MAIL*paste

Closed-system Integration with Mail*STAR

If the target application can be modified, use IntegrationTutorial.pdf to guide open system integration.

If the target application is a closed system, or you need the added investigation capability of the address painter, you should consider this alternate integration method.

How MPASTE Works

Mail*STAR can emulate keyboard and mouse input of local non-iconified windows. <Clicking> on the "copy" icon triggers pasting CASS address data into the target Windows application.

a) Finds the target application's window using all or part of the window's title

b) Sets the target window as the foreground window causing it to receive input

c) Issues a mouse click on the first target address data field

d) Formats and pastes CASS certified address data navigating through all the target fields

e) Restores Mail*STAR as the foreground window

Integrating a Closed Windows Application

Three configuration keys found in C:\AES\cfg\config.svc control closed-system integration.

- MPASTE_TARGET will have all or part of the target window's title.
- MSPASTE directs navigation through the target window's address fields pasting address data.
- MSPASTE_LOG supports integration and should be removed for production.

Demonstration Setup

1. Set keys shown below in C:\AES\cfg\config.svc using the last line of this page for the MSPASTE value.

```
[PROFILE PAINTER]
MSPASTE_TARGET=Integration prototype
MSPASTE={100/527,137/547}{HOME}+{END}{DELETE}<aux>{TAB}{HOME}+{END}{DELETE}… explained [1]
MSPASTE_LOG=C:\AES\log\MSPASTE_TARGET_KEYBD.LOG
```

2. Get a copy of mspaste.dll from AES support and put it in C:\AES\bin
3. Start Mail*STAR – Start→Programs→AES→Mail-STAR
4. Start TestClient – Start→Programs→AES→TestClient

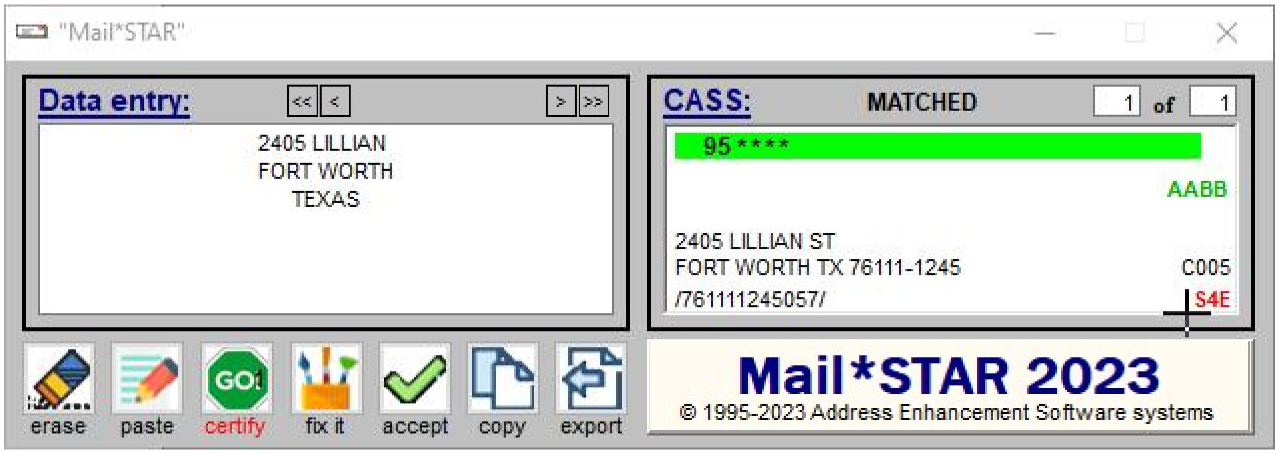
[1] The MPASTE directive begins with a mouse click on the first target field using client field coordinates {100/527,137/547}. No matter where the cursor is within a field, <HOME> <SHIFT><END> <DELETE> clears it. There are 26 address variables available to MPASTE. The directive to write address fields is <VAR> where "VAR" is the three-character address variable name. The directive <aux> pastes auxiliary address line three into the current field. The directive <TAB>'s, clears and writes address data for the delivery line <del>, city name <cty>, state-code <stc> and compound Zip <zp5>-<zp4> specifying insertion of the literal dash. See Appendix B for MPASTE syntax. The demo requires the following MPASTE directive.

```
{100/527,137/547}{HOME}+{END}{DELETE}<aux>{TAB}{HOME}+{END}{DELETE}<del>{TAB}{HOME}+{END}{DELETE}<cty>{TAB}{HOME}+{END}{DELETE}<stc>{TAB}{HOME}+{END}{DELETE}<zp5>-<zp4>
```

Mail*STAR Demonstration

The user enters an address into Mail*STAR's Data entry: field and <clicks> "GO". The certified address appears in the CASS: field to the right. The user sees address quality detail:

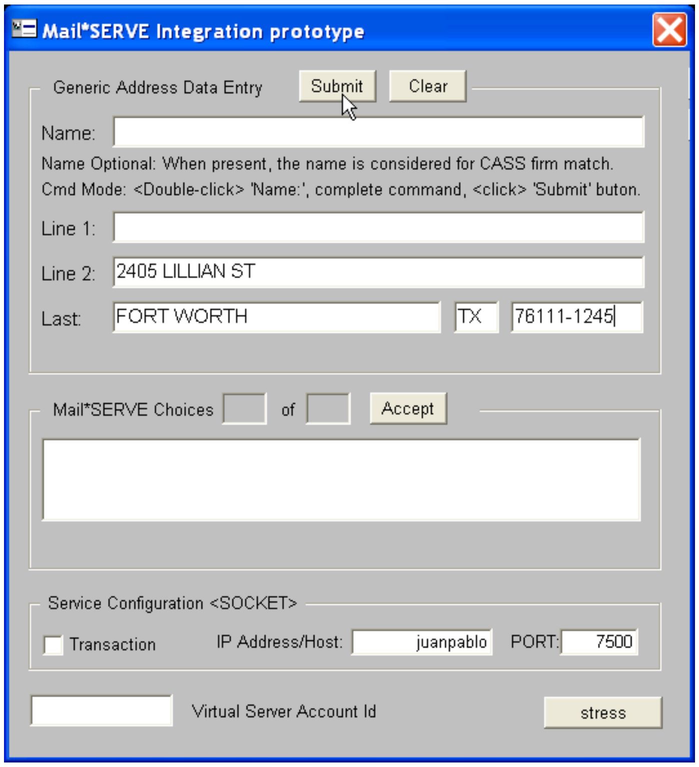
- ✓ "88 ****" – text in the green bar reports reliability and that the address is delivery point valid
- ✓ "AABB" – text in green signifies that the address is deliverable - uses USPS standard DPV footnotes
- ✓ "S4OE" – text in red reports - DPV valid street address with orthographic change and added suffix[2]



2405 LILLIAN
FORT WORTH
TEXAS

95 ****
AABB
2405 LILLIAN ST
FORT WORTH TX 76111-1245
/761111245057/
C005
S4E

<Click> of the "copy" icon starts keyboard emulation.


Mail*SERVE Integration prototype
Generic Address Data Entry
Submit
Clear
Mail*SERVE Choices
of
Accept
Service Configuration <SOCKET>
Transaction
Virtual Server Account Id
stress

Name:

Name Optional: When present, the name is considered for CASS firm match.
Cmd Mode: <Double-click> 'Name:', complete command, <click> 'Submit' buton.

Line 1:

Line 2: 2405 LILLIAN ST

Last: FORT WORTH TX 76111-1245

IP Address/Host: juanpablo PORT: 7500

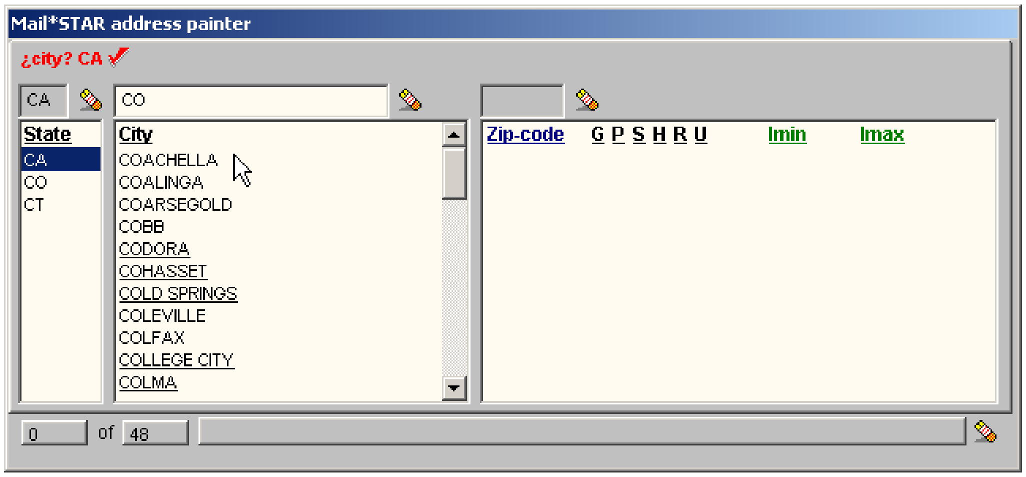
[2] See Mail*STAR tutorial for complete usage details.

Painter Demonstration – minimal key-entry

Mail*STAR has a minimal key-entry painter for hard to decipher addresses. <Click> the Mail*STAR paint icon to open the child address painter window.

Start by typing “C” in the state-code field reducing states to three possibilities. We clicked “CA”. Alternately typing “CA” would have done the same thing avoiding the click.

With “CA” established, there were 1,239 cities listed for California. Typing “C” reduced the list to 270, typing an “O” to 48 and “S” reduced the list to only one, “COSTA MESA CA” which became the scope for queries using the painter.


Mail*STAR address painter

¿city? CA ✓

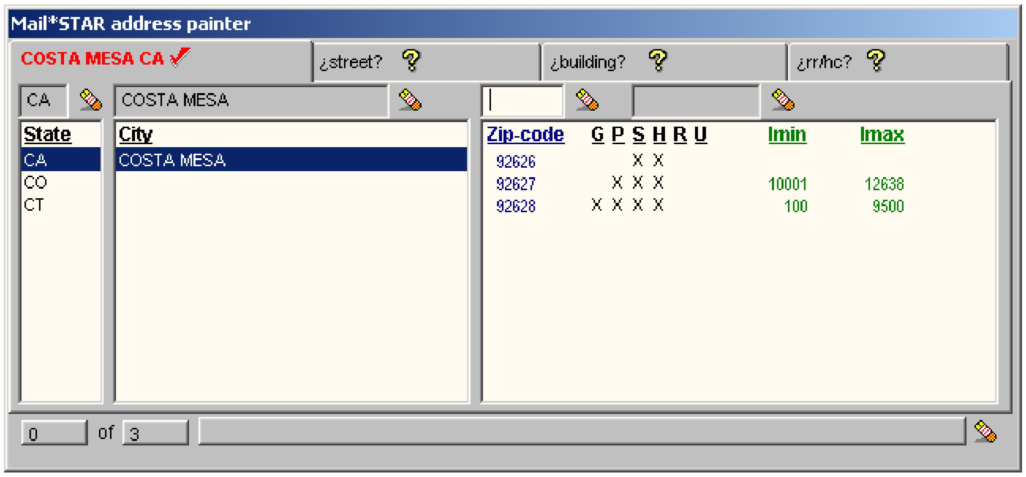
CA | CO

State	City
CA	COACHELLA
CO	COALINGA
CT	COARSEGOLD
	COBB
	CODORA
	COHASSET
	COLD SPRINGS
	COLEVILLE
	COLFAX
	COLLEGE CITY
	COLMA

Zip-code	G	P	S	H	R	U	lmin	lmax

0 of 48

The user can click a city to establish a list of Zip codes at any time. In this example that was done automatically when the list was reduced to only one possible city.


Mail*STAR address painter

COSTA MESA CA ✓ | ¿street? | ¿building? | ¿rr/hc?

CA | COSTA MESA

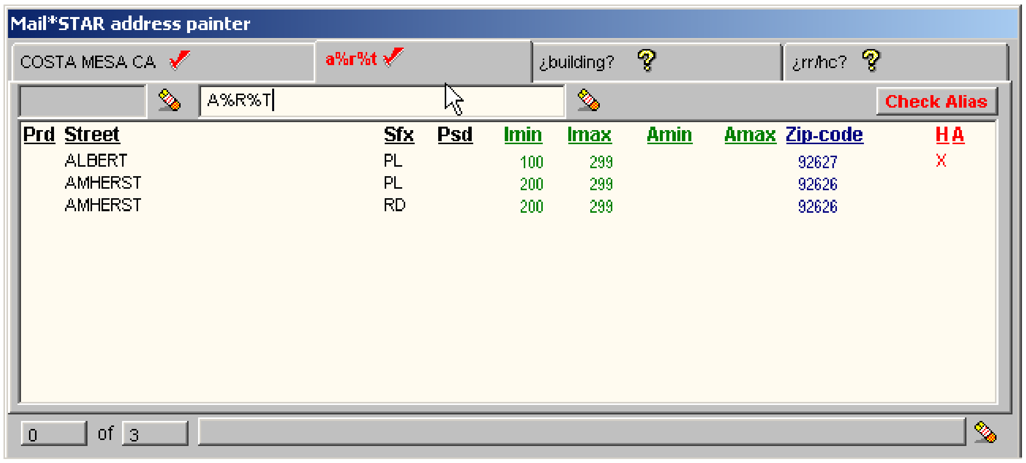
State	City
CA	COSTA MESA
CO	
CT	

Zip-code	G	P	S	H	R	U	lmin	lmax
92626			X	X				
92627		X	X	X			10001	12638
92628	X	X	X	X			100	9500

0 of 3

Once the city is established, other tabs appear to investigate streets, buildings and rural routes in the city.

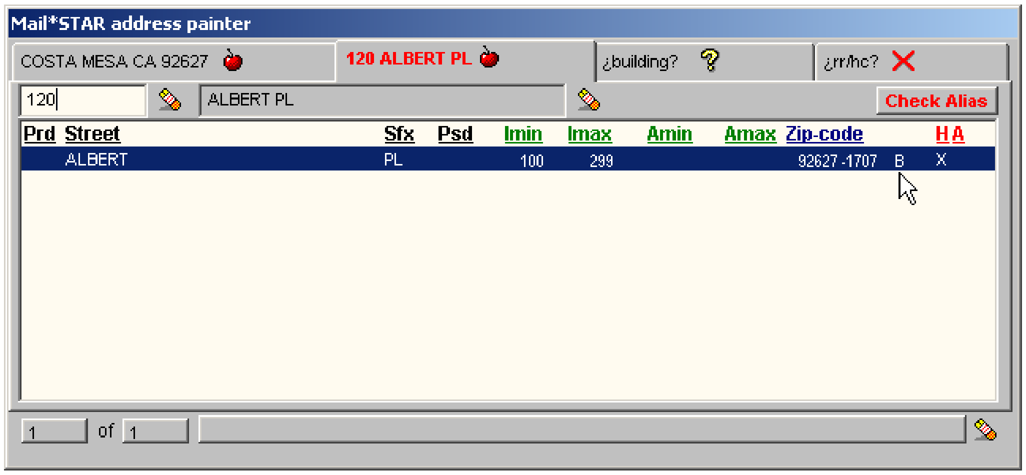
<Click> the “street” tab. Enter the clue “A%R%T” for streets that starts with “A” and have an “R” and end with “T”. The clue results in three streets.



Prd	Street	Sfx	Psd	lmin	lmax	Amin	Amax	Zip-code	H	A
	ALBERT	PL		100	299			92627	X	
	AMHERST	PL		200	299			92626		
	AMHERST	RD		200	299			92626		

0 of 3

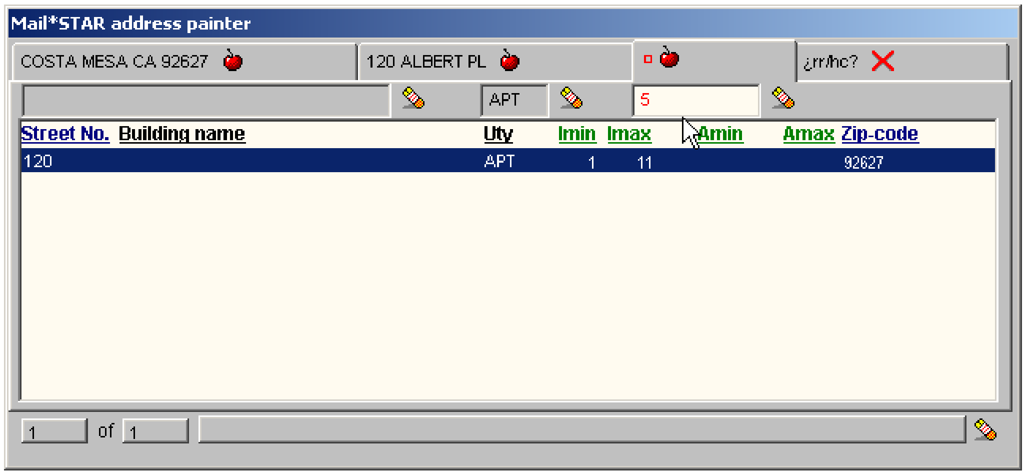
Typing “120” narrows the list down to only one possible street in “COSTA MESA”.



Prd	Street	Sfx	Psd	lmin	lmax	Amin	Amax	Zip-code	H	A
	ALBERT	PL		100	299			92627 -1707 B	X	

1 of 1

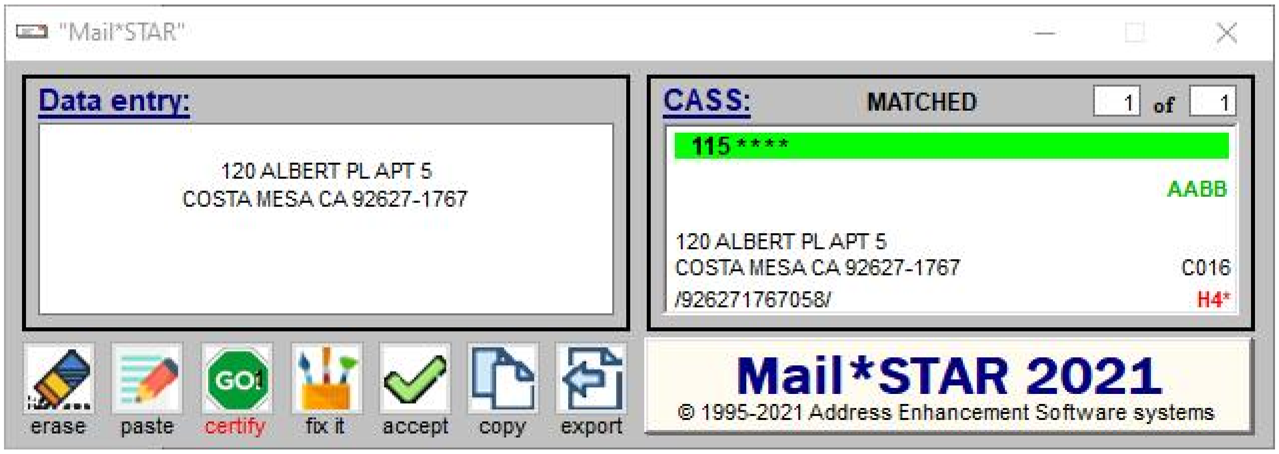
The “B” pointed to by the arrow tells us to <click> building. Here we enter apartment number “5”.



Street No.	Building name	Uty	lmin	lmax	Amin	Amax	Zip-code
120		APT	1	11			92627

1 of 1

As the painter investigates an address, it writes the address to the Mail*STAR window. <Click> "accept" to move the painter address to the data entry side and then <click> GO to certify the address.


"Mail*STAR"

Data entry:

120 ALBERT PL APT 5
COSTA MESA CA 92627-1767

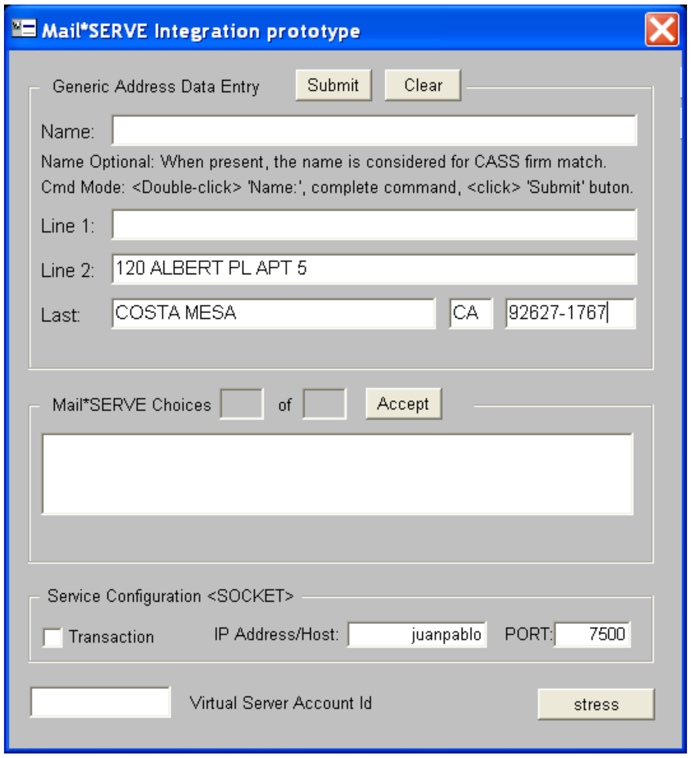
CASS: MATCHED 1 of 1

115 * * * *

AABB

120 ALBERT PL APT 5
COSTA MESA CA 92627-1767 C016
/926271767058/ H4*

This address took thirteen keystrokes to produce. "COSTA MESA CA" alone is thirteen characters long.


Mail*SERVE Integration prototype

Generic Address Data Entry

Name:

Name Optional: When present, the name is considered for CASS firm match.
Cmd Mode: <Double-click> 'Name:', complete command, <click> 'Submit' buton.

Line 1:

Line 2: 120 ALBERT PL APT 5

Last: COSTA MESA CA 92627-1767

Mail*SERVE Choices of

Service Configuration <SOCKET>

Transaction IP Address/Host: juanpablo PORT: 7500

Virtual Server Account Id

<Click> on "copy" executes the closed-system paste:

Target Window Analysis

This will de done by example using the TestClient target to construct the MPASTE and MSPASTE_TARGET keys. Repeat the same five steps for your application.

A – TAB stop order

From the target window, click the mouse on the first address field. Now use the <TAB> key to find the <TAB> stops. List every stop in order noting what data goes in each field, starting with 1) the first field entered with a mouse click and number every <TAB> stop even if it is not an address field, but stop when you get to the last address field. This is what we came up with for the TestClient screen.

1- <Click> on the field labeled "Line 1:" – **3rd address line**
2- <TAB> moves to the field labeled "Line 2:" – **delivery line**
3- <TAB> moves to the field labeled "Last:" – **city name**
4- <TAB> moves to the field after city – **state-code**
5- <TAB> moves to the last field – **Zip + '-' + Zip+4**

There might be intermediate <TAB> stops that are not used for address data. That is ok, but the skip field must be included in the numbered list labeled as "skipped".

QUESTION: Did <TAB> stops pass through every address field in the target window? If so, skip to the next step. If <TAB> stops did not include every address field, start a second lettered list starting with "A" listing address fields that cannot be reached with a <TAB>.

Try doing a mouse click to get into the fields in the second lettered list that were not <TAB> stops.

QUESTION: Can all the lettered list address fields be <clicked> into? If they can, skip to the next step. If they cannot, how do you get to these fields?

B – First address field coordinates

Appendix A will show you how to get the client area coordinates of the first target field.

If there were any fields in the second list of target address fields that were not <TAB> stops, get the client area coordinates of these address fields also.

If there were fields that you enter address data into that could not be <clicked> into, there has to be a command sequence for them. Beside the field in the third list, note the command sequence.

C – Input analysis

Below is an analysis of the TestClient. Being precise will pay off. If done carefully, this should work the first time. Starting from the field named "Line 1", paste requires the following user input:

What needs to happen	Keyboard Actions
1- Clear the field	<HOME> <SHIFT><END> <DELETE>
2- Write third address line	key characters from Mail*STAR address line 3
3- <TAB> and clear the field	<TAB> <HOME> <SHIFT><END> <DELETE>
4- write second address line	key characters from Mail*STAR delivery line
5- <TAB> and clear the field	<TAB> <HOME> <SHIFT><END> <DELETE>
6- write the city name	key characters from Mail*STAR city name
7- <TAB> and clear the field	<TAB> <HOME> <SHIFT><END> <DELETE>
8- write the state-code	key characters from Mail*STAR state-code
8- <TAB> and clear the field	<TAB> <HOME> <SHIFT><END> <DELETE>
9- write Zip plus '-' plus Zip+4	key characters from Mail*STAR Zip'-'Zip+4

If you can do it as a user, mspaste can too simulating keystrokes and mouse clicks.

D – Construct the MSPASTE directive

Refer to appendix B for MPASTE syntax.

Code MSPASTE piecemeal. Mail*STAR report errors that C:\AES\log\MSPASTE_TARGET_KEYBD.LOG will help clarify. Coding MPASTE a field at a time does not require starting and stopping Mail*STAR.

Mail*STAR reads MPASTE_TARGET and MPASTE_LOG on startup, but MSPASTE is refreshed every time the "copy" icon is clicked. All you need to do is <click> "copy" between incremental changes.

- **{100/527,137/547}**
  This directive causes a mouse click on the target window. The fractional coordinates make the click device and OS independent; { field_x / image_width , field_y / image_height }. Use values derived from Appendix A.
- **{HOME}+{END}{DELETE}** *see footnote* [3]
  This directive translates the analysis in C.1 above to clear the field the cursor is in.
- **<aux>** *see footnote* [4]
  Three-character names between '<' and '>' are address variables. The "aux" variable contains the third address line. This causes the CASS certified third line to be written into the field just cleared.
- **{TAB}{HOME}+{END}{DELETE}**
  The directive will go to the next <TAB> stop and clear the field. If there had been a skipped field **{TAB 2}** would issue two <TAB>'s and if there were 10 skipped fields **{TAB 11}** would issue 11 <TAB>'s.
- **<del>**
  The variable "del" contains the delivery line.
- **{TAB}{HOME}+{END}{DELETE}**
  The directive will go to the next <TAB> stop and clear the field.
- **<cty>**
  The variable "cty" contains the city name.
- **{TAB}{HOME}+{END}{DELETE}**
  The directive will go to the next <TAB> stop and clear the field.
- **<stc>**
  The variable "stc" contains the two-character state-code.
- **{TAB}{HOME}+{END}{DELETE}**
  The directive will go to the next <TAB> stop and clear the field.
- **<zp5>-<zp4>**
  This is a compound field having Zip followed by the literal '-' followed by Zip+4.

E – Comment MPASTE_LOG for production

```
[PROFILE PAINTER]
MSPASTE_TARGET=Integration prototype
MSPASTE={100/527,137/547}{HOME}+{END}{DELETE}<aux>{TAB}{HOME}+{END}{DELETE}<del>…
//MSPASTE_LOG=C:\AES\log\MSPASTE_TARGET_KEYBD.LOG
```

[3] Examining the first sequence to clear the current field "{HOME}+{END}{DELETE}":
- "{HOME}" – keyboard <Home> moves the cursor to the beginning of the current field
- "+{END}" – keyboard <Shift><End> selects all the text from the cursor to the end of the field
- "{DELETE}" – keyboard <DELETE> deletes selected text

[4] Address representation varies between applications. There are a total of 26 address variables to choose from while the demo only uses six. Address variables include both composite and discrete elements. The demo required discrete city, state, Zip and Zip+4. Your target might need the composite last line, "<lst>".

Appendix A – Mapping Target Window

Find the coordinates of the first data field needed for the mouse click directive. Using a screen capture program, copy a bitmap of the target window and paste it into "Paint". *see how-to footnote* [5]

Confirm the bitmap image by comparing its bottom right coordinates with target width and height in the paste log, C:\AES\log\MSPASTE_TARGET_KEYBD.LOG. "Paint" reported 527, 547 confirmed by the log:

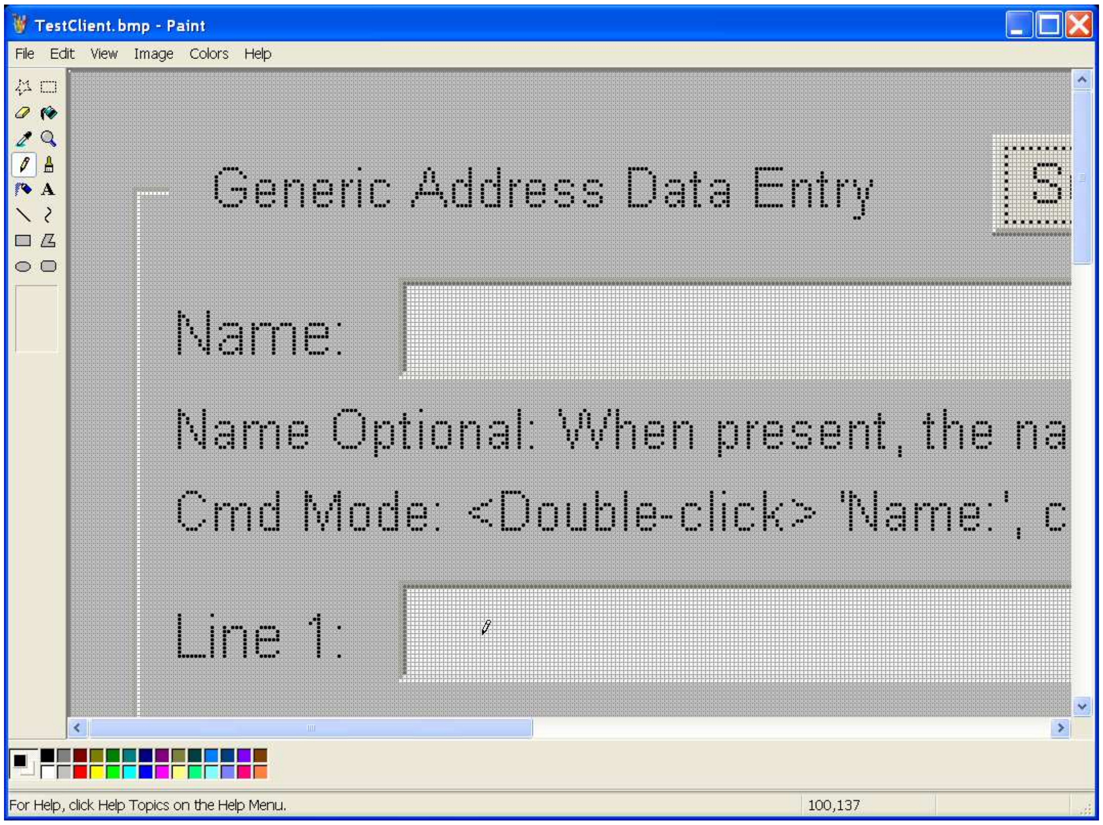
*** FOUND TARGET **width=527 height=547** *** - Mail*SERVE Integration prototype *see footnote* [6]

From "Paint" we did View→Zoom→Custom and chose 400% and then View→Zoom->ShowGrid.

The bottom of the image shows the location of the cursor as **100, 137**. Notice that the "Paint" cursor is in the middle of the field with respect to height and to the right a similar distance from the left margin of the field, splitting the difference between min-y and max-y and allowing adequate room for error in the x.

The field coordinates are expressed as a fractional pair for device and OS independence. This X is field_x / image_width and the Y is field_y / image_height. The TestClient coordinates are 100/527, 137/547.

Having found the first data field's coordinates, return to page 7, Target Window Analysis, step B.



Generic Address Data Entry

Name:

Name Optional: When present, the na

Cmd Mode: <Double-click> 'Name:', c

Line 1:

For Help, click Help Topics on the Help Menu.

100,137

[5] We used a trial version of SnagIt. From SnagIt 1) chose "Profiles" - "Object", 2) <PrintScreen>, 3) moved the mouse over the window capturing all but the title-bar and 4) <clicked>. The SnagIt editor opens with the image. <Click> the "clipboard" icon and paste into Paint.

[6] The objective is to get the log message reporting the target client's rectangle size. Code MSPASTE_TARGET in C:\AES\cfg\config.svc with enough of the target window's name to be unique. The target title is case sensitive.

Change MSPASTE=… to MSPASTE={TAB}. We don't want to start writing to the target window just yet.

Run Mail*STAR as shown on page 3, Mail*STAR Demonstration. If the target window is not found, Mail*STAR will report the error. Look at C:\AES\log\MSPASTE_TARGET.LOG for clues to adjust the MSPASTE_TARGET as needed.

The objective is to get the target client's rectangle size. Having done so, MSPASTE_TARGET configuration is done.

Appendix B – MPASTE Directive Syntax

Shift keys ('+', '^', '%') represent (<SHIFT>, <CTRL>, <ALT>) respectively

(i.e. - "+c" is <SHIFT>c - "^c" is <CTLR>c - "%c" is <ALT>c)
Can be applied to groups between '(' and ')' (i.e. "+(yes)" --> "YES")

CASS address elements between '<' and '>' (i.e. "<del>" delivery line)

<typ>	- address type
<pmb>	- private mailbox sequence
<urb>	- urbanization
<shd>	- shadow data
<bld>	- building name
<frm>	- firm name
<pno>	- primary number
<prd>	- predirectional
<str>	- street name
<sfx>	- street suffix
<psd>	- post directional
<uty>	- unit type (designator)
<sno>	- secondary number
<sn2>	- 2nd secondary number
<cty>	- city name
<stc>	- state-code
<zp5>	- Zip
<zp4>	- Zip+4
<dpc>	- three-digit delivery point includes the check-digit
<crt>	- carrier route
<lot>	- five-character line of travel
<z4f>	- address quality flags
<aux>	- calculated auxiliary line (see config.svc PMB_FORMAT, PR_BLDG, PR_RR_SHD, PR_PO_SHD)
<del>	- calculated delivery line
<lst>	- calculated last line
<bar>	- barcode

Special characters except for <SHIFT>, <CTRL>, <ALT> between '{' and '}'

Optionally indicate number of repetitions (i.e. "{TAB 12}")
For complete list see Appendic C, Named Keys

Mouse click client area coordinates between '{' and '}'

The field coordinates are expressed as a fractional pair for device and OS independence. This X is field_x / image_width and the Y is field_y / image_height. (i.e. “{100/527,137/547}”)

Literal characters

Except reserved characters below requiring their <SHIFT>ed equivalent

Reserved characters

'<' represented by "+,"
'>' represented by "+."
'(' represented by "+9"
')' represented by "+0"
'{' represented by "+["
'}' represented by "+]"
'+' represented by "+="
'^' represented by "+6"
'%' represented by "+5"

Appendix C – Named Keys

```
typedef struct NAMED_KEYS_CLASS
{           char *nam;
            BYTE vk;
            char ext;
} NAMED_KEYS;

NAMED_KEYS named_keys[] =
{           "BACKSPACE",VK_BACK,0,
            "BS",VK_BACK,0,
            "BKSP",VK_BACK,0,
            "BREAK",VK_PAUSE,1,
            "CAPSLOCK",VK_CAPITAL,0,
            "DELETE",VK_DELETE,1,
            "DEL",VK_DELETE,1,
            "DOWN",VK_DOWN,1,
            "END",VK_END,1,
            "ENTER",VK_RETURN,0,
            "ESC",VK_ESCAPE,0,
            "HELP",VK_HELP,0,
            "HOME",VK_HOME,1,
            "INS",VK_INSERT,1,
            "INSERT",VK_INSERT,1,
            "LEFT",VK_LEFT,1,
            "NUMLOCK",VK_NUMLOCK,1,
            "PGDN",VK_NEXT,1,
            "PGUP",VK_PRIOR,1,
            "PAUSE",VK_PAUSE,0,
            "PRINT",VK_PRINT,1,
            "PRTSC",VK_SNAPSHOT,1,
            "PRTSCN",VK_SNAPSHOT,1,
            "PRINTSCRN",VK_SNAPSHOT,1,
            "PRINTSCREEN",VK_SNAPSHOT,1,
            "RIGHT",VK_RIGHT,1,
            "SCROLLLOCK",VK_SCROLL,0,
            "SELECT",VK_SELECT,0,
            "TAB",VK_TAB,0,
            "UP",VK_UP,1,
            "F1",VK_F1,0,
            "F2",VK_F3,0,
            "F3",VK_F3,0,
            "F4",VK_F4,0,
            "F5",VK_F5,0,
            "F6",VK_F6,0,
            "F7",VK_F7,0,
            "F8",VK_F8,0,
            "F9",VK_F9,0,
            "F10",VK_F10,0,
            "F11",VK_F11,0,
            "F12",VK_F12,0,
            ""
};
```