

# **Developer**

Georg Steger

**COLLABORATORS**

	<i>TITLE :</i> Developer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Georg Steger	August 6, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

---

# Contents

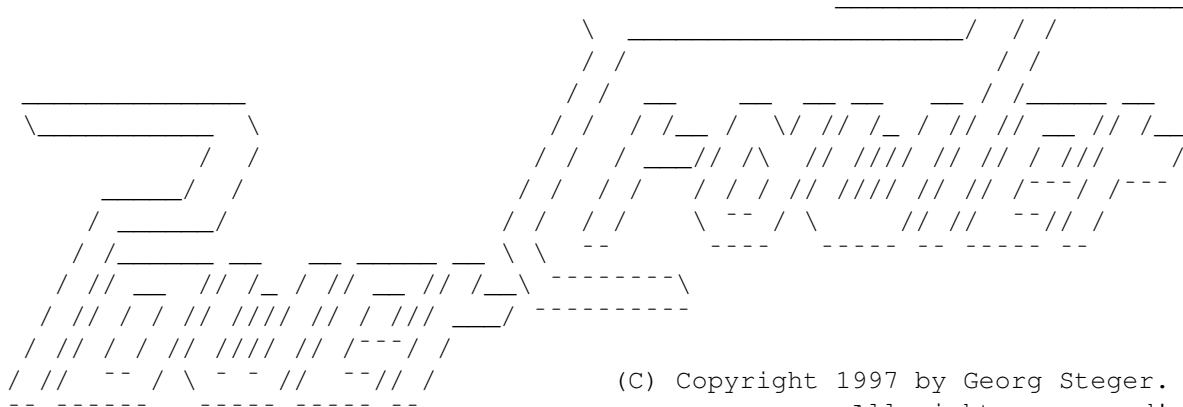
<b>1</b>	<b>Developer</b>	<b>1</b>
1.1	Developing external Routines for PowerCrawler . . . . .	1
1.2	Documetation for Map-IO Routines . . . . .	1
1.3	Documentation for external Picture-Loading-Routines . . . . .	1
1.4	How to code external IO-Routines . . . . .	2
1.5	How to code external Picture-Loaders for PowerCrawler . . . . .	3

---

# Chapter 1

# Developer

## 1.1 Developing external Routines for PowerCrawler



Map IO-Routines

Picture Loaders

## 1.2 Documetation for Map-IO Routines

Documentation  
Example Source  
Empty Source

## 1.3 Documentation for external Picture-Loading-Routines

Documentation

---

## 1.4 How to code external IO-Routines

You have to use an Assembler to create IO-Routines for PowerCrawler. The code has to be saved in absolute mode (don't create Executables!!). It is very important to write it PC-relative and to initialize the variables because the Routine is only loaded once. Include `PowerCrawler.i` into the Source-Code!

Your Routine must have the following header:

```

RECOGNITION:
    dc.b 'PCEX'

ENTRY:
    bra.s Main    ; must be 4 Bytes long!!!!
    nop          ; (therefore the »nop«)

ACTIONCODE:
    dc.w 0       ; You will find PC_IO_LOADMAP
                ; PC_IO_SAVEMAP or PC_IO_GETINFO here!

RETURNCODE:
    dc.w 0       ; Put PC_RETURN_???? here
                ; before leaving

RESERVED1: dc.l 0 ; DON'T LEAVE THIS OUT!!!!
RESERVED2: dc.l 0 ; "      "      "      "
RESERVED3: dc.l 0 ; "      "      "      "

ACTIONVAR:
    dc.l 0       ; Points to the Filename

```

At 4(sp) you find a Pointer to the PCInformation-Structure (See `Include` ).

The Loading-Function (`PC_IO_LOADMAP`) must:

- AllocVec the memory for the new Map
- update the `pci_MapData` Structure (part of PCInformation)
- make a FreeVec of `pci_MapBuffer` (if it is <> 0!!)
- update the `pci_MapBuffer` Pointer (with the AllocVec's result)

The Saving-Function (`PC_IO_SAVEMAP`) finds out:

- the Address of the MapBuffer in `pci_MapBuffer`
- the Width, Height, # Layers ... of the Map in `pci_MapData`

The GetInfo-Function (`PC_IO_GETINFO`) must:

- set ACTIONVAR to the address of an Info-Text which PowerCrawler will display using `EasyRequestArgs`!

The Routines have to be put in the "ExternalIO"-Directory!

VERY IMPORTANT:

IO-Routines should only touch (read or write) the following parts of the PC Information Structure:

```
pci_MapData
pci_MapBuffer
```

All the other stuff may (or may not) be free for access in future Versions of PowerCrawler (For Plugins or something like that). DONT TOUCH IT NOW!

## 1.5 How to code external Picture-Loaders for PowerCrawler

You have to use an Assembler to create Pic-Loader-Routines for PowerCrawler. The code has to be saved in absolute mode (don't create Executables!!). It is very important to write it PC-relative and to initialize the variables because the Routine is only loaded once. Include PowerCrawler.i into the Source-Code!

Your Routine must have the following header:

```
RECOGNITION:
    dc.b 'PCEX'

ENTRY:
    bra.s Main    ; must be 4 Bytes long!!!!
    nop          ; (therefore the »nop«)

ACTIONCODE:
    dc.w 0        ; You will find PC_PL_LOADPIC
                  ; or PC_PL_GETINFO here!

RETURNCODE:
    dc.w 0        ; Put PC_RETURN_???? here
                  ; before leaving

RESERVED1: dc.l 0 ; DON'T LEAVE THIS OUT!!!!
RESERVED2: dc.l 0 ; " " " "
RESERVED3: dc.l 0 ; " " " "

ACTIONVAR:
    dc.l 0        ; Points to the Filename of
                  the Picture you should
                  try to load
```

At 4(sp) you find a Pointer to the PCInformation-Structure (See Include ).

The Loading-Function (PC\_PL\_LOADPIC) must:

- Open a Screen using the following Tags:

```
SA_BEHIND,TRUE,
```

```
SA_WIDTH, <Width of the Picture>,
SA_HEIGHT, <Height of the Picture>,
SA_DEPTH, <# Planes of the Picture>,
SA_QUIET, TRUE,
SA_SHAREPENS, TRUE,
SA_DISPLAYID, <Value in pci_PicDisplayID>,
SA_AUTOSCROLL, TRUE,
SA_OVERSCAN, <Value in pci_PicOverScanType>
```

- On the Screen open a window using the following Tags:

```
WA_CUSTOMSCREEN, <Pointer to your Screen>
WA_LEFT, 0,
WA_TOP, 0,
WA_WIDTH, <Width of the Picture>,
WA_HEIGHT, <Height of the Picture>,
WA_BORDERLESS, TRUE,
WA_REPORTMOUSE, TRUE,
WA_RMBTRAP, TRUE,
WA_IDCMP, IDCMP_MOUSEBUTTONS | IDCMP_INACTIVEWINDOW | IDCMP_RAWKEY |
        IDCMP_MOUSEMOVE | IDCMP_VANILLAKEY
```

- Create a Colormap-Chunk (using AllocVec) like the ones in IFF-Files

- If you support Masks then you have to:

```
AllocBitmap(<Width of PIC>, <Height of PIC>, <Planes of PIC>, ???, <Bitmap of ↵
        your Screen>)
```

In the Bitmap each Pixel being a Mask-Pixel must be of Color  $(2^{\text{Planes}} - 1)$  [= each Bitplane set] and each Pixel not being Part of the Mask must be of Color 0 [= each Bitplane cleared].

Example: 4 Bitplanes --> Mask-Pixel must be Color 15  
 --> Non-Mask Pixel must be Color 0

- If loading is successfull then:

a) if pci\_PicScreen<>0 then:

- Close pci\_PicScreen->FirstWindow after having replied all outstanding Messages
- Close pci\_PicScreen

b) if pci\_MaskBitmap<>0 then make a FreeBitmap-Call to it

c) if pci\_PicColors<>0 then make a FreeVec-Call to it

d) Update: pci\_PicScreen (pointer to your Screen)  
 pci\_PicColors (pointer to your Colormap-Chunk)  
 pci\_PicPlanes (# of Planes of the Picture)

pci\_MaskBitmap (if you have created a Mask-Bitmap)

Note: You don't need to set up the Palette of the Screen.

PowerCrawler does that for you!

The GetInfo-Function (PC\_PL\_GETINFO) must:

- set ACTIONVAR to the address of an Info-Text which PowerCrawler will display using EasyRequestArgs!

The Routines have to be put in the "ExternalPL"-Directory!

VERY IMPORTANT:

PL-Routines should only touch (read or write) the parts of the PC Information Structur mentioned above!

All the other stuff may (or may not) be free for access in future Versions of PowerCrawler. DONT TOUCH IT NOW!

---