

MsHandler CLASS 技術手冊

滑鼠事件處理統一架構

一、概述

MsHandler 是 STK 系統的滑鼠事件統一管理 CLASS。所有 browse 模組 (TABLE, BROWSE, ARRAY, BRRAY, WINFORM, P113, dbEditEx) 透過此 CLASS 處理滑鼠事件，確保邏輯一致、維護集中。

Singleton 模式：程式啟動時建立唯一實例

```
PUBLIC oMs := MsHandler():New() // main.prg, MouseInit() 之前
```

向後相容：所有方法有對應的 wrapper 函式，呼叫端不需直接使用 oMs

```
// 舊寫法 ( wrapper )      // 新寫法 ( 直接呼叫 )  
MsHitTest()               oMs:HitTest()  
MsHoverUpdate()           oMs:HoverUpdate()  
MsBrowseKey(0, oBrowse)   oMs:BrowseKey(0, oBrowse)
```

二、DATA 成員

DATA | 初始值 | 說明

nHoverIdx | 0 | 目前 hover 的 aMsReg 索引

nSayHoverIdx | 0 | 目前 hover 的 aSayReg 索引

ISayRegActive | .F. | aSayReg 是否啟用 (修改模式)

lInjectQuestion | .F. | 延遲注入 "?" 旗標

nInjectTarget | 0 | 延遲注入目標 GET index

nInjectCount | 0 | 延遲注入計數器

aRegStack | {} | PushReg/PopReg 狀態堆疊

三、核心 METHOD — BrowseKey

用途

統一所有 browse 模組的 idle 等待，取代 InKey(0, INKEY_ALL) + 手動滑鼠處理。

語法

```
nKey := oMs:BrowseKey( nTimeout, oBrowse )  
// 或透過 wrapper:  
nKey := MsBrowseKey( 0, oBrowse )
```

事件處理表

事件 | browse 內 | browse 外

K_MOUSEMOVE | HoverCheck + HoverUpdate + LOOP | 同 (消化不回傳)
 K_RBUTTONDOWN | RETURN K_ESC | RETURN K_ESC
 K_LDBLCLK(1006) | hb_keyPut(K_F10) + LOOP | LOOP (消化)
 K_MWFORWARD/BACKWARD | RETURN nKey | LOOP (消化)
 K_LBUTTONDOWN | RETURN nKey (行選擇) | MsClick→MsHitTest→hb_keyPut 或 LOOP
 其他鍵盤事件 | RETURN nKey | RETURN nKey

部署位置

所有 browse 模組的主迴圈 idle 等待處：

- WINFORM.PRG — 第 88 行
- TABLE.PRG — 第 209 行
- BROWSE.PRG — 第 136 行
- ARRAY.PRG — 第 177 行
- BRRAY.PRG — 第 107 行
- P113.PRG — 第 183 行
- dbedit_mouse.prg (dbEditEx) — idle 等待處

Hover 配合要求

呼叫端必須宣告：

```
PRIVATE aMouse := {}, nHover := 0
```

並在顯示功能提示時：

```
M->aMouse := BottomMessage( acMessage ) // TLIB 的全螢幕提示
M->aMouse := WindowMessage( ... ) // WINDOW 的視窗內提示
```

四、狀態隔離 — PushReg / PopReg

用途

進入修改模式時隔離滑鼠狀態，離開時還原。防止子層功能提示擾亂父層。

語法

```
MsPushReg() // 進入修改模式前 ( BottomMessage 之前 )
// ... BottomMessage(修改 hints) ...
// ... ReadModal(GetList) ...
MsPopReg() // 離開修改模式後
```

PushReg 儲存/清空的項目

項目 | 儲存 | 清空

```
M->aMsReg | | 清為 {}
M->aMouse | | 清為 {}
M->nHover | | 清為 0
nHoverIdx | | 清為 0
nSayHoverIdx | | 清為 0
ISayRegActive | | 清為 .F.
**M->aSayReg** | **不動** | **不動**
```

重要：aSayReg 不被 PushReg 清空，因為修改模式的 SAY 標頭需要它。

部署位置

- WINFORM.PRG — INSERT 和 MODIFY 區塊的 BottomMessage 之前

五、SAY 標頭系統 — SayReg

用途

讓修改模式中 GET 欄位旁的 SAY 標籤可以被點擊，點擊後切換 focus 到對應的 GET。

使用流程

```
// 1. 在 SAY+GET 定義後，READ 之前
aSayReg := {}
SayReg( 01, 00, "訂單編號", 1 ) // row, col, text, GET index
SayReg( 02, 00, "客戶代號", 2 )
SayReg( 03, 00, "品名規格", 3 )
// ...

// 2. READ 前啟用
SayRegEnable()

// 3. READ
READ

// 4. READ 後停用
SayRegDisable()
```

相關 METHOD

METHOD | 說明

SayRegAdd(row, col, text, oGet) | 註冊 SAY 標頭 (wrapper: SayReg)

SayRegClick() | K_LBUTTONDOWN 時檢查標頭點擊 (GETSYS 呼叫)

SayRegEnable() | 啟用 aSayReg hover/click

SayRegDisable() | 停用 aSayReg hover/click

SayRegClear() | 清空 aSayReg

SayRegHit(aSR, aGL, nKey) | 檢測命中

SetInject(nTarget) | 延遲注入 "?"

CheckInject() | idle 時執行注入 (StkCheckSysMenu 呼叫)

部署位置

- GETSYS.PRG — K_LBUTTONDOWN handler 呼叫 SayRegClick()
 - BUYSSELL.PRG — SayReg 註冊 + Enable/Disable
 - QUOT.PRG — 同上
 - INQUIRY/ENQUIRY/DOM*/FOR*.PRG — 同上
-

六、註冊系統 — aMsReg

用途

註冊螢幕上的可點擊區域 (功能提示、按鈕、GET 欄位等)。

格式

aMsReg[n] := { nRow, nColStart, nColEnd, nKey, nBtnIdx, cProcName, cSayStr, cNormColor }

相關 METHOD

METHOD | 說明

AddBtn(row, colS, colE, btnIdx, sayStr, color) | 註冊 dialog button

AddKey(row, colS, colE, key, sayStr, color) | 註冊功能提示/按鈕

MsClear() | 清空 aMsReg

HitTest() | 檢測 click 命中 · 有 flash 動畫

HoverUpdate() | K_MOUSEMOVE hover 變色/還原

HoverReset() | 清除 hover 狀態

部署位置

- BottomMessage (TLIB.PRG) — 自動呼叫 MsAddKey 註冊
- WindowMessage (WINDOW.PRG) — 自動呼叫 MsAddKey 註冊
- 各模組的 DIALOG/MENU — 手動呼叫 MsAddBtn/MsAddKey

七、aMouse 系統 (BottomMessage/WindowMessage)

用途

BottomMessage 和 WindowMessage 回傳的 LOCAL 陣列 · 供 hover 和 click 使用。

格式

aMouse[n] := { nRow, nCol, nWidth, cSayStr, cColor, nKey, cSource }

相關 METHOD

METHOD | 說明

HoverCheck(aMouse, @nHover) | aMouse hover 變色/還原

MsClick(aMouse, @nHover) | aMouse click 檢測 + flash

八、鍵盤解析 — KeyFromLabel

用途

解析功能提示文字中的 key glyph · 回傳 Harbour key code 。

語法

```
nKey := oMs:KeyFromLabel( "F1:查詢" ) // 回傳 K_F1
```

重要原則

必須使用 CUSTOMCHAR.ch 的 KY_* 定義 · 不可硬編碼 glyph code 。

支援 KY_ALT/KY_ALT2 + KY_CTRL/KY_CTRL2 雙格式 。

九、其他 METHOD

METHOD | 說明

New() | 建構子

IsClick(nKey) | 判斷 K_LBUTTONDOWN 或 1006

IsRight(nKey) | 判斷 K_RBUTTONDOWN

IsMove(nKey) | 判斷 K_MOUSEMOVE

ReadKey(nTimeout) | Dialog/Menu 用的 InKey wrapper

MenuSel(altems, nCurrent, nLevel) | 取代 MENU TO 的滑鼠選單

SayClick(row, col, text, bAction) | SAY + 點擊動作

AutoRegGetLabels(aGetList) | 自動註冊 GET 標籤

十、完整 wrapper 函式列表

wrapper 函式 | CLASS METHOD

MouseInit() | oMs:Init()

MsAddBtn() | oMs:AddBtn()

MsAddKey() | oMs:AddKey()

MsHoverUpdate() | oMs:HoverUpdate()

MsHoverReset() | oMs:HoverReset()

MsClear() | oMs:MsClear()

MsHitTest() | oMs:HitTest()

MsReadKey() | oMs:ReadKey()

MsMenuSel() | oMs:MenuSel()

MsKeyFromLabel() | oMs:KeyFromLabel()

MsHoverCheck() | oMs:HoverCheck()

MsClick() | oMs:MsClick()

SayClick() | oMs:SayClick()

SayReg() | oMs:SayRegAdd()

AutoRegGetLabels() | oMs:AutoRegGetLabels()

MsCheckInjectQuestion() | oMs:CheckInject()

SayRegClear() | oMs:SayRegClear()

SayRegHit() | oMs:SayRegHit()

SayRegClick() | oMs:SayRegClick()

SayRegSetInject() | oMs:SetInject()

SayRegEnable() | oMs:SayRegEnable()

SayRegDisable() | oMs:SayRegDisable()

MsPushReg() | oMs:PushReg()

MsPopReg() | oMs:PopReg()

MsBrowseKey() | oMs:BrowseKey()

文件版本 : 1.0

建立日期 : 2026-04-17

檔案位置 : E:\convertFW\HBSTK\compat\mouse_handler.prg