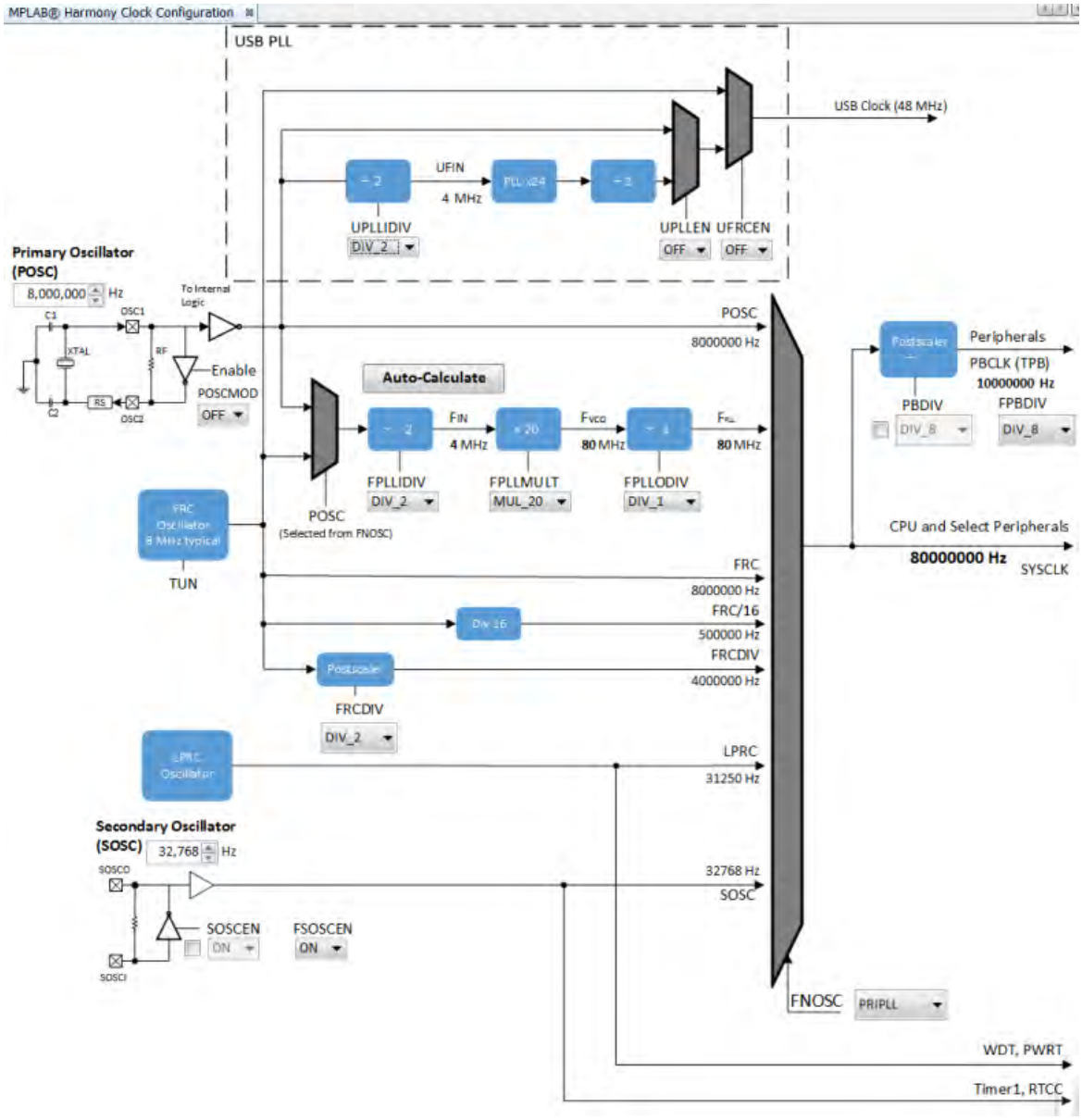
Each of these documents are available for download from the Microchip website (www.microchip.com).

The following figure shows the configuration screen for PIC32MX1XX/2XX, PIC32MX 330/350/370/430/450/470, and PIC32MX1XX/2XX/5XX

64/100-pin Family devices.



The next figure shows the configuration screen for PIC32MX320/340/360/420/440/460 and PIC32MX5XX/6XX/7XX devices.



Configuring the System Clock Frequency

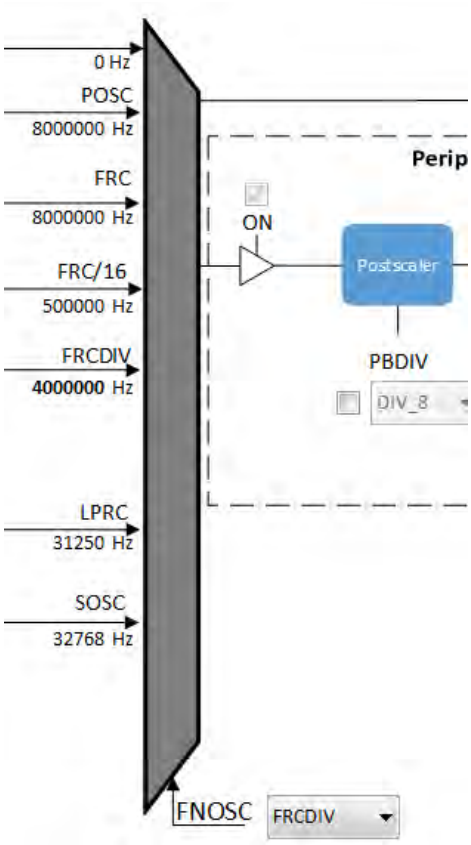
Provides information configuring the system clock frequency for PIC32MX family devices.

Description

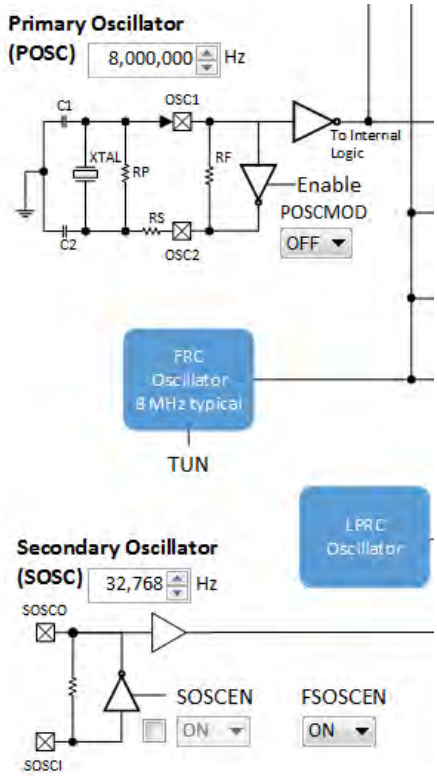
There are a total of five external and internal oscillator options as clock source:

- Internal Fast RC Oscillator (FRC) divided by the FRCDIV bits in the OSCCON register
- Internal Fast RC Oscillator (FRC) divided by 16
- Internal Low-Power RC (LPRC) Oscillator
- Secondary Oscillator (SOSC)
- Primary Oscillator with PLL module (PRIPLL)
- Primary Oscillator (POSCMOD: XT, HS, or EC)
- Internal Fast Internal RC Oscillator with PLL module via Postscaler (FRCPLL)
- Internal Fast Internal RC Oscillator (FRC)

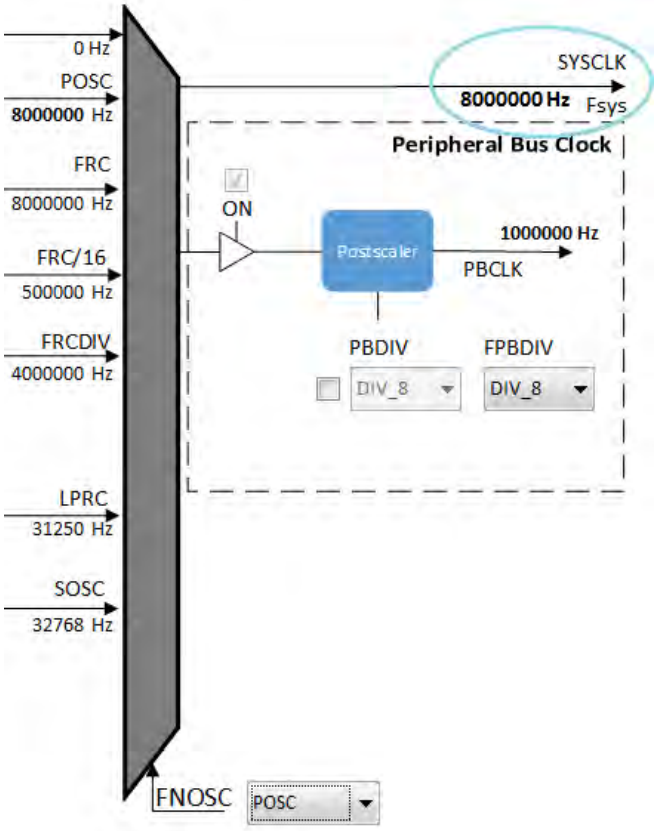
The device configuration bit FNOSC is represented as a drop-down with the above selections in the MHC Clock Configuration. The current selection is represented in **bold**.

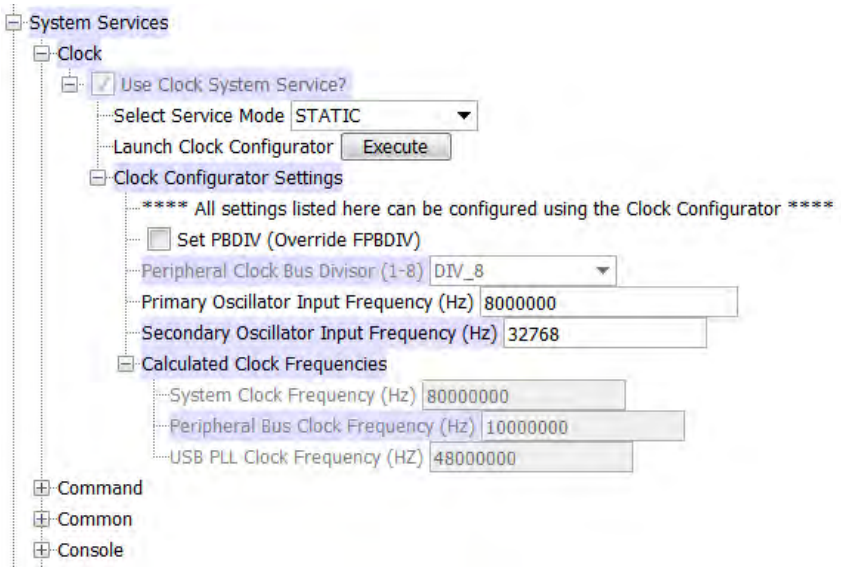


Primary Oscillator (POSC) and Secondary Oscillator (SOSC) are customizable external clock source. For POSC, the device configuration bit POSCMOD needs to be set to EC, XT, or HS. If FNOSC is set to SOSC, the device configuration bit FSOSCEN needs to be set to ON.

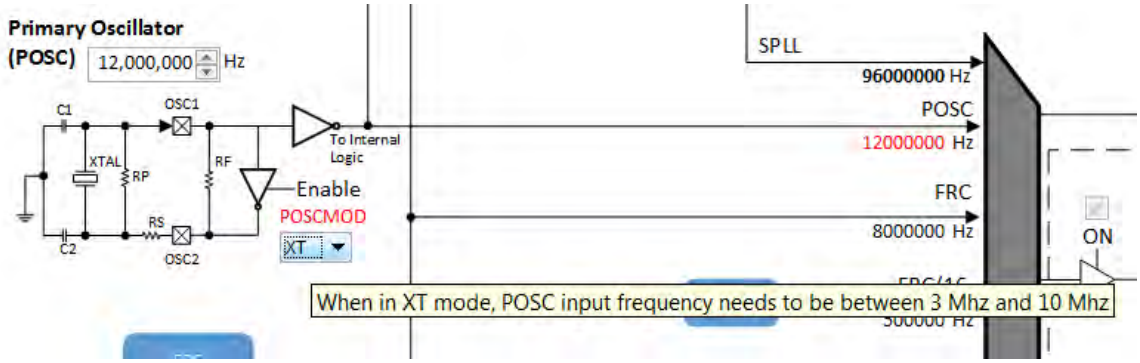


The output system clock frequency (SYSCLK) is displayed on the left side. This value (in Hz) corresponds to System Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

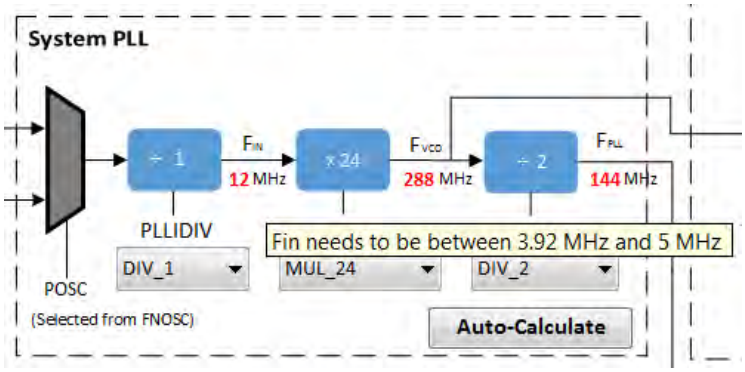




Certain frequency values may be displayed in red when the input value does not meet specification and may cripple performance of the device. An example is shown in the following figure, when the XT Oscillator Mode is selected for POSCMOD and the POSC input frequency set is outside of the 3 MHz - 10 MHz range. A dynamic help tip will also appear if the user hovers over the POSCMOD control or any of the red text.



Another example is the SPLL, where FPLL (40 MHz – 120 MHz), FVCO (60 MHz – 120 MHz), and FIN (3.92 MHz – 5 MHz) will appear in red text, including an explanation tool tip, if they fall outside of their respective required ranges.

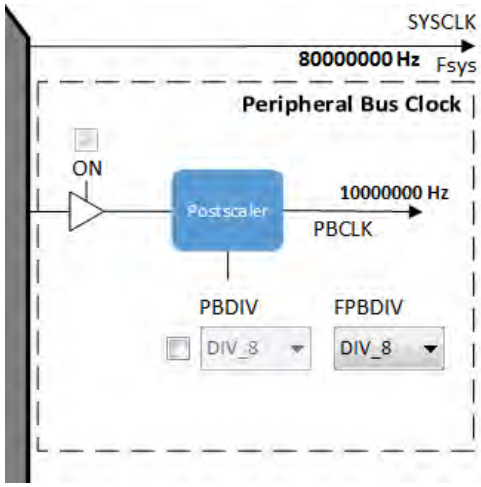


Configuring the Peripheral Bus Clock

Provides information on configuring the peripheral bus clock for PIC32MX family devices.

Description

The Peripheral Bus Clock on the MX Family of device can be configured on the left.



The output frequency is in **bold**. This value (in Hz) corresponds to Peripheral Bus Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

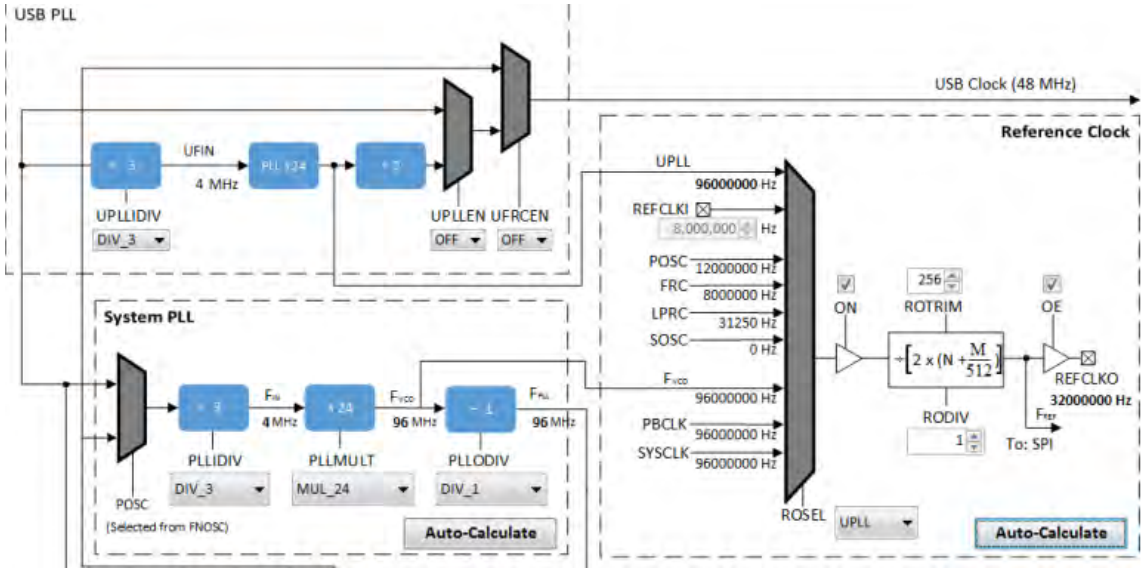
Note: It is important to know the acceptable clock range for the peripherals. The Clock Configurator will NOT provide a warning if the output peripheral clock frequency falls outside of specified range of the peripheral.

Configuring the Reference Clock

Provides information on configuring the reference clock for PIC32MX family devices.

Description

The Reference Clock on the PIC32MX1XX/2XX, PIC32MX 330/350/370/430/450/470, and PIC32MX1XX/2XX/5XX 64/100-pin Family devices can be configured in the section labeled Reference Clock on the upper right area of the screen.



The clock input source (ROSEL), divider (RODIV), trim value (ROTRIM) are independently configurable. The output frequency (REFCLKO) is in bold.

This value (in Hz) corresponds to Reference Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

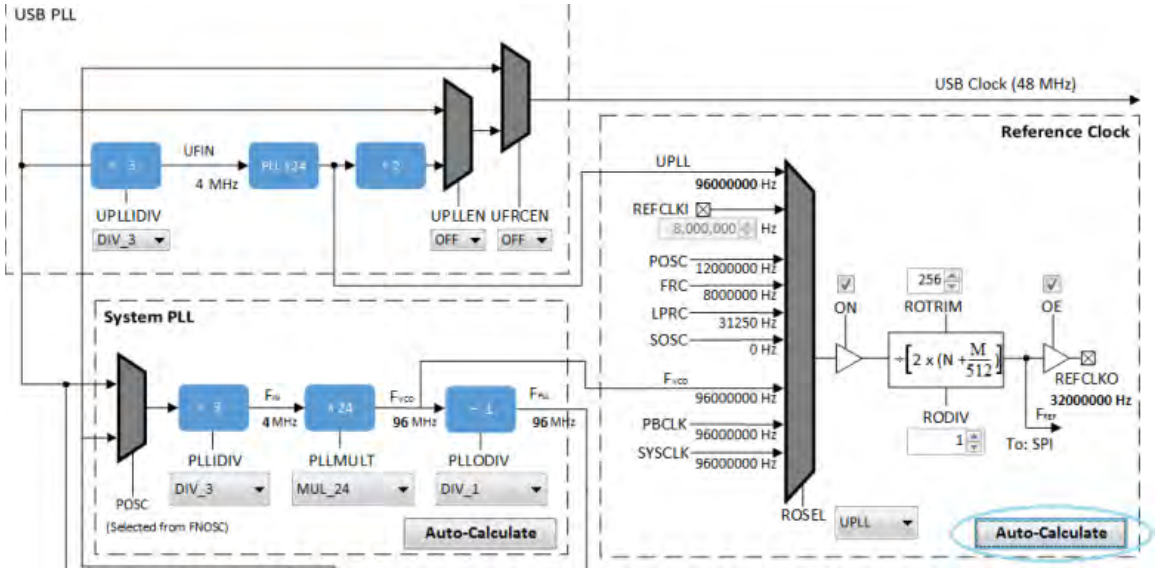
Using the Reference Clock Auto-Calculate Feature

Provides information on the reference clock auto-calculate feature for PIC32MX family devices.

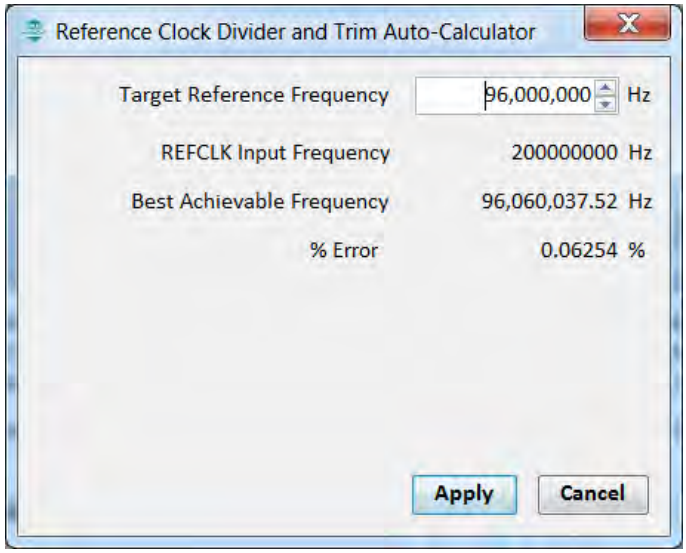
Description

The MHC Clock Configurator is equipped with the ability to help the user establish closest possible match to a user-desired target reference clock frequency. The Auto-Calculation feature is designed to determine the divider and trim values for the reference clock based on a user requested clock output frequency.

The feature can be accessed via the Auto-Calculate button in the Reference Clock section of the Clock Configurator.

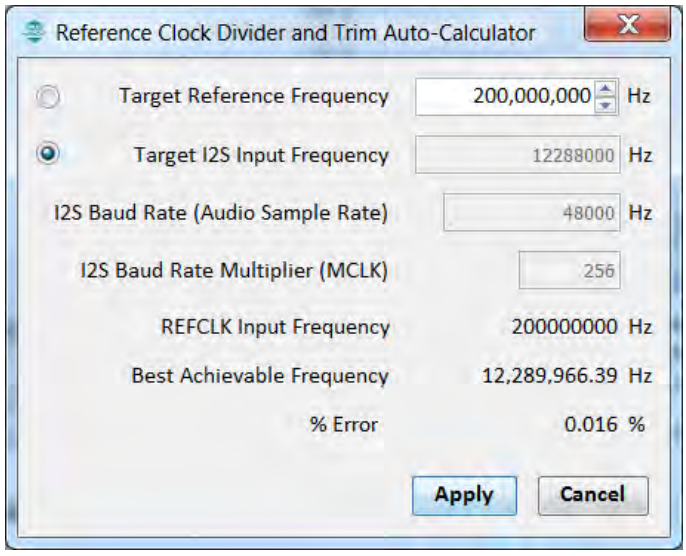


Clicking the **Auto-Calculate** button opens the Auto-Calculate dialog.



Enter the desired system clock frequency (remember to press the <Enter> key), and the dialog window will display the best achievable frequency that can be provided by the Reference Clock Divider (RODIV) and Trim (ROTRIM) combination, as well as the percentage discrepancy from the desired value, if any. The REFCLK Input Frequency is determined based on selection at ROSEL.

If the I2S driver is selected as part of the configuration, the Reference Clock Divider and Trim Auto-Calculator dialog opens automatically reconfigured with the option to use the target I2S input frequency as the target reference frequency.



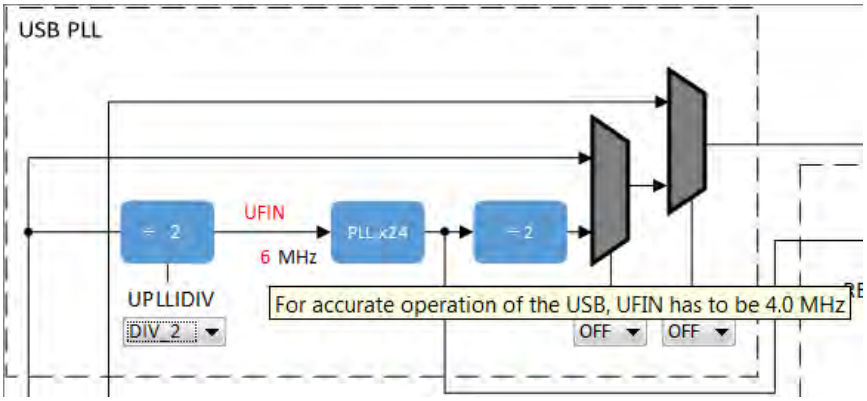
Clicking the **Apply** button will cause the MHC Clock Configurator to update the Reference Clock divider and trim to establish the closest achievable frequency.

Configuring the USB PLL

Provides information on configuring the USB PLL for PIC32MX family devices.

Description

Part of enabling the USB peripheral is to enable the USB PLL. The USB PLL requires 4 MHz input clock frequency for accurate operation. With POSC being a variable value, it is important to configure the correct USB PLL Input Divider (UPLLIDIV) value. The MHC Clock Configurator will provide visual warning if the value can lead to inaccuracy in USB operation.



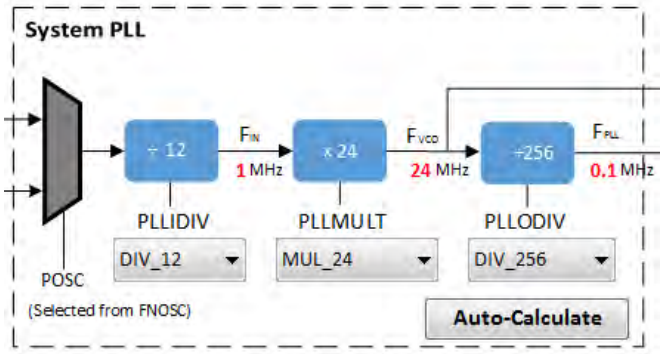
Using the SPLL Divider Auto-Calculate Feature

Provides information on using the SPLL Divider Auto-Calculate feature for PIC32MX family devices.

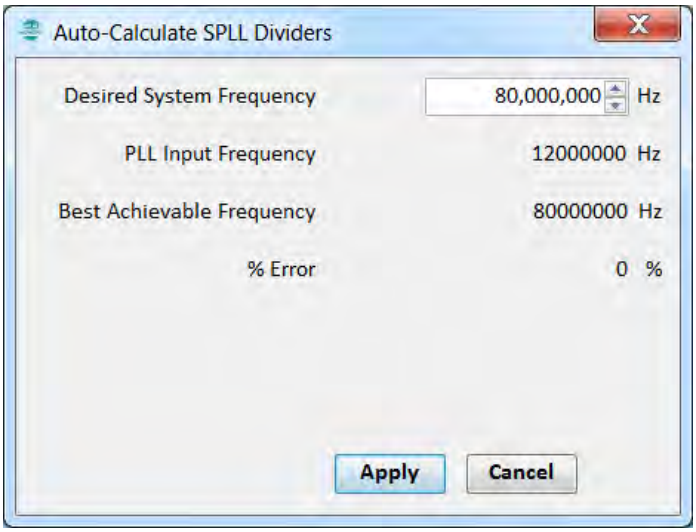
Description

The MHC Clock Configurator is equipped with the ability to help the user establish closest possible match to a user-desired target system clock frequency. The Auto-Calculation feature is designed to determine the divider and multiplier values in the SPLL-based on a user requested system clock frequency.

The feature can be accessed via the Auto-Calculate button in the System PLL section of the Clock Configurator.



Clicking the **Auto-Calculate** button opens the Auto-Calculate dialog.



Enter the desired system clock frequency (remember to press the <Enter> key), and the dialog window will display the best achievable frequency that can be provided by the SPLL divider/multiplier combination, as well as the percentage discrepancy from the desired value, if any. The PLL Input Frequency is determined based on selection at FNOSC (FRCPLL or PRIPLL).

Clicking the **Apply** button will cause the MHC Clock Configurator to update the SPLL dividers and multiplier to establish the closest achievable frequency.

MPLAB Harmony Graphical Pin Manager

Provides information on the MPLAB Harmony Graphical Pin Manager tool that resides within MHC.

Description

This graphical management tool exists for the purpose of enabling users to configure the pins of Microchip devices in a fast and intelligent manner. The tool consists of a graphical representation of the state of the component and table that provides the means to configure the pins of the device. Users intending to use this tool should be familiar with the MPLAB Harmony configuration tree.

The user configures a device using the following process:

- Launch the tool (if not already running)
- Add modules by enabling desired functionality in the configuration tree (e.g., USART or SPI)
- Using the pin table to “Lock” cells representing function and pin pairings
- Using the pin flag management dialog to change pin register values
- Generating resultant code through the **Generate** button

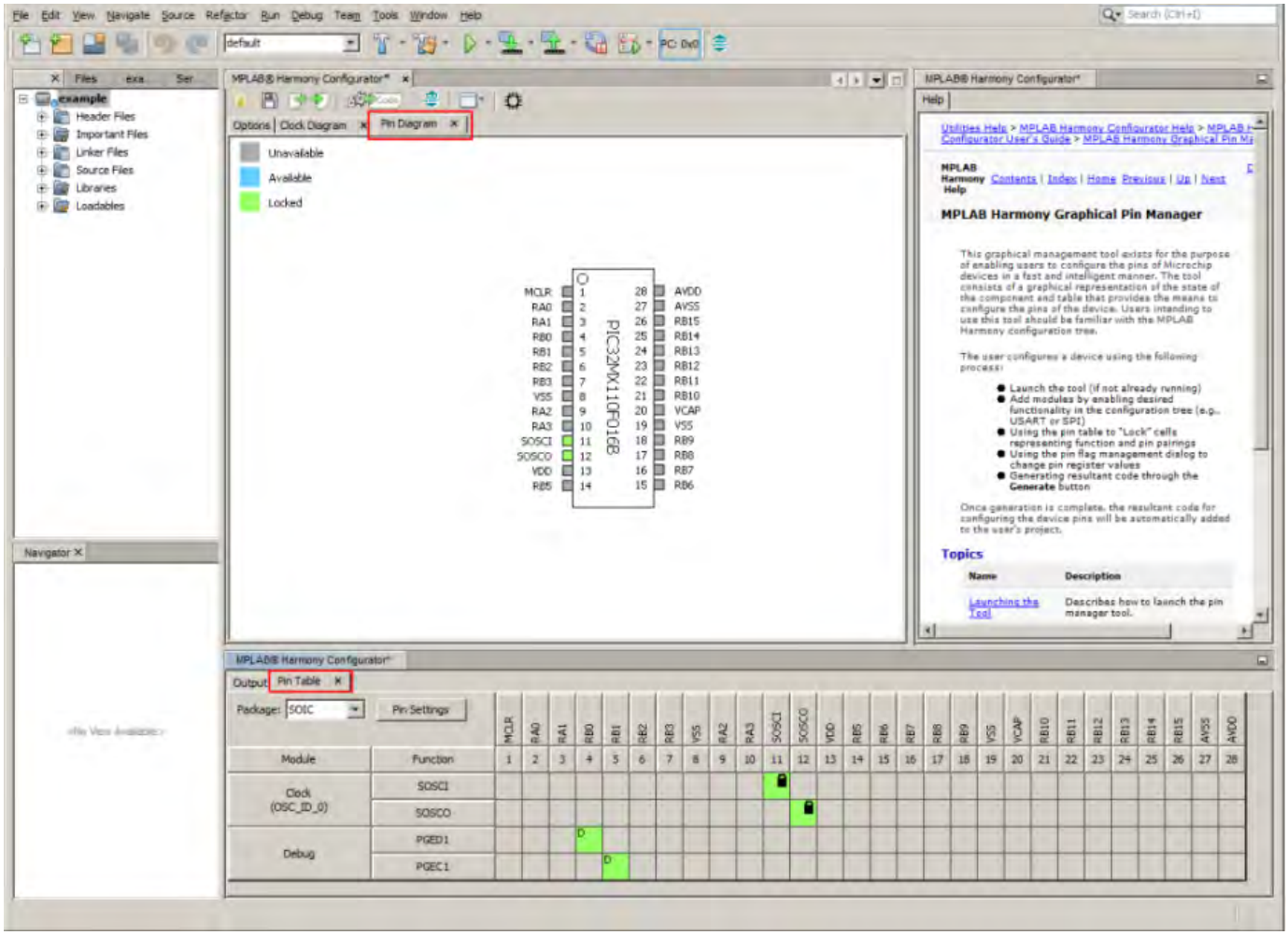
Once generation is complete, the resultant code for configuring the device pins will be automatically added to the user’s project.

Launching the Tool

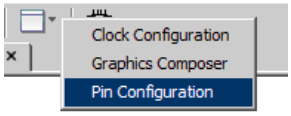
Describes how to launch the pin manager tool.

Description

The pin manager tool automatically launches when MHC starts.



The pin manager tool can be launched from the main window toolbar application launcher or from the option tree.

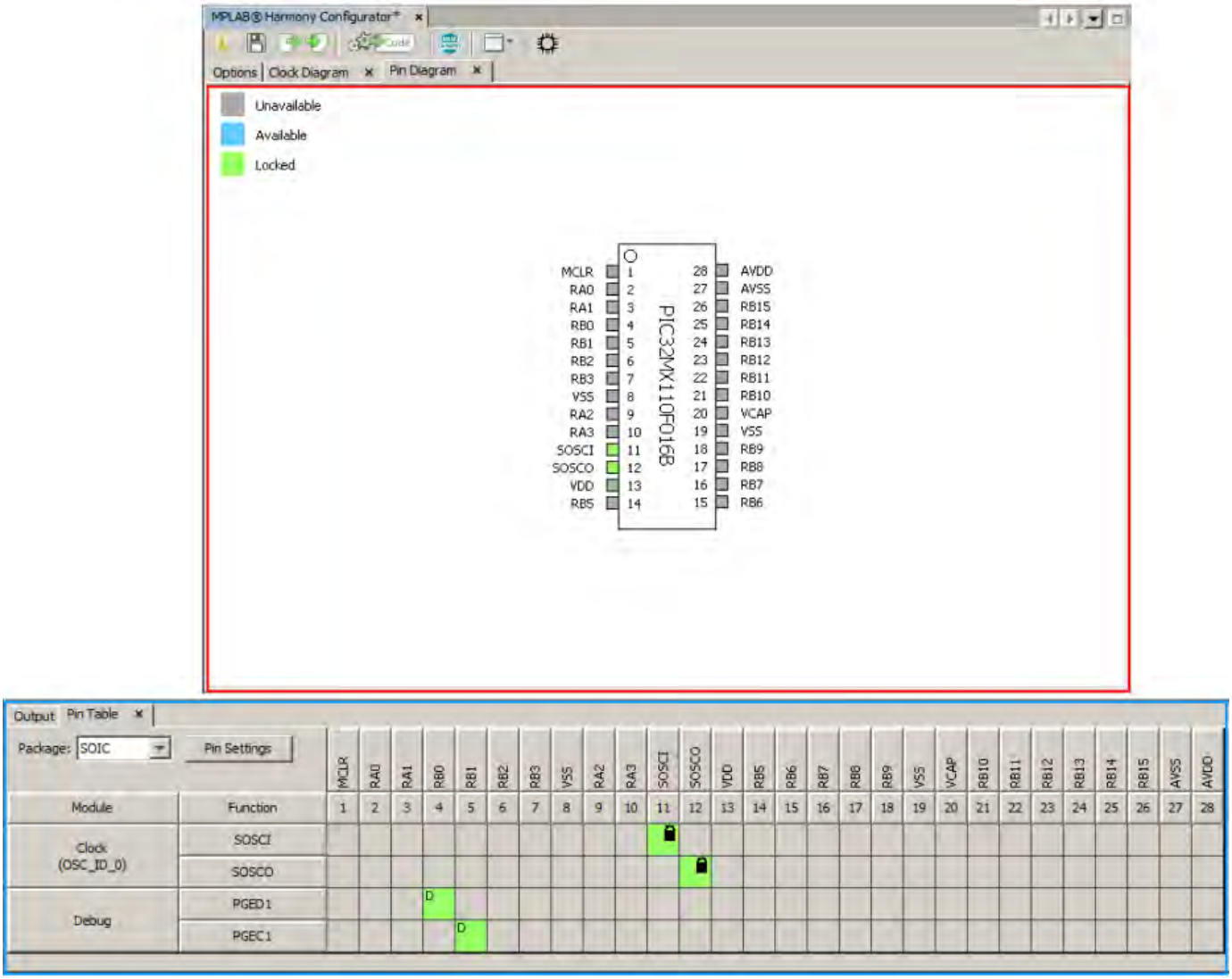


The pin manager tool can also be launched from the configuration tree.

Tool Tabs

The pin manager tool has two tabs:

- Pin Diagram (see the red section in the following figure)
- Pin Table (see the blue section in the following figure)



Pin Diagram Tab

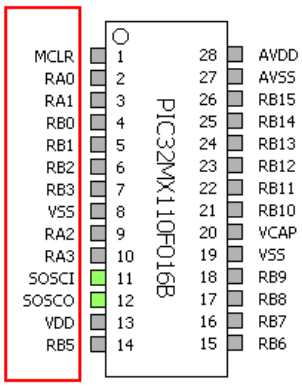
Describes the pin diagram features.

Description

This diagram is a graphical representation of the selected component to be configured. The diagram contains the following:

Pin Names

These are the base names of each pin. These names will change based on the selected function for this pin.

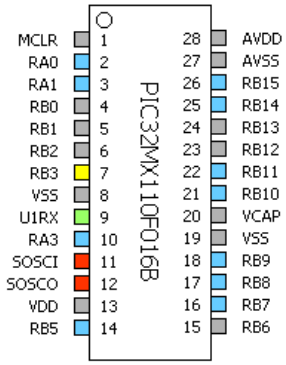


Pin States

This is a graphical indication of the state of the pin.

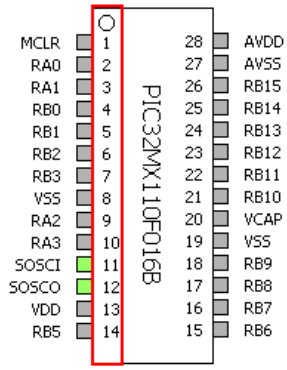
Pin States Legend:

Color	Icon	Description
Blue		This pin can be locked to an available function in the table.
Gray		This pin is currently unavailable based on the state of the pin table.
Green		This pin has been locked to a function.
Red		This pin has been automatically locked to a pin based on function priority.
Yellow		This pin is currently highlighted by the cursor.



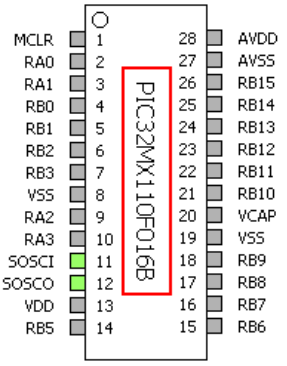
Pin Numbers

The number for each pin.



Component Name

The name of this component.



Pin Table Tab

Describes the pin table features.

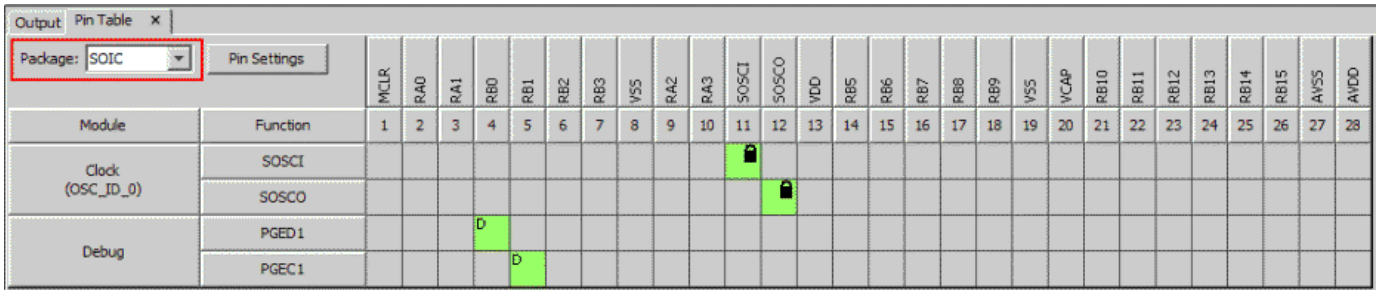
Description

The pin table allows the user to graphically configure the pins for the given component. The table contains the following areas of interest:

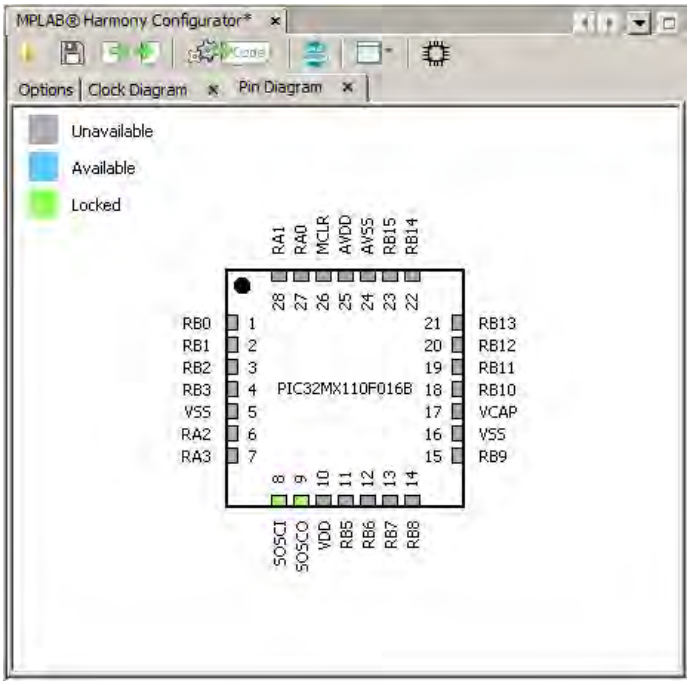
Package Selector

This menu contains the available packages for the selected component.

Note: Changing this value will reset the state of the pins to default.



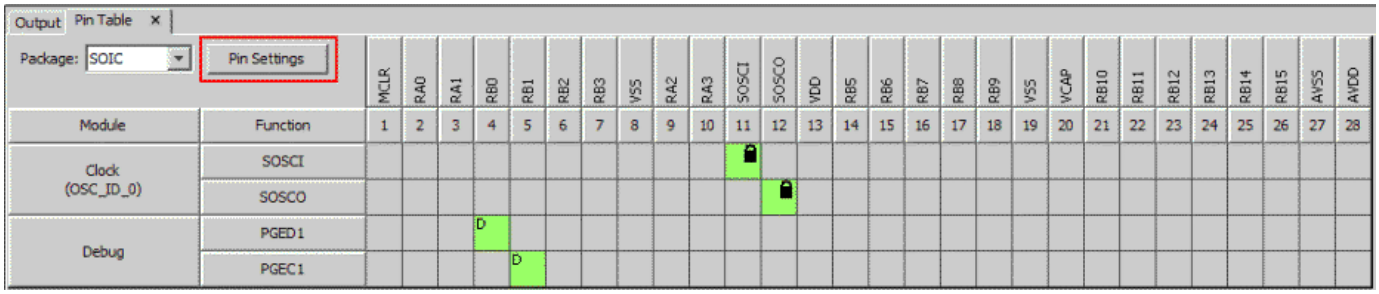
Observe the changes in the diagram and table when the QFN package is selected for this device.



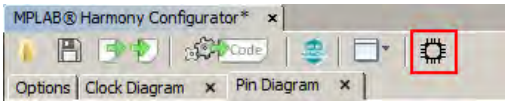
Pin Settings Button

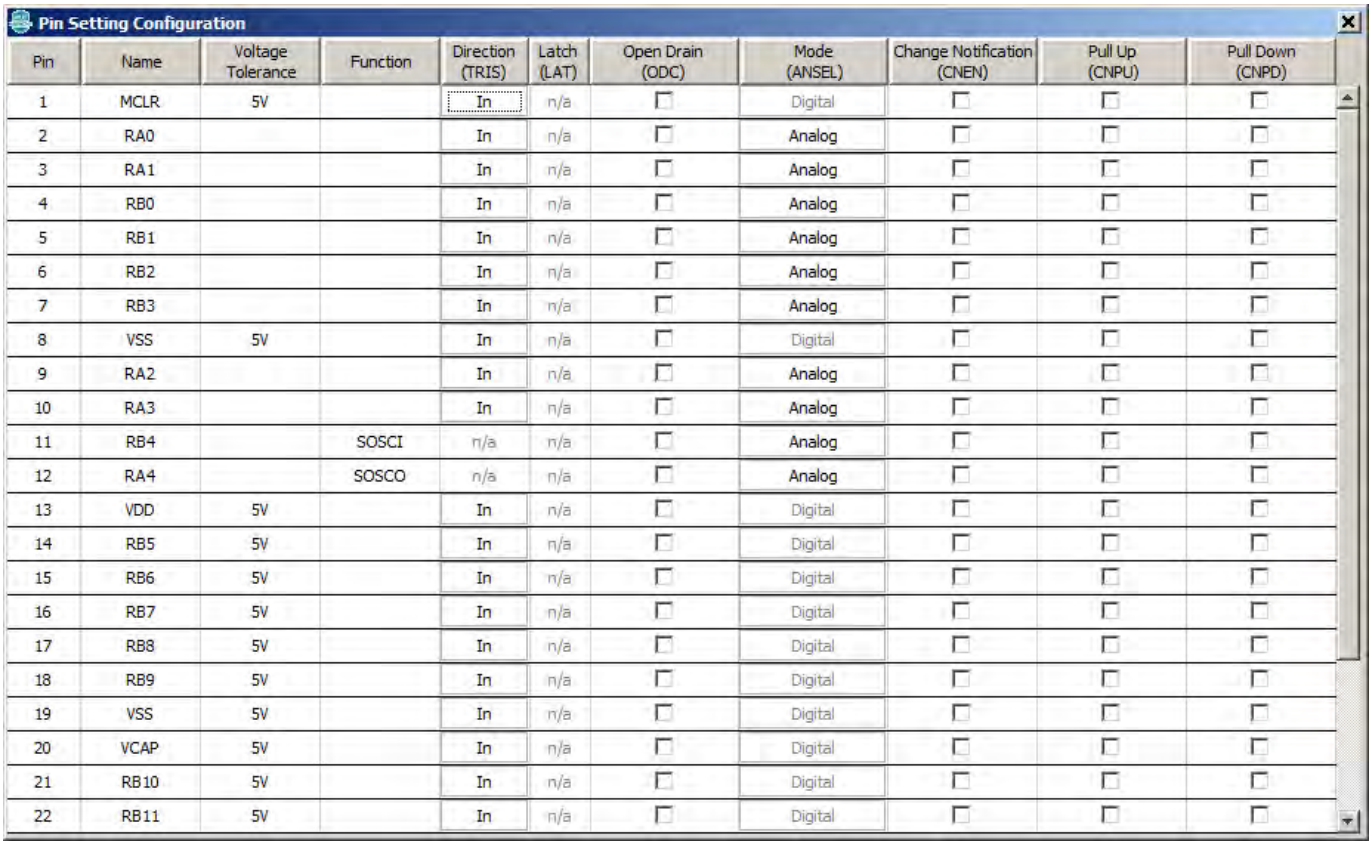
This button shows the pin settings configuration menu. This dialog allows for the configuration of pin direction, drain, mode, latch, change notification, and pull-up and pull-down options.

Note: The direction and mode options are dependent on the function that is assigned to the pin. Board Support Package functions may lock other options as well.



The pin settings dialog can also be launched from the main toolbar when the pin diagram is visible.

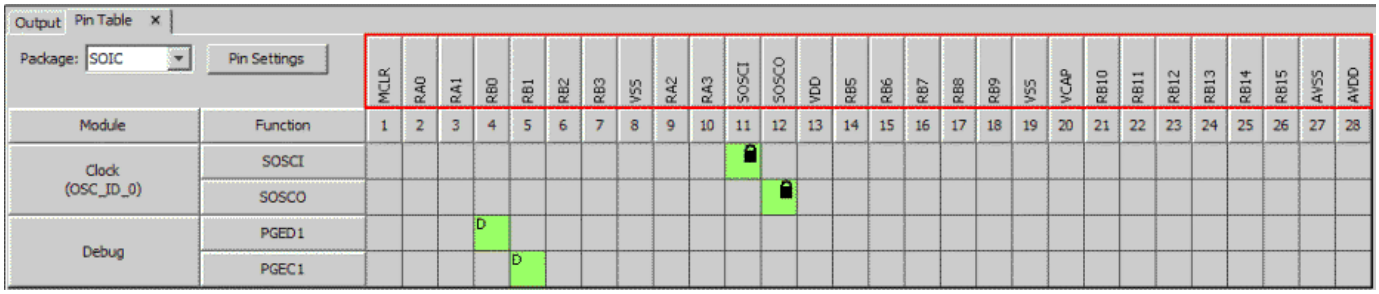




Pin	Name	Voltage Tolerance	Function	Direction (TRIS)	Latch (LAT)	Open Drain (ODC)	Mode (ANSEL)	Change Notification (CNEN)	Pull Up (CNPU)	Pull Down (CNPD)
1	MCLR	5V		In	n/a	☐	Digital	☐	☐	☐
2	RA0			In	n/a	☐	Analog	☐	☐	☐
3	RA1			In	n/a	☐	Analog	☐	☐	☐
4	RB0			In	n/a	☐	Analog	☐	☐	☐
5	RB1			In	n/a	☐	Analog	☐	☐	☐
6	RB2			In	n/a	☐	Analog	☐	☐	☐
7	RB3			In	n/a	☐	Analog	☐	☐	☐
8	VSS	5V		In	n/a	☐	Digital	☐	☐	☐
9	RA2			In	n/a	☐	Analog	☐	☐	☐
10	RA3			In	n/a	☐	Analog	☐	☐	☐
11	RB4		SOSCI	n/a	n/a	☐	Analog	☐	☐	☐
12	RA4		SOSCO	n/a	n/a	☐	Analog	☐	☐	☐
13	VDD	5V		In	n/a	☐	Digital	☐	☐	☐
14	RB5	5V		In	n/a	☐	Digital	☐	☐	☐
15	RB6	5V		In	n/a	☐	Digital	☐	☐	☐
16	RB7	5V		In	n/a	☐	Digital	☐	☐	☐
17	RB8	5V		In	n/a	☐	Digital	☐	☐	☐
18	RB9	5V		In	n/a	☐	Digital	☐	☐	☐
19	VSS	5V		In	n/a	☐	Digital	☐	☐	☐
20	VCAP	5V		In	n/a	☐	Digital	☐	☐	☐
21	RB10	5V		In	n/a	☐	Digital	☐	☐	☐
22	RB11	5V		In	n/a	☐	Digital	☐	☐	☐

Pin Names

This row indicates the currently selected function for each pin. If no function is selected, the default pin name is shown instead.



Pin Numbers

This row indicates the number of each pin in the table.

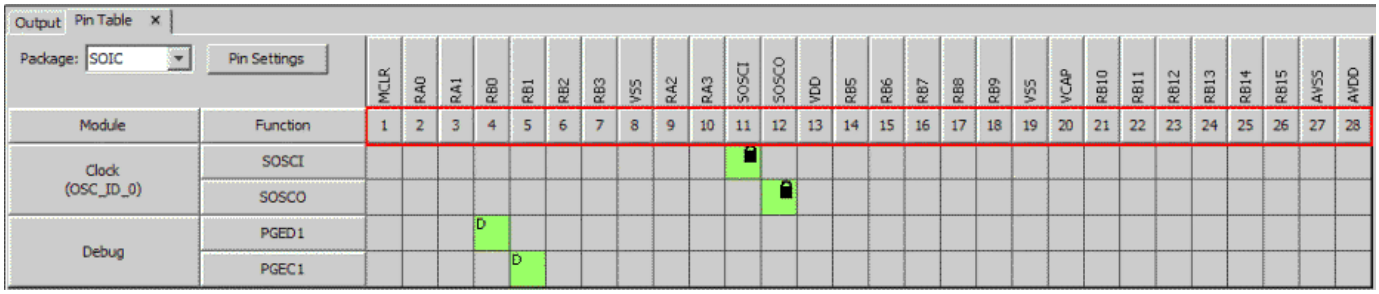


Table Modules

This column contains the modules, or groups of functions, for the current configuration. These modules are controlled by the MHC configuration tree.

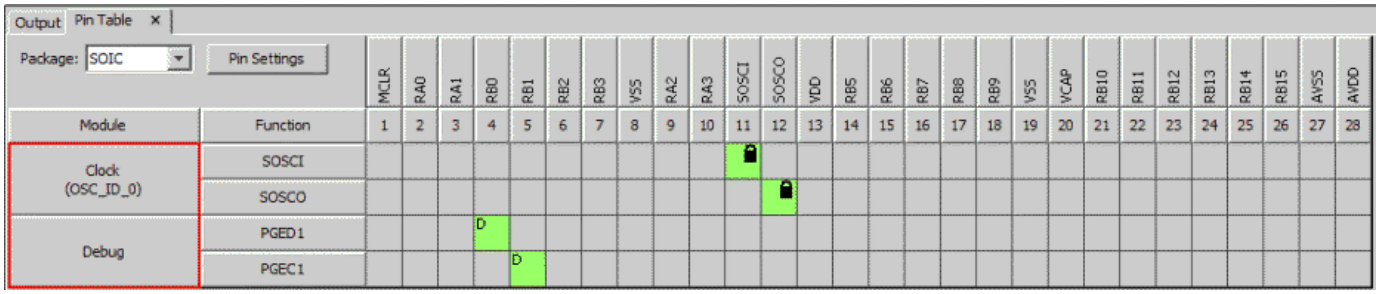


Table Functions

This column displays the functions that belong to each module.

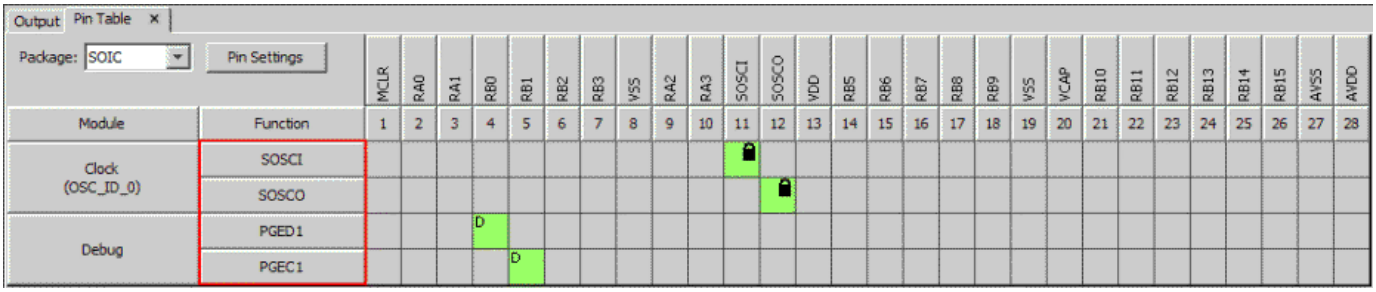


Table Grid

This area contains the grid cells. This area is for making connections between pins and functions.

Table Grid Cell Legend:

Icon	Description
	The cell is currently unavailable and cannot be selected.
	The cell is available for selection.
	The cell has been locked by the user.
D	The cell is a special debug indicator. This cell does not actually lock to a pin but is a visual debug reminder. This indicator means that the pin this cell resides on will be appropriated for debugging purposes based on the currently selected debug options.
	This cell has been automatically locked based on the available choices. This selection takes function priority into account. This lock cannot be changed by the user.

Module Management

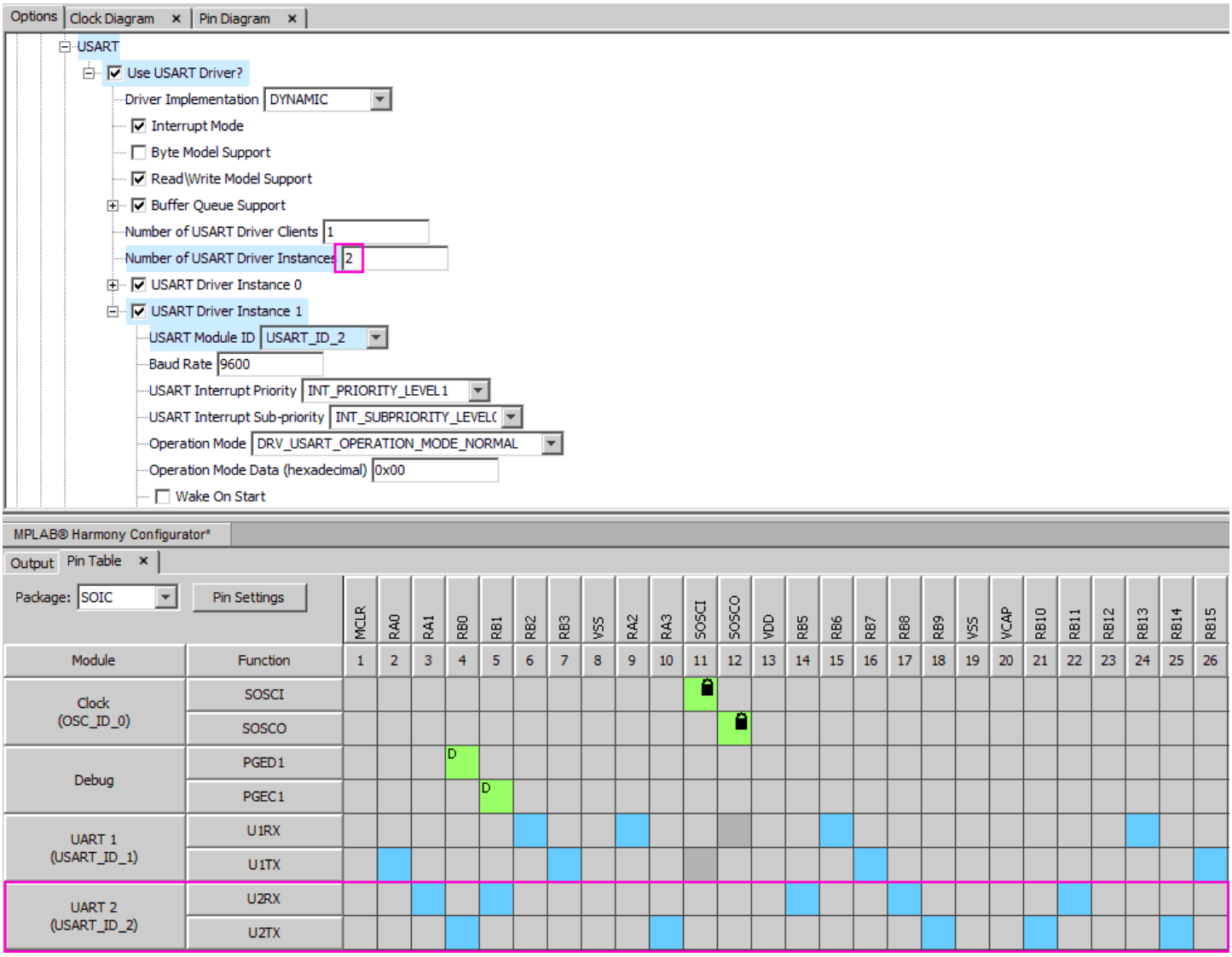
Describes the module management features.

Description

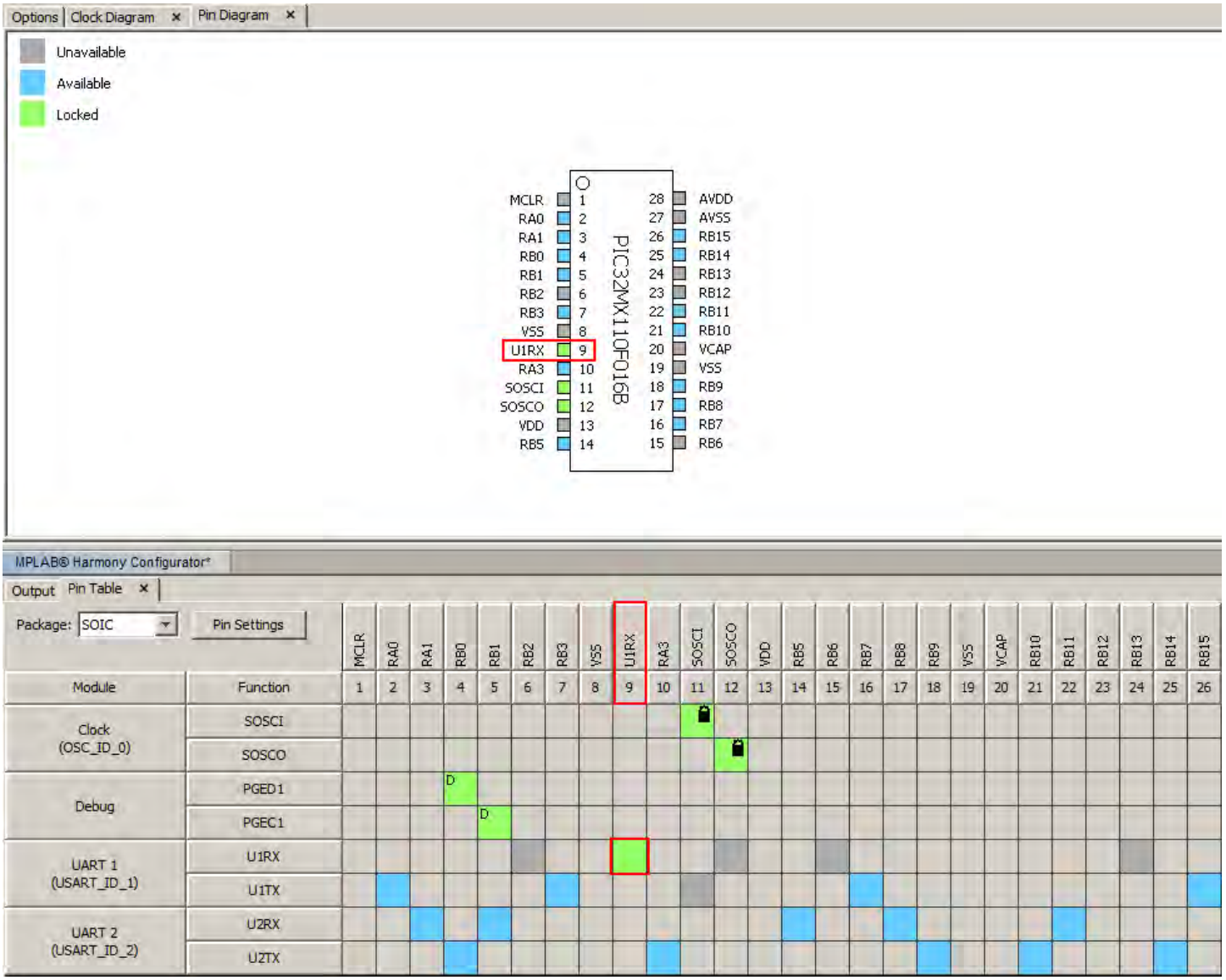
The Pin Manager table displays modules based on selections made in the configuration tree.

Observe that by enabling the USART driver instance that the USART1 module appears in the pin table.

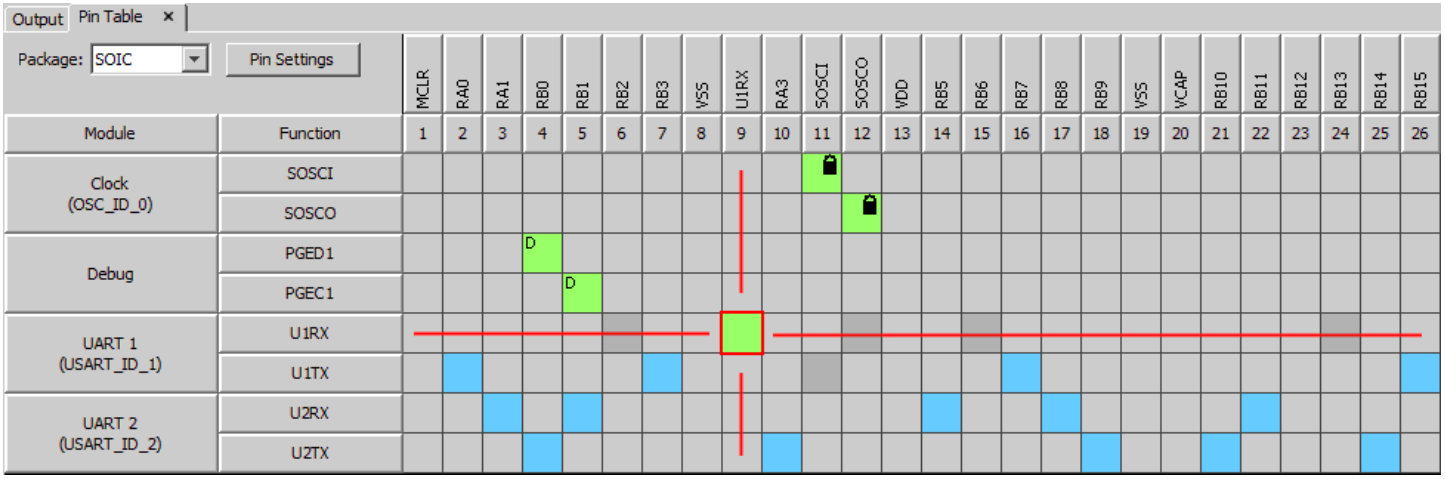
Now increase the number of USART driver instances to 2. Once the second USART instance is set to USART_ID_2, the table will display the second USART module.



The U1RX, U1TX, U2RX, and U2TX functions are Peripheral Pin Select functions and can be assigned to multiple pins. Blue cells indicate a potential pin-to-function lock. Observe that left-clicking the blue cell corresponding to pin 9 and U1RX locks that cell to that pin/function pair. U1RX is now assigned to pin 9. Observe also that the name above pin 9 has changed to indicate the locked function, as well as the name of pin 9 in the pin diagram.



With pin 9 locked, the other options for pin 9 and U1RX are now marked unavailable.



The green cell can be left-clicked again to unlock the pin and function.

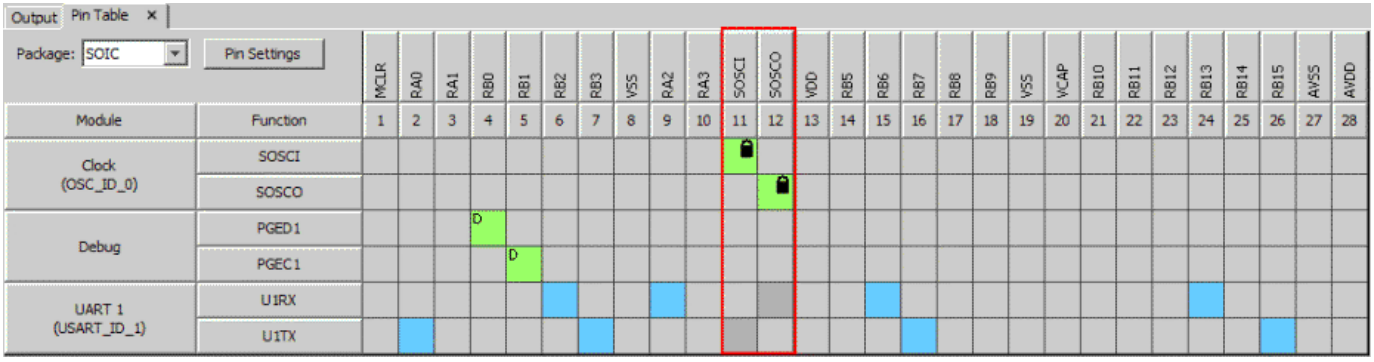
Conflict Resolution

Describes conflict resolution features.

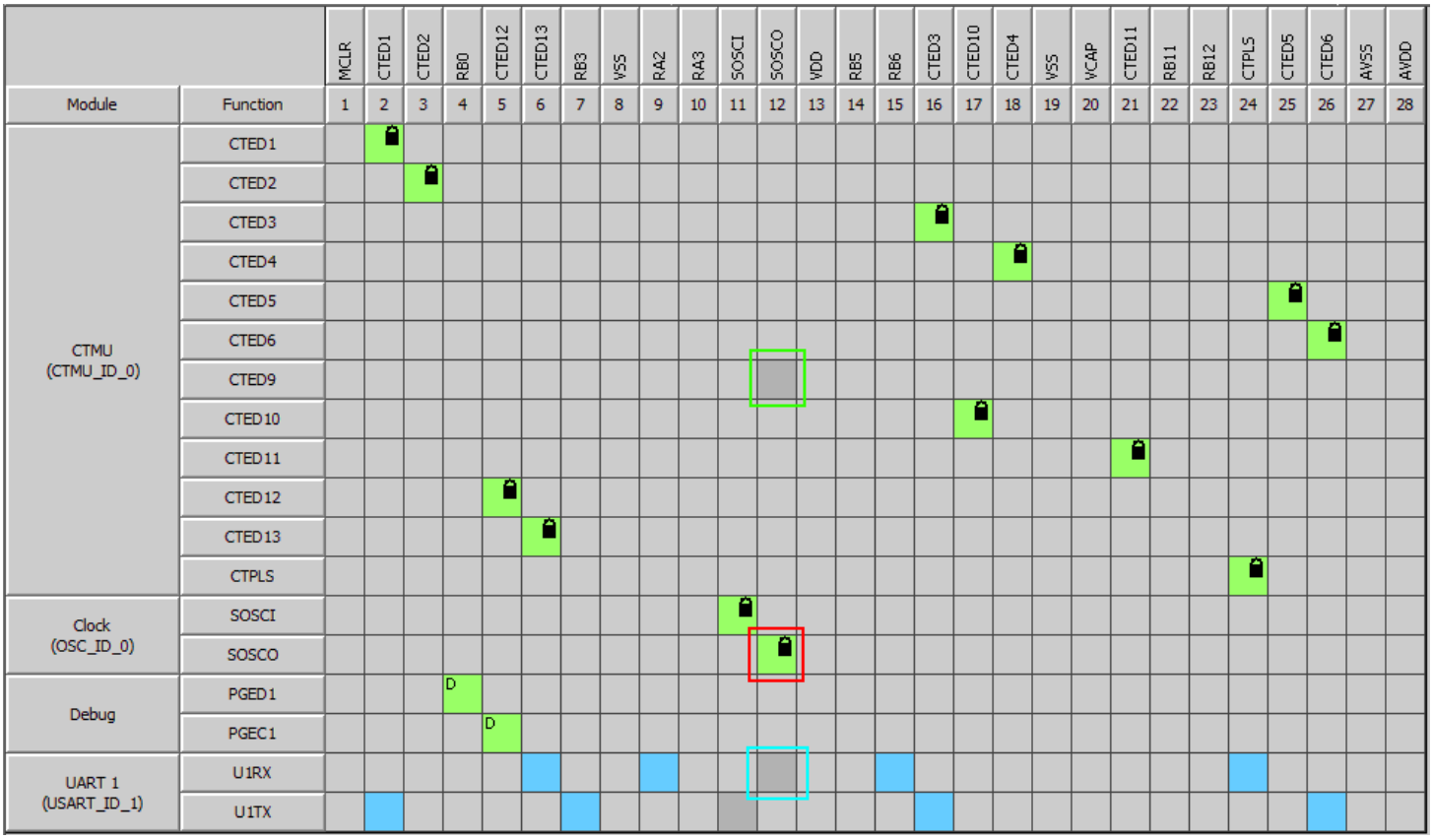
Description

The Pin Manager uses automatic conflict resolution to determine the proper function when multiple options are available.

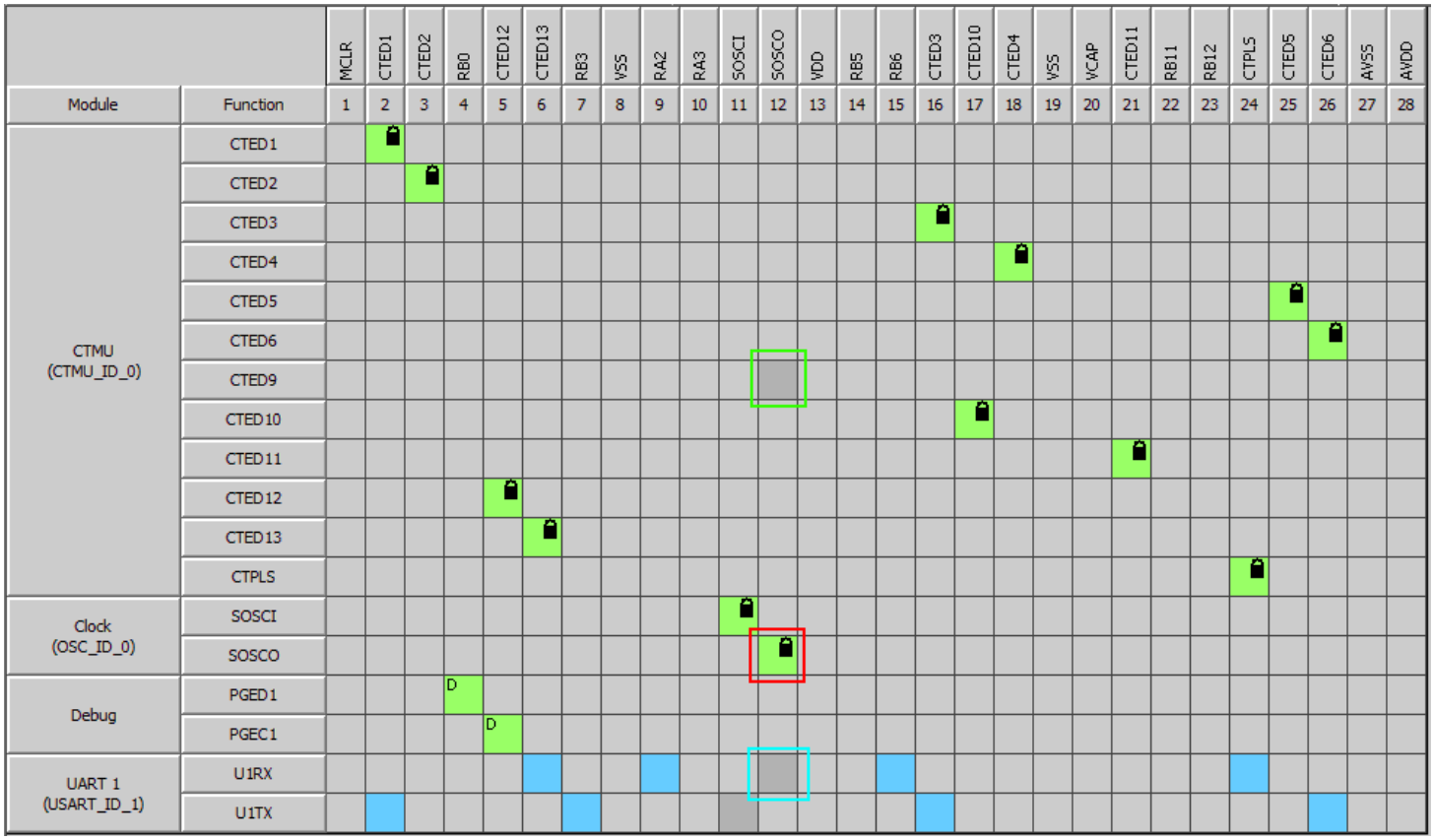
Consider the available functions for pin 12: SOSCO/RPA4/T1CK/CTED9/PMA1/RA4. Observe that the SOSCO function was given automatic priority over RPA4 (U1RX).



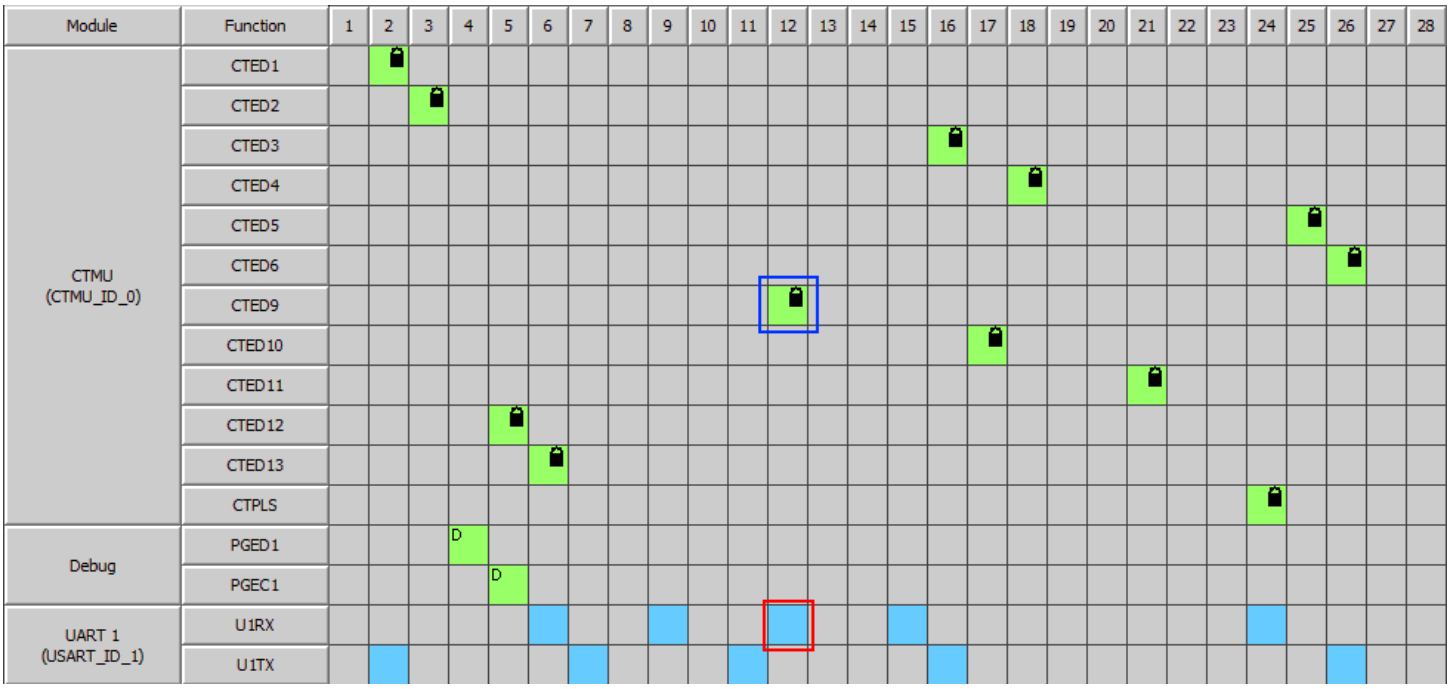
The output window displays a detailed message of this event.



Observe also that with the addition of another lower priority function that the selection does not change. The higher priority function SOSCO (red) is still automatically selected while lower priority functions RPA4 (PPS) and OC1 are disabled.



If the highest priority is a Peripheral Pin Select function (red highlight) a choice is given to the user. The next lowest priority function is automatically selected (blue highlight), but this can be overridden by user action.



If the Peripheral Pin Select function (red highlight) is manually selected then the automatic choice (blue highlight) is overridden. A conflict is still reported. If the Peripheral Pin Select function is unlocked then the lower priority function will be automatically locked again.

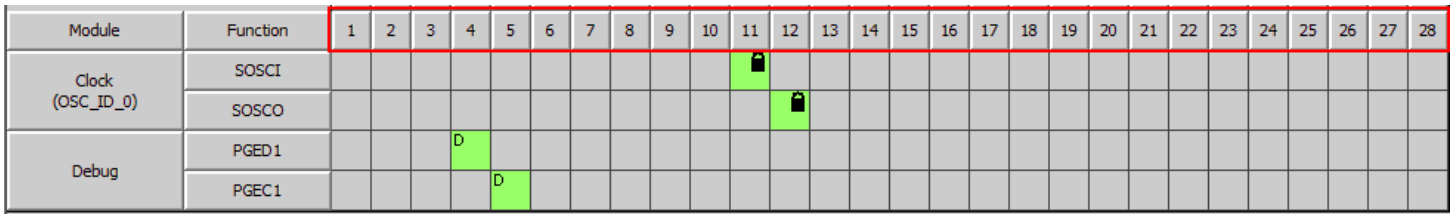
Pin Table Features

Describes pin table features.

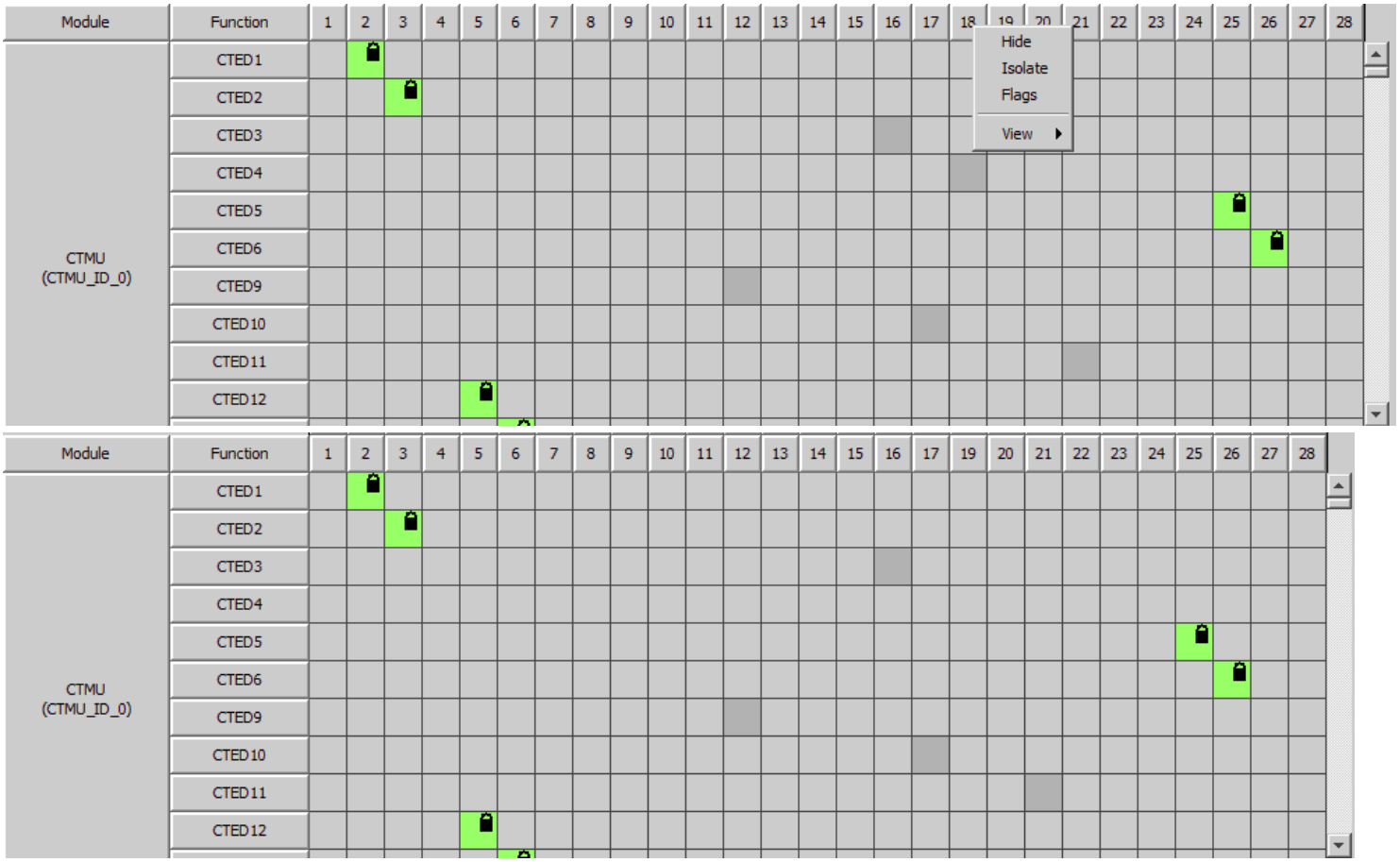
Description

The Pin Table can be reconfigured to show as little or as much information as the user desires. For example, individual pin rows can be hidden or

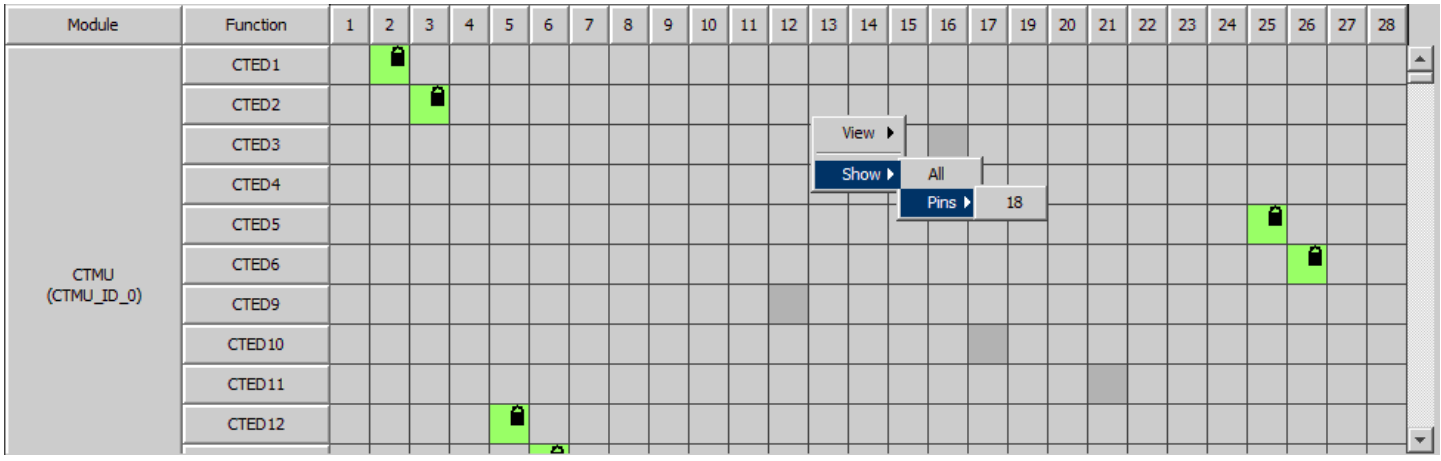
isolated depending on how much information is desired. This is accomplished by right-clicking on a pin number and selecting a desired option from the context menu.



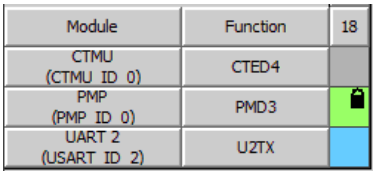
To remove pin 18 from the table, right-click the pin 18 number box. Select **Hide** from the context menu.



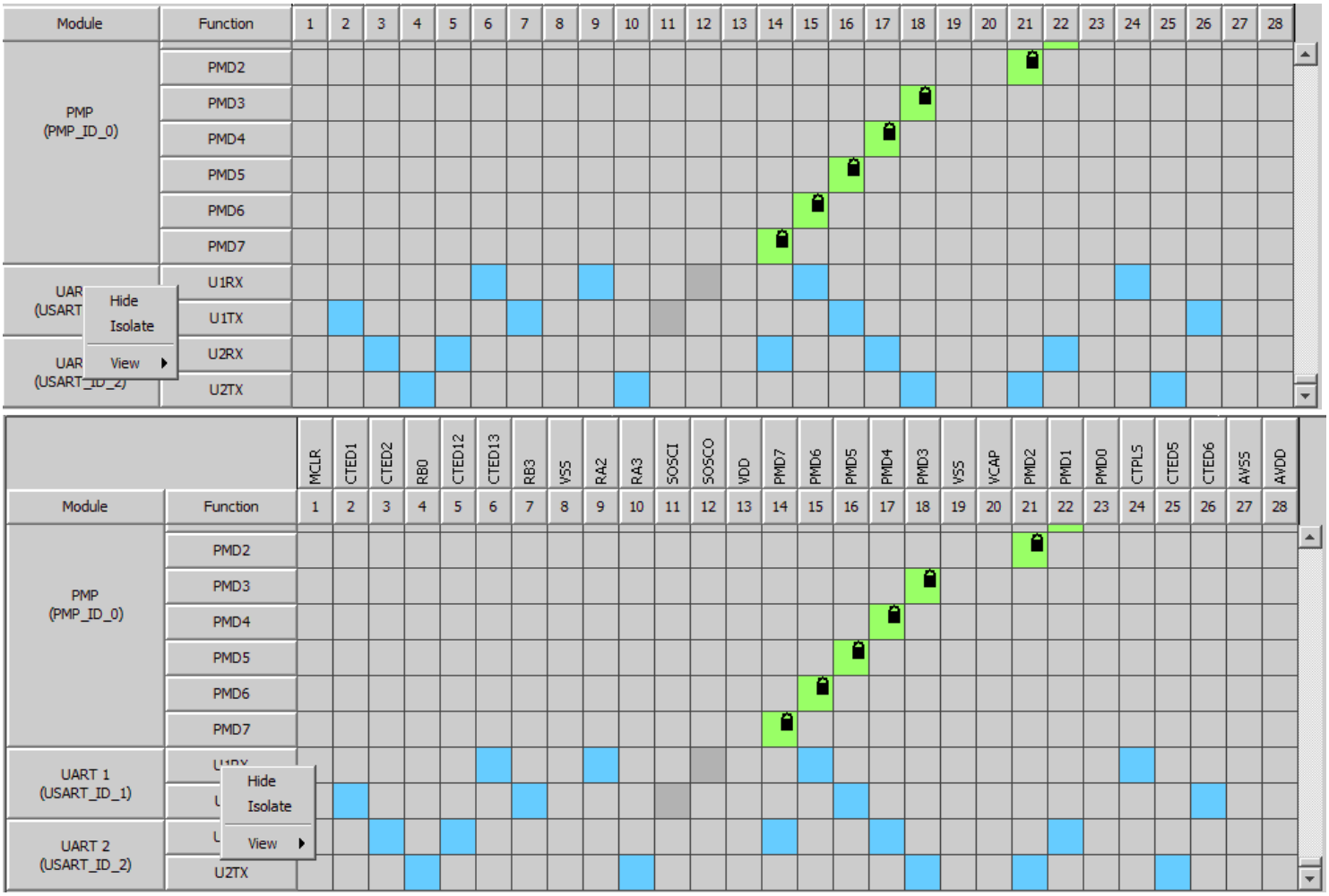
Observe that pin 18 has been removed from the table. To restore the column, right-click in the table and select **Show > All** or navigate the available sub-menus and select pin 18.



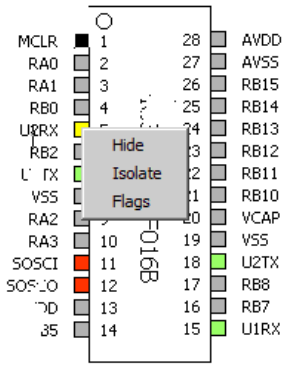
The table can also be reduced to show only desired pins and functions by using the "Isolate" command. To show only pin 18, again right-click on the pin 18 number box and select **Isolate**.



This functionality also exists for pin modules, functions, and ports.



The table can also be modified by right-clicking the pin boxes in the pin diagram.



The table can also be reconfigured to display pins according to their respective ports. To do this, right-click the table, navigate to the View sub-menu, and select **Ports**. The top row is the original pin number, the middle row shows the port grouping, and the bottom row is the pin's number inside the port grouping. Ports can also be hidden and isolated in the same manner as pins, modules, and functions. This is accomplished by right-clicking on the port name box.

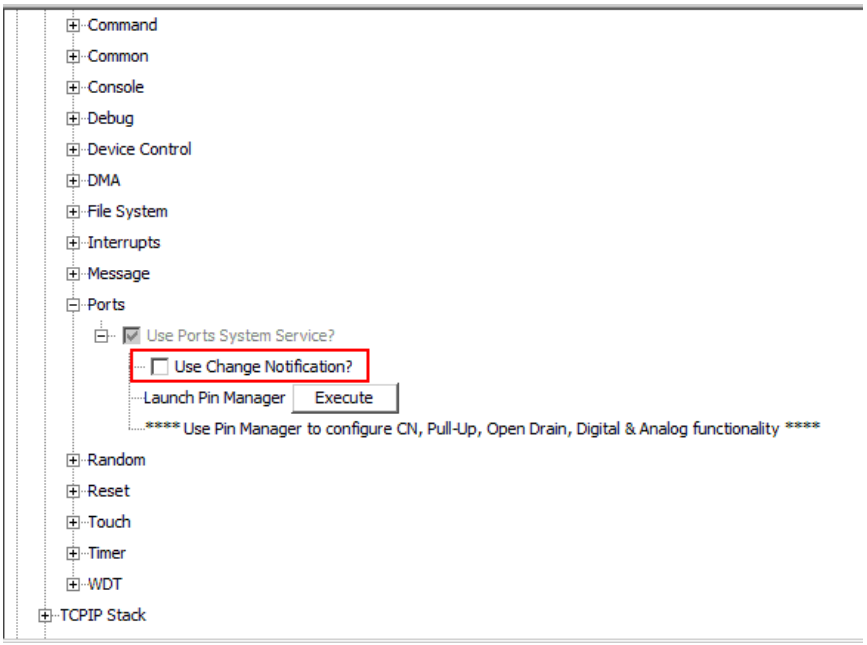
		2	3	9	10	12	4	5	6	7	11	14	15	16	17	18	21	22	23	24	25	26
		A					B															
Module	Function	0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Clock (OSC_ID_0)	SOSCI																					
	SOSCO																					
Debug	PGED1						D															
	PGEC1							D														
UART 1 (USART_ID_1)	U1RX																					
	U1TX																					
UART 2 (USART_ID_2)	U2RX																					
	U2TX																					

Change Notification and Non-PPS Devices

Describes handling change notification for non-PPS devices.

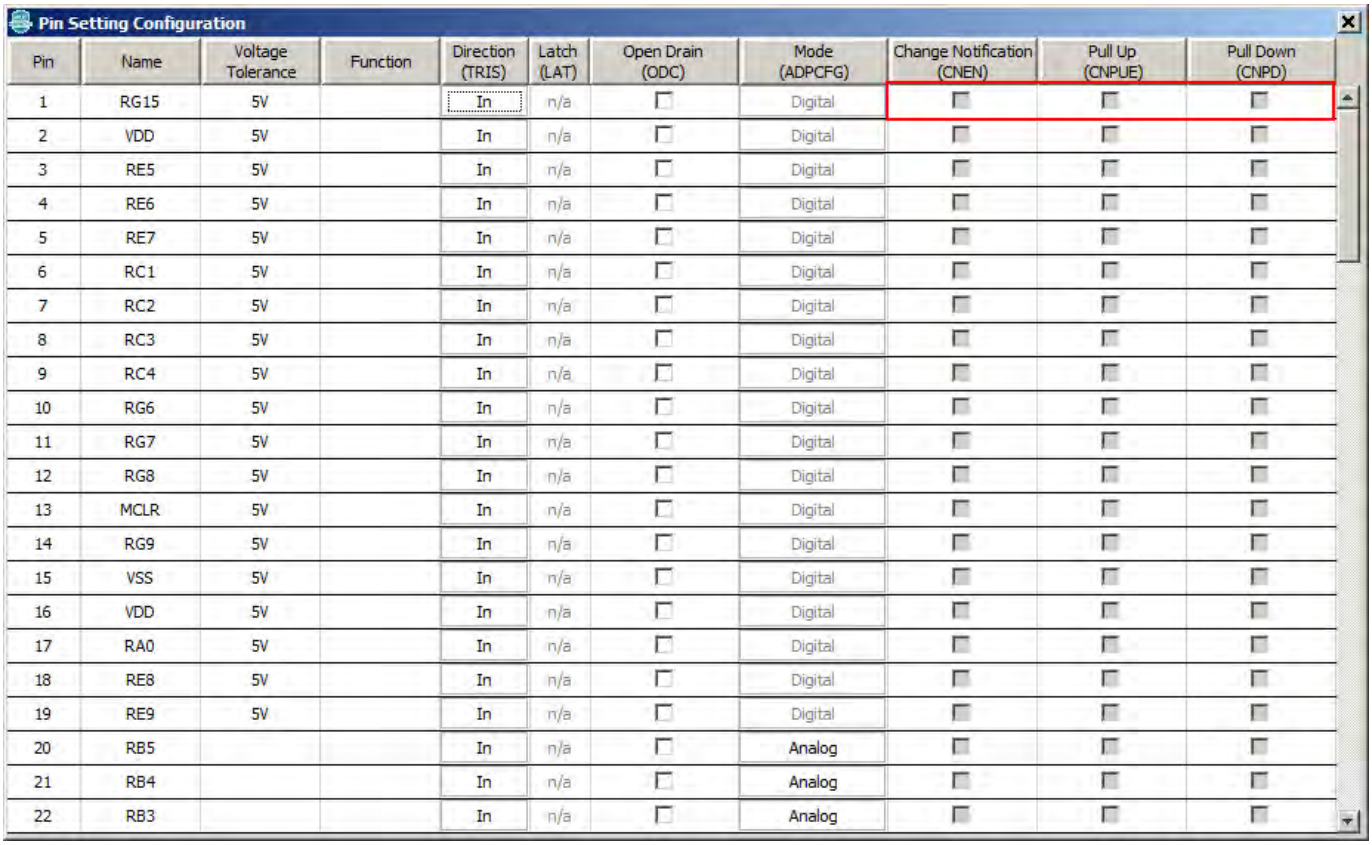
Description

For non PPS parts, change notifications behave differently. They must be explicitly enabled in the configuration tree.



When enabled, the Change Notification module appears in the table. Change notification cells behave similarly to Peripheral Pin Select functions. They will be overridden by higher priority functions, but will provide a user choice if they are the highest priority.

The pin flag dialog also behaves differently for Non-PPS parts. The "Change Notification", "Pull Up", and "Pull Down" options are disabled.



Pin	Name	Voltage Tolerance	Function	Direction (TRIS)	Latch (LAT)	Open Drain (ODC)	Mode (ADPCFG)	Change Notification (CNEN)	Pull Up (CNPUE)	Pull Down (CNPD)
1	RG15	5V		In	n/a		Digital			
2	VDD	5V		In	n/a		Digital			
3	RE5	5V		In	n/a		Digital			
4	RE6	5V		In	n/a		Digital			
5	RE7	5V		In	n/a		Digital			
6	RC1	5V		In	n/a		Digital			
7	RC2	5V		In	n/a		Digital			
8	RC3	5V		In	n/a		Digital			
9	RC4	5V		In	n/a		Digital			
10	RG6	5V		In	n/a		Digital			
11	RG7	5V		In	n/a		Digital			
12	RG8	5V		In	n/a		Digital			
13	MCLR	5V		In	n/a		Digital			
14	RG9	5V		In	n/a		Digital			
15	VSS	5V		In	n/a		Digital			
16	VDD	5V		In	n/a		Digital			
17	RA0	5V		In	n/a		Digital			
18	RE8	5V		In	n/a		Digital			
19	RE9	5V		In	n/a		Digital			
20	RB5			In	n/a		Analog			
21	RB4			In	n/a		Analog			
22	RB3			In	n/a		Analog			

Exporting Pin Mapping

Provides information on exporting pin mappings.

Description

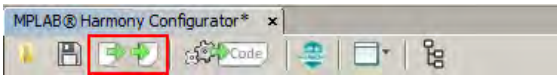
The MPLAB Harmony Graphical Pin Manager provides the ability to export the pin mapping of the current configuration into Excel in `.xls` format for the purpose of printing out the pin mapping. Refer to Importing and Exporting Data for the steps to export the pin mapping.

Importing and Exporting Data

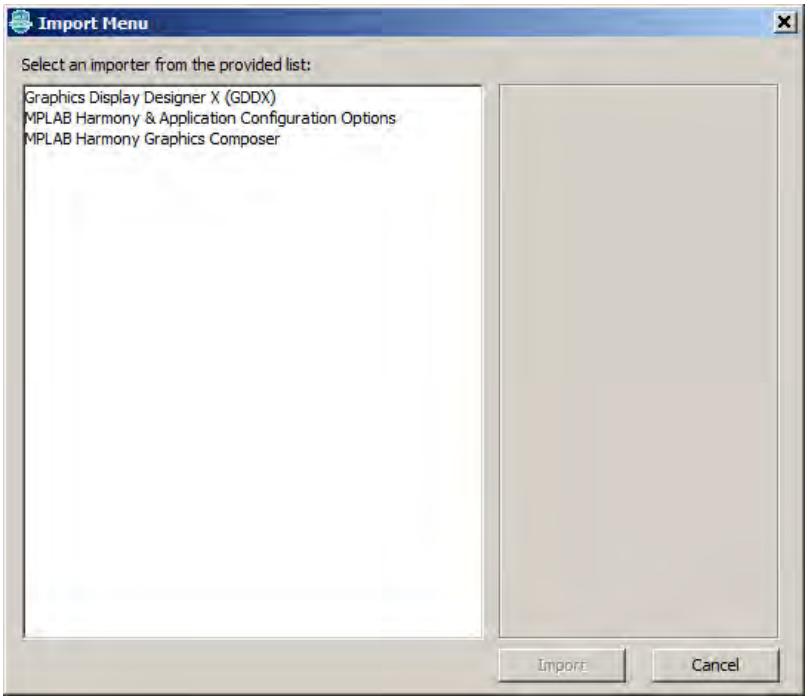
Provides information on importing and exporting data to/from the MHC.

Description

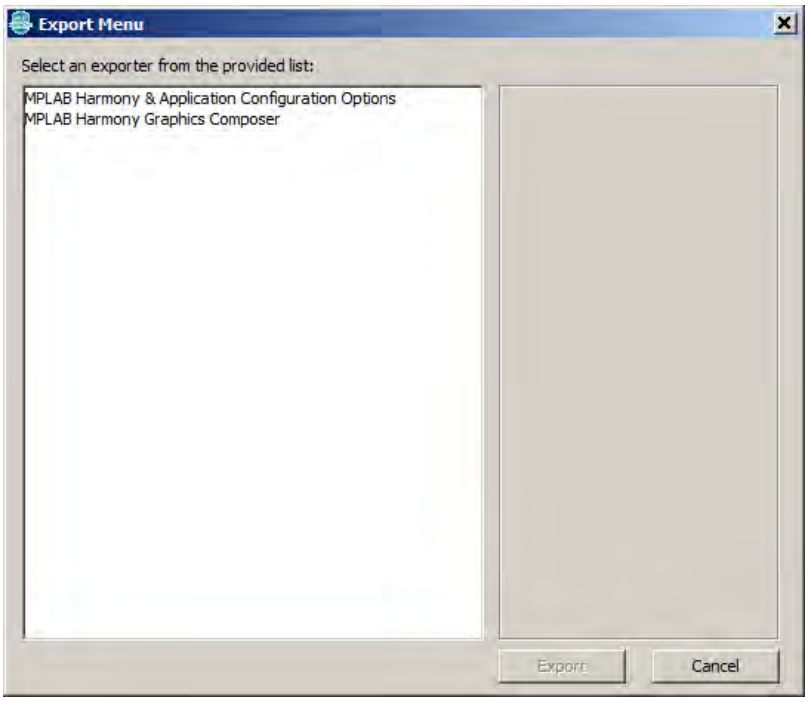
The MPLAB Harmony Configurator provides several options for importing and export various types of data to and from the application. The import and export icons can be found in the main window toolbar.



The Import dialog shows the various data sources that can be imported into MPLAB Harmony Configurator. To import, select an item from the list and click **Import**.



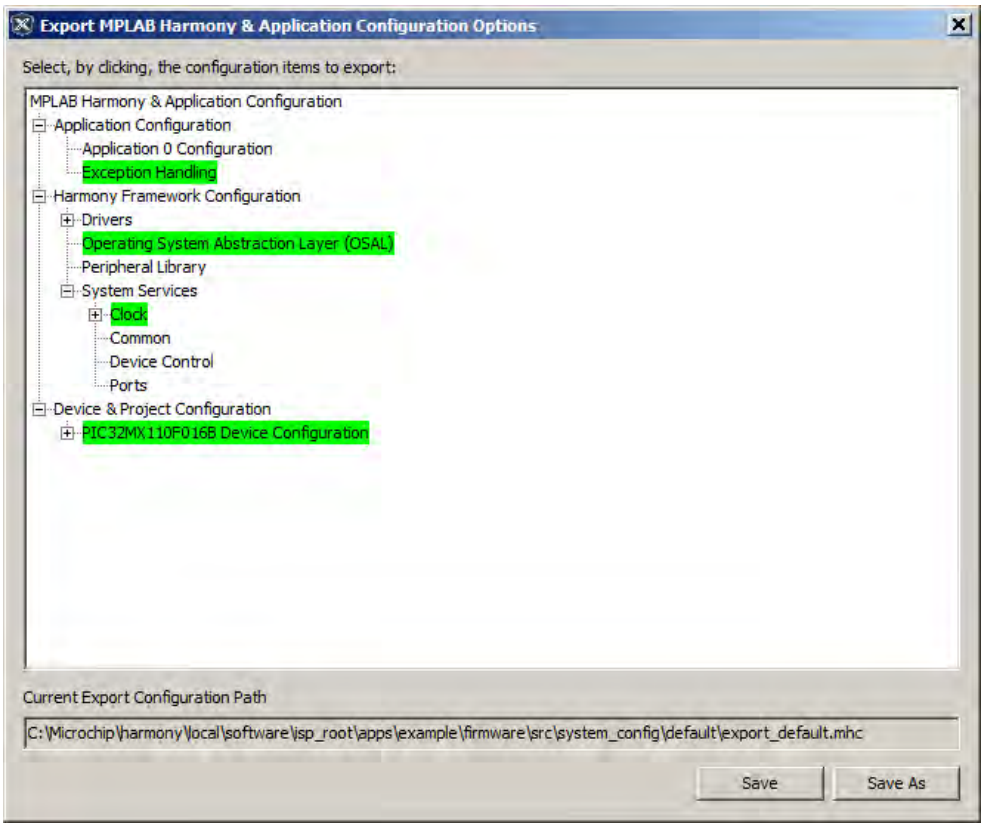
The Export dialog shows the various data sources that can be exported from MPLAB Harmony Configurator. To export, select an item from the list and click **Export**.



Importing and Exporting MPLAB Harmony Configurator Configuration Options

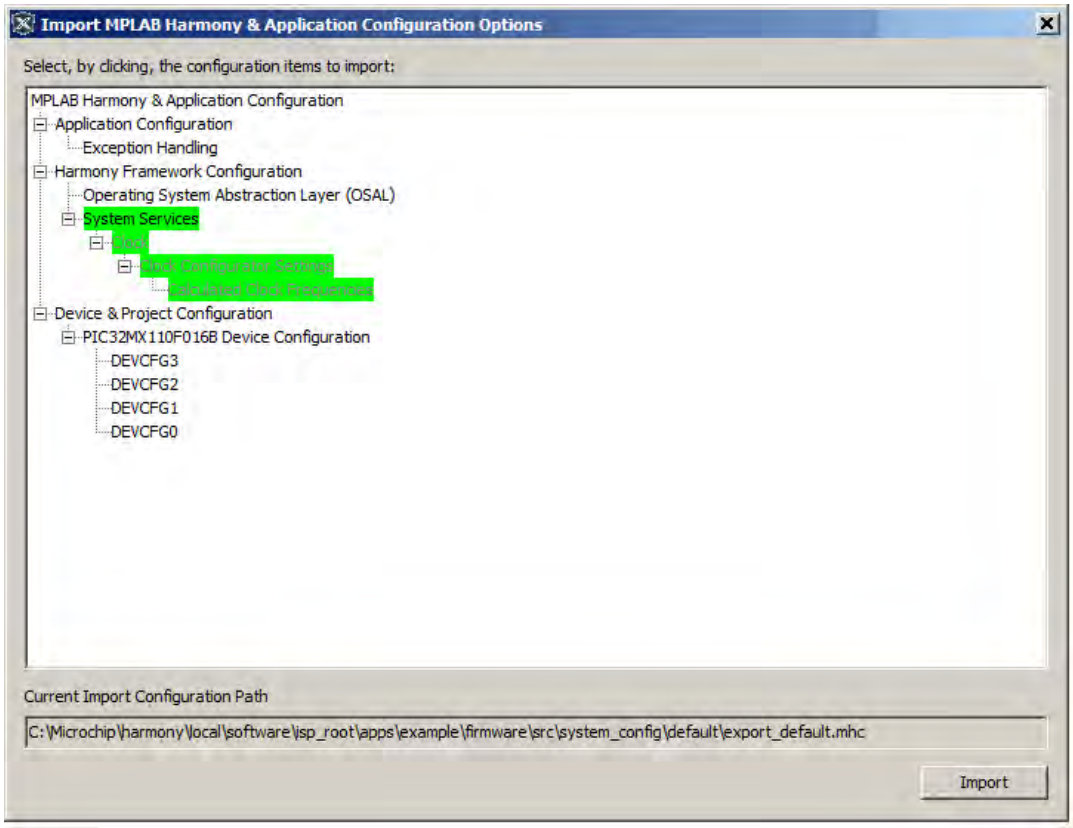
By selecting **MPLAB Harmony & Application Configuration Options** from either the Import or Export dialog, the user has the ability to create or import `.mhc` files with only user-selected options.

The following figure provides an example of the option export dialog.



To use this feature, left-click any desired option to toggle its state. Green-highlighted options will be exported. Then, use the **Save** and **Save As** buttons as desired to write the file.

To import, select the option import from the Import dialog and select the previously exported file. Observe that only the exported options are visible in the import window. The user can again select and highlight items in green to select them for import. When all desired settings have been highlighted, click **Import**.


Import MPLAB Harmony & Application Configuration Options
Select, by clicking, the configuration items to import:
MPLAB Harmony & Application Configuration
Application Configuration
Exception Handling
Harmony Framework Configuration
Operating System Abstraction Layer (OSAL)
System Services
Device & Project Configuration
PIC32MX110F016B Device Configuration
DEVCFG3
DEVCFG2
DEVCFG1
DEVCFG0
Current Import Configuration Path
C:\Microchip\harmony\local\software\sp_root\apps\example\firmware\src\system_config\default\export_default.mhc
Import

Index