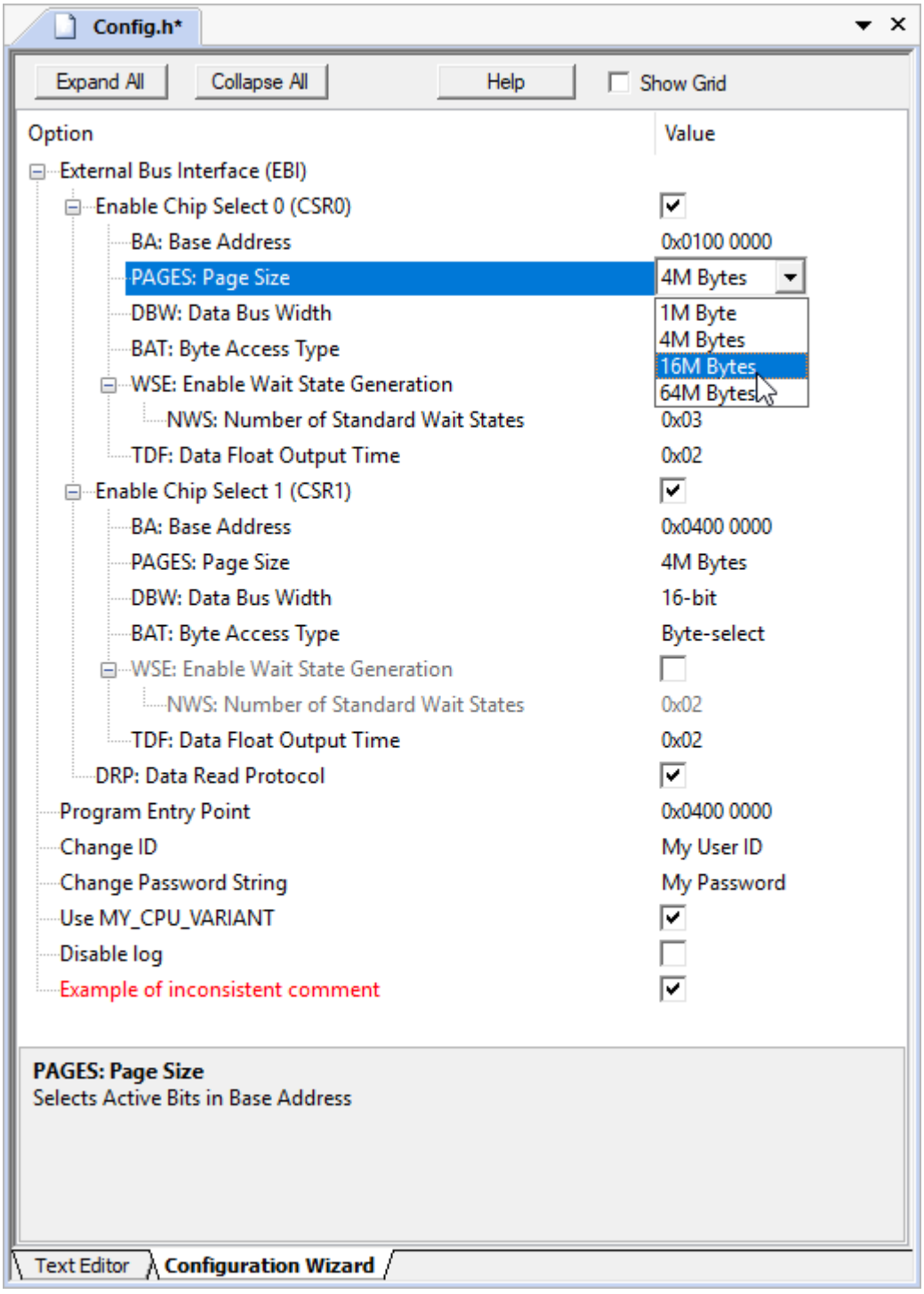
Configuration Wizard

The Configuration Wizard is a utility integrated in the editor for generating GUI-like configuration controls in assembler, C/C++, or initialization files. Configuration Wizard Annotations generate the controls.
This picture is the result of the example source code further down.



Where

Option	Are the device properties, which can be represented in a tree structure. Each item can have an explanatory tool-tip.
Value	Sets the option value. Can contain controls to encapsulate data in predefined drop-down lists.

Click the tab Configuration Wizard and set device options via graphical controls. Click the tab Text Editor to view or change the source code.

Example

The example code below generates the Configuration Wizard represented above and could be part of an initialization file:

```
//*** <<< Use Configuration Wizard in Context Menu >>> ***


FUNC void Setup (void) {

// <h> External Bus Interface (EBI)

//   <e1.13> Enable Chip Select 0 (CSR0)
//     <o1.20..31> BA: Base Address <0x0-0xFFF00000:0x100000><#/0x100000>
//                 <i> Start Address for Chip Select Signal
//     <o1.7..8>   PAGES: Page Size      <0=> 1M Byte    <1=> 4M Bytes
//                                       <2=> 16M Bytes  <3=> 64M Bytes
//                 <i> Selects Active Bits in Base Address
//     <o1.0..1>   DBW: Data Bus Width   <1=> 16-bit     <2=> 8-bit
//     <o1.12>     BAT: Byte Access Type <0=> Byte-write
//                                       <1=> Byte-select
//     <e1.5>      WSE: Enable Wait State Generation
//       <o1.2..4>   NWS: Number of Standard Wait States <1-8><#-1>
//     </e>
//     <o1.9..11>  TDF: Data Float Output Time <0-7>
//                 <i> Number of Cycles Added after the Transfer
//   </e>
  _WDWORD(0xFFE00000, 0x010024A9);   // EBI_CSR0: Flash

//   <e1.13> Enable Chip Select 1 (CSR1)
//     <o1.20..31> BA: Base Address <0x0-0xFFF00000:0x100000><#/0x100000>
```

```
//               <i> Start Address for Chip Select Signal
//     <o1.7..8>   PAGES: Page Size      <0=> 1M Byte    <1=> 4M Bytes
//                                       <2=> 16M Bytes  <3=> 64M Bytes
//               <i> Selects Active Bits in Base Address
//     <o1.0..1>   DBW: Data Bus Width   <1=> 16-bit     <2=> 8-bit
//     <o1.12>     BAT: Byte Access Type <0=> Byte-write
//                                       <1=> Byte-select
//     <e1.5>      WSE: Enable Wait State Generation
//       <o1.2..4>   NWS: Number of Standard Wait States <1-8><#-1>
//     </e>
//     <o1.9..11>  TDF: Data Float Output Time <0-7>
//               <i> Number of Cycles Added after the Transfer
//   </e>
  _WDWORD(0xFFE00004, 0x040034A5);   // EBI_CSR1: RAM

//   <q1.4>       DRP: Data Read Protocol
//                    <0=> Standard Read
//                    <1=> Early Read
  _WDWORD(0xFFE00024, 0x00000010);   // EBI_MCR:  Data Read Protocol

  _WDWORD(0xFFE00020, 0x00000001);   // EBI_RCR:  Remap Command

// </h>

// <o> Program Entry Point
  PC = 0x04000000;

}

// <s> Change ID
// <s1.30> Change Password String
#define ID  "My User ID"
char pw[] = "My Password";


/*********************************************************/
/*  Example for enabling and disabling code              */

// <c1> Use MY_CPU_VARIANT
// <i> Use MY_CPU_VARIANT, and set the include file
```

```
#define MY_CPU_VARIANT
#include "MyCpuVariant.h"
// </c>

// <!c1> Disable log
// <i> Disable log file generation
#define _USE_LOG
// </c>

// <c1>  Example of inconsistent comment
// <i> a mix of commented and uncommented lines in the block create an
inconsistency

// This type of comment, mixed with uncommented lines, creates the inconsistency
/* You can use this type of comment without creating an inconsistency */

   do_whatever;             // adding this type of comment here is allowed

// </c>


//*** <<< end of configuration section >>>    ***
```

```
// <<< Use Configuration Wizard in Context Menu >>>

/*可展開一個標題*/
// <h> 棧大小
//   <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
// </h>

#define Stack_Size 0x00008000

// <c1>Enable HAL Driver Component
//  <i>Enable HAL Driver Component
//#define ENABLE_HAL_DRIVER_SUPPORT
// </c>

/*e 展開項帶單選框覺得後面的宏值,o 進一步提供下拉複選框，所以 e 提供了一個框架,框架下的值可以與宏值沒關係*/
// <e>MCU ADC CH1 Param Config
```

```
//  <o>MCU ADC CH1 AF Port
//   <i> Default:PC1
//   <0=> PC1
#define MSP_ADC_CH1_AF_PORT 0
// </e>


/*通過獨立的位計算給宏賦值*/
// <q.0> Enable UART0 Rx ISR
// <q.1> Enable UART0 Rx DMA
#define MSP_UART0_RX_MODE 0x02

/*位帶運算*/
// <o.8..15>Set MCU Clock Output Port
//  <0=>GPIO_PORT_A
//  <1=>GPIO_PORT_B
//  <2=>GPIO_PORT_C
//  <3=>GPIO_PORT_D
//  <4=>GPIO_PORT_E
// <o.0..7>Set MCU Clock Output Pin
//  <0=>GPIO_PIN_0
//  <1=>GPIO_PIN_1
//  <2=>GPIO_PIN_2
//  <3=>GPIO_PIN_3
//  <4=>GPIO_PIN_4
//  <5=>GPIO_PIN_5
//  <6=>GPIO_PIN_6
//  <7=>GPIO_PIN_7
//  <8=>GPIO_PIN_8
#define MSP_CKO_PORT 0x104

/*下來選擇框中選擇給宏賦值*/
// <o>Set Internal high speed RC clock frequency(unit: hz)
//  <i>Default: 80MHz
//  <80000000UL=> 80MHz
//  <64000000UL=> 64MHz
//  <48000000UL=> 48MHz
//  <32000000UL=> 32MHz
//  <16000000UL=> 16MHz
//  <8000000UL=> 8MHz
```

```
#define MCU_CLK_FREQUENCE 64000000UL

/*在範圍內帶步長的輸入值進行宏賦值*/
// <o>Set SysTick Period
//  <i>Default: 1000 (Unit:us)
//  <0-16777215:3>
#define MCU_SYSTICK_FREQUENCE 9

// <s>Define MSP Config File Version
//  <i>version
#define MSP_CONFIG_VERSION "1.0.1"


// <<< end of configuration section >>
```

Configuration Wizard Annotations

Configuration Wizard annotations are items and modifiers which create the GUI-like elements. The following rules apply:

- Annotations are written as comments in the code.
- The Configuration Wizard section must begin within the first 100 lines of code and must start with:

  ```
  // <<< Use Configuration Wizard in Context Menu >>>
  ```

  Optionally, the section can explicitly end with:

  ```
  // <<< end of configuration section >>>
  ```
- The next value that follows the annotation is modified. However, items marked with can have a skip-value to jump a number of values (or lines in <c> and <!c> items). The example skips the first and modifies the second number that follows the comment.

  ```
  // <o1>
  ```
- White space characters are ignored in items or modifiers.
- Avoid using < or > within Configuration Wizard lines other than for enclosing annotation items.

The table lists the Configuration Wizard annotations:

Item	Text	Description
<c>	yes	Code enable: creates a checkbox to uncomment or comment code. All lines, including those with white spaces, get commented with double slashes (//) at the first found character when you disable the checkbox. Red text in the Configuration Wizard indicates an inconsistency. Do not nest this item.
<!c>	yes	Code disable: creates a checkbox to comment or uncomment code. All lines, including those with white spaces, get commented with double slashes (//) at the first found character when you enable the checkbox. Red text in the Configuration Wizard indicates an inconsistency. Do not nest this item.
</c>	no	End of code enable/disable block started with <c> or <!c>.
<h>	yes	Heading: the following options belong to a group.
<e>	yes	Heading with Enable: the following options belong to a group, which can be enabled via a checkbox.
<e.4>	yes	Heading with Enable: modifies a specific bit (bit 4, in this example).
</h> or </e>	no	Heading or Enable end.
<i>	yes	Tool-tip help for previous item.
<q>	yes	Option for bit values which can be set via a checkbox.
<o>	yes	Option with selection or number entry.
<o.4..5>	yes	Modify a range of bits (example: bit 4 to 5).
<o.4>	yes	Modify a single bit (example: bit 4).
<s>	yes	Option with ASCII string entry.

Item	Text	Description
<s.10>	yes	Option with ASCII string entry and a size limit of 10 characters.
Modifier	**Text**	**Description**
<0-31>	no	Value range for option fields.
<0-100:10>	no	Value range for option fields with step 10.
<0x40-0x1000:0x10>	no	Value range in hex format and step 10.
<0=>	yes	Value and text for selection.
<#+1> <#-1> <#*8> <#/3>	no	Value modification (add, sub, mul, div) before number is merged into field.

Item	Text	Description
<c>	yes	程式碼啟用：建立一個複選框以取消註釋或註釋程式碼。所有行（包括帶有空格的行）在停用複選框時，在第一個找到的字元處都會用雙斜槓 （//） 進行註釋。組態嚮導中的紅色文字表示不一致。不要巢狀此項目.
<!c>	yes	程式碼停用：建立一個複選框來註釋或取消註釋程式碼。啟用複選框時，所有行（包括帶有空格的行）都會在第一個找到的字元處用雙斜槓 （//） 進行註釋。組態嚮導中的紅色文字表示不一致。不要巢狀此項目。
</c>	no	以 <c> 或 <！c 開頭的程式碼啟用/停用塊結束>.
<h>	yes	標題：以下選項屬於一個組。
<e>	yes	帶有“啟用”的標題：以下選項屬於一個組，可以通過複選框啟用。
<e.4>	yes	帶啟用的標題：修改特定位（本例中為位 4）。
</h> or </e>	no	標題或啟用結束。
<i>	yes	上一項的工具提示幫助。

Item	Text	Description
<q>	yes	可通過複選框設定的位值選項。
<o>	yes	帶有選擇或數字輸入的選項。
<o.4..5>	yes	修改位範圍（例如：位 4 到 5）。
<o.4>	yes	修改單個位（例如：位 4）。
<s>	yes	帶有 ASCII 字串條目的選項。
<s.10>	yes	具有 ASCII 字串條目且大小限製為 10 個字元的選項。
Modifier	**Text**	**Description**
<0-31>	no	選項欄位的值範圍。
<0-100:10>	no	步驟 10 的選項欄位的值範圍。
<0x40-0x1000:0x10>	no	十六進制格式和步驟 10 的值範圍。
<0=>	yes	供選擇的值和文字.
<#+1> <#-1> <#*8> <#/3>	no	在數字合併到欄位中之前，值修改（add，sub，mul，div）。