

PRINT

{onerror preror,prerrmess} {indicate PRINT} {paneloff} {s} wuprint~{windowsoff}
{s} wutitle~{s} psnutitle~qag {s} wuintro~{s} psnuintro~qg {s} wutoc~{s} psnutoc~qg
{s} wumca~{s} psnumca~qg {s} wuri~{s} psnuri~pnp11~qqg
{s} wusymphony~{windowson} {s} wi {panelon} {indicate}

PREROR / STARTUP

{windowson} {beep} {panelon} {indicate ERROR} {paneloff} {s} wuerror~{s} wi{?}
{indicate} {paneloff} {s} wusymphony~{s} wi

Printer error

>>Be sure your printer is on line<<
Press [RETURN] to exit PRINT macro
| L T T T T T T Rc1
....User documentation is currently printing....

Printed output will include

- a title page,
- an introduction,
- a table of contents,
- and 15 pages of text.

| L Rc1
SYMPHONY<TM>
Macro Conversion Aid
and
Range Input Command
USER DOCUMENTATION

Copyright (c) 1984
LOTUS DEVELOPMENT CORPORATION

To print a formatted copy of this document,
press [USER], type PRINT, and press [RETURN].

| L T T T T T T T T T R11
| L R11
| L T T T T T T T T Rc1

—BSYMPHONY<TM>
MACRO CONVERSION AID
AND
RANGE INPUT COMMAND

LOTUS DEVELOPMENT CORPORATION

|@TEXT

Macro Conversion Aid and Range Input Command are copyrighted by Lotus Development Corporation.

You may copy Macro Conversion Aid, Range Input Command, and the accompanying user documentation for your use or the use of others, provided that you do not accept payment for copies that you make or distribute.

Lotus Development Corporation
161 First Street
Cambridge, MA 02142

NOTE: This copyright notice does NOT constitute permission to copy and distribute other Lotus products.

|::

|@HEADER

—BINTRODUCTION~

|@TEXT

Two Symphony add-in applications are specifically designed to help the 1-2-3<TM> user who is now working with Symphony.

—BMacro Conversion Aid (MCA.APP)~ translates most 1-2-3 macros to Symphony macros.

—BRange Input Command (INPUT.APP)~ adds the equivalent of the 1-2-3 /Range Input command to the Symphony SHEET menu.

You can use Range Input Command in conjunction with Macro Conversion Aid to execute converted 1-2-3 macros that require the Range Input command. You can also use it alone if you want to manually execute the Range Input command in Symphony, or to write Symphony macros that use the Range Input command.

|@HEADER

—BWhat you Need to Know~

|@TEXT

Before you use Macro Conversion Aid and/or Range Input Command, you should be familiar with how Symphony works. If you are unfamiliar with the subjects listed below, refer to the Symphony documentation.

You should know how to execute commands using Symphony menus.

You should know how to edit a macro. In addition, you may want to redesign some of your macros to take advantage of the new features and more powerful programming instructions that Symphony provides.

|@HEADER

—BGetting Ready~

|@TEXT

If you want to use the Macro Conversion Aid and/or the Range Input Command, you must transfer the following file(s) to your system:

For Macro Conversion Aid, you need MCA.APP and MCA.HLP.

For Range Input Command, you need INPUT.APP.

—BIf you are using a hard disk system, copy the files you need to your Symphony directory.

If you are using a two-diskette system, copy the files to your Help and Tutorial Disk.~

|::

|@HEADER

—BTABLE OF CONTENTS~

|@TEXT

INTRODUCTION.....	i
PART I -- MACRO CONVERSION AID.....	1
What MCA Does.....	1
How to Use MCA.....	2
Retrieving a worksheet with the 1-2-3 macro.....	2
Loading MCA.....	3
Invoking MCA.....	3
Checking the converted macro.....	4
Executing the converted macro.....	4

What to do if the converted macro doesn't work properly.....	4
Reference.....	5
1-2-3 commands and key names that MCA cannot translate.....	5
Error messages during conversion.....	7
Error messages during macro execution.....	8
 PART II -- RANGE INPUT COMMAND.....	 11
What Range Input Command Does.....	11
How to Use Range Input.....	11
With MCA.....	12
As a new Symphony command.....	13
Reference -- Error messages.....	15
::	

|@HEADER

—B

PART I -- MACRO CONVERSION AID

What MCA Does~

|@TEXT

Macro Conversion Aid (MCA) is a Symphony add-in application that translates most 1-2-3 macros into Symphony macros.

In the range you specify, MCA replaces the following with their Symphony equivalents:

—UComplete~ 1-2-3 command sequences.

A complete 1-2-3 command sequence begins and ends in READY mode when executed. It begins with a slash (/) and ends at the next point at which the user could begin another command sequence with a slash.

The Symphony translation always starts with either {SERVICES} or {MENU}.

- o Special 1-2-3 programming (/X) commands.

MCA replaces these with Symphony Command Language statements. /XLprompt~location~, for example, becomes {GETLABEL "prompt",location}.

- o 1-2-3 special key names.

Most 1-2-3 key names are the same in Symphony. A few are converted: {DEL} becomes {DELETE}, {GRAPH} is replaced with the command sequence {MENU}GPQ, and {NAME} is replaced with {MENU}.

- o @TODAY.

@TODAY is replaced with @NOW. All other 1-2-3 @functions are also Symphony @functions and are not changed.

1-2-3 macros can also contain text (for example, titles or range names), formulas, and numerical data. MCA leaves these unchanged.

|@NOTE

NOTES: —UTwo~ 1-2-3 key names and —Useven~ 1-2-3 commands have no counterparts in Symphony. These are listed in —BMCA Reference~. If a 1-2-3 macro that you convert contains one of these key names or commands, MCA replaces it with a message identifying the untranslated key name or command, and continues conversion.

In 1-2-3, formulas can contain both numeric and string expressions (string expressions are treated as 0), but in Symphony, mixed formulas generate a value of ERR. To perform string arithmetic, use Symphony's string-handling capabilities.

|@HEADER

—B
How To Use MCA
~

|@TEXT

To convert one or more 1-2-3 macros to Symphony Command Language macros, the essential steps are:

Start Symphony and retrieve the worksheet containing the 1-2-3 macro(s) you want to convert.

2. Load MCA.APP into memory.
3. Invoke MCA.APP; specify the range that you want MCA to convert.

Check the converted macro for messages indicating untranslated commands or key names. If necessary, edit the original macro and reconvert it, or edit the converted macro.

5. Execute the converted macro and verify that it works as intended.

This section covers each of these steps in detail.

—BStart Symphony and retrieve the worksheet containing the 1-2-3 macro(s) that you want to convert.~

If you are using a two-diskette system, be sure to replace the Symphony Program Disk with the Help and Tutorial Disk when Symphony

asks you to do so.

Press [SERVICES], select File Retrieve, and specify the worksheet name.

If the worksheet has the .WRK (Symphony) file extension, you can retrieve it as you would any Symphony worksheet. If it has the .WKS (1-2-3) file extension, then you must include the file extension when you specify the worksheet.

2. —BLoad MCA.APP into memory.~

Press [SERVICES], select Application Attach, and specify MCA.APP.

If you have a two-diskette system, MCA.APP should be on your Help and Tutorial Disk. If you have a hard disk system, MCA.APP should be in the Symphony directory on your hard disk.

3. —BInvoke MCA.APP; specify the range that you want MCA to convert.~

Select Invoke and specify MCA.APP.

MCA prompts you to enter the range of the macro to be converted.

Specify a range that includes the —Uentire~ macro (or macros) that you want to convert.

(At this point, you may press [HELP] for information about what you should include in the range and the different methods for specifying a range.)

|@NOTE

NOTE: MCA only converts cells in the range you specify.

If, for example, you specify the macro name as the conversion range, and the macro name identifies only the first cell of the macro, then MCA will convert only the first cell of the macro.

|@TEXT

MCA does not convert anything except 1-2-3 commands, key names, and @TODAY. It leaves all other data in the range untouched. If you are uncertain of the boundaries of the macro(s) you want to convert, you can convert the entire active area of the worksheet. To do so, move to the top of your worksheet, and press [END] then [HOME] to highlight the entire active portion of the worksheet.

|@NOTE

CAUTION: Any label in the conversion range that includes a slash (/) is interpreted as a macro statement if it replicates a valid and complete sequence of 1-2-3 commands. If you convert the entire worksheet, or a range that includes more than macros, you may want to change any slashes outside the macro to some other character.

For example, if the label Hingham/Weymouth is in the conversion range, MCA converts it to Hingham{SERVICES}NYmouth. ("/Wey" is read as /Worksheet Erase Yes, which corresponds to [SERVICES] New Yes in Symphony.)

|@TEXT

MCA converts the 1-2-3 macro at the rate of about 100 cells per second. The screen is redrawn after each 100 cells, so you may be able to see the conversion taking place. MCA overwrites the original macro with the new Symphony macro.

|::

—BCheck the converted macro for messages indicating untranslated commands or key names. If necessary, edit the original macro and reconvert it, or edit the converted macro.~

If MCA does not translate a command or key name, it replaces it with a message in curly braces like:

```
{**'RJ' COMMAND DOESN'T EXIST IN SYMPHONY**}
```

or

```
{**'TABLE' KEY DOESN'T EXIST IN SYMPHONY**}
```

If the converted macro contains any of these messages, edit the original 1-2-3 macro so that it does not include the untranslatable commands or key names (/RJ or {TABLE} in the examples above), or edit the converted macro to include Symphony commands that perform the desired task.

See —BMCA Reference~ for a list of 1-2-3 commands and key names that MCA cannot translate, and for suggestions on how to replace them with Symphony commands.

5. —BExecute the converted macro and verify that it works as intended.~

The new macro has the same range name that it did in 1-2-3 (range definitions are preserved when you load a 1-2-3 worksheet in Symphony), so you can execute it by pressing [MACRO] and the appropriate letter.

You can also rename the macro using an ordinary range name and execute it with the [USER] key. See the Symphony documentation for information on naming and executing Symphony macros.

To have Symphony automatically execute the macro each time you retrieve the worksheet, press [SERVICES], select Settings Auto-Execute Set, and enter the macro name or point to the first cell in the macro. This corresponds to the special macro name —B\0~ in 1-2-3.

|@HEADER

—U

What to do if the converted macro doesn't execute properly

~

|@TEXT

If you get an error message during execution, see —BMCA Reference~ for errors that may be related to the conversion. If the error message is not listed there, see the Symphony documentation.

Verify that you have converted the entire macro and that it doesn't include any messages identifying untranslated commands or key names. See —BMCA Reference~ for a listing of the 1-2-3 commands and key names

that MCA cannot translate.

If your original 1-2-3 macro modifies itself during execution, and the modification depends upon the menu structure of 1-2-3 commands, then the converted version may not work. MCA is NOT designed to handle self-modifying macros.

|@HEADER

—B
MCA Reference
~

—U
1-2-3 commands and key names that MCA cannot translate
~

|@TEXT

MCA converts a 1-2-3 command or key name with no Symphony counterpart into a message in curly braces.

The 1-2-3 commands that MCA cannot translate and corresponding messages are:

—B
/Print File Clear Format {**'PFCF' COMMAND DOESN'T EXIST IN SYMPHONY**}
/Print Printer Clear Format {**'PPCF' COMMAND DOESN'T EXIST IN SYMPHONY**}
~

These two commands reset margins, page length, and the printer control sequence (called "setup" in 1-2-3 and "init-string" in Symphony) to the /Worksheet Global Default settings. To accomplish the same task in Symphony, you can write a series of macro commands to explicitly reset margins, page length, and printer control sequence to the values you want. The following example shows one possibility:

{SERVICES}PSML4~
R76~
T2~
B2~Q

PL66~Q
I\027!\013~QQ

—B /Range Input {**'RI' COMMAND DOESN'T EXIST IN SYMPHONY**}~

/Range Input restricts movement of the cell pointer to unprotected cells within a specified range. If you have attached INPUT.APP, you can enter {MENU}RI in the macro, and the Range Input command will work just as in 1-2-3. For information on the Symphony Range Input Command, see —BPart II~ of this document.

Whereas 1-2-3 relies on the Range Input command to manage forms-oriented data entry, Symphony provides data-entry forms that allow you to set up a database and enter information quickly and easily. For information on the FORM work environment and database management, see the Symphony documentation.

|::

—B

/Range Justify {**'RJ' COMMAND DOESN'T EXIST IN SYMPHONY**}~

/Range Justify formats a column of consecutive long labels with the right margin you establish (text is left justified).

Symphony allows you to format text in a DOC window, using margins, tabs, justification (none, left, right, even, center), and more. For information on the DOC work environment, including justification, see the Symphony documentation.

—B

/Worksheet Window Sync {**'WWS' COMMAND DOESN'T EXIST IN SYMPHONY**}

/Worksheet Window Unsync {**'WWU' COMMAND DOESN'T EXIST IN SYMPHONY**}

/Worksheet Window Clear {**'WWC' COMMAND DOESN'T EXIST IN SYMPHONY**}~

/Worksheet Window Sync synchronizes movement in two 1-2-3 windows.

/Worksheet Window Unsync allows independent movement in each window. Symphony enables you to create multiple windows and arrange them on the screen in any way you want. You can move in only one window at a time (window movement is unsynchronized). If

you want to move in more than one window, you must do so in each.

/Worksheet Window Clear restores a single window. In Symphony, you can delete, hide, isolate, and expose windows.

See the Symphony documentation for a detailed discussion of how to create and use windows.

The 1-2-3 key names that MCA cannot translate and corresponding messages are:

—B
{QUERY} {**'QUERY' KEY DOESN'T EXIST IN SYMPHONY**}~

In 1-2-3, [QUERY] repeats the most recent /Data Query operation. In Symphony, you must explicitly repeat the operation.

You can scan the macro to determine the most recent Query operation, and use the appropriate commands as follows:

—UQuery operation~	—UCommands~
Query Delete Yes	{MENU}QDY{ESC 2}
Query Extract	{MENU}QE{ESC 2}
Query Unique	{MENU}QU{ESC 2}
Query Find	{MENU}QF***{ESC 3}

|@NOTE

NOTE: Replace *** with the keystrokes you want to include during the Query Find operation. Use {?} to interrupt the macro and allow the user to enter keystrokes.

|@TEXT

—B
{TABLE} {** 'TABLE' KEY DOESN'T EXIST IN SYMPHONY**}~

In 1-2-3, [TABLE] repeats the most recent /Data Table operation.

In Symphony, you must explicitly repeat the operation, but you can accept the values that Symphony offers for table range and input cells: these are the values that you used the last time you performed the operation. The equivalent of /Data Table 1 is [MENU] Range What-If 1-Way. The equivalent of /Data Table 2 is [MENU] Range What-If 2-Way.

You can scan the macro to determine the most recent Range What-If operation, and use the appropriate commands as follows:

one input cell	{MENU}RW1~~
two input cells	{MENU}RW2~~~

|@NOTE

NOTE: If you execute a converted macro that contains a message identifying an untranslated command or key name, Symphony displays

"Unrecognized key/range name {...}"

and halts execution. Redesign or edit your original 1-2-3 macro so that it doesn't include these commands or key names, or edit the converted Symphony macro.

|@HEADER

—U
Error messages during conversion
~

|@TEXT

—BInput command too long~

CAUSE: MCA cannot translate complete 1-2-3 command sequences of more than 500 characters. The cell pointer highlights the cell where conversion stopped.

SOLUTION: Shorten the command sequence or subdivide it into two or

more complete sequences, and convert the remainder of the macro. (You can reconvert the entire macro: MCA does not change the portion that it has already converted.)

A complete 1-2-3 command sequence begins and ends in READY mode when executed. It begins with a slash (/) and ends at the next point at which the user could begin another command sequence with a slash. Two examples are:

```
/WGC15~
```

```
/GTLXxrange~Aarange~Bbrange~Ccrange~Ddrange~E~erange~Ffrange~  
OLAalabel~LBblabel~LCclabel~LDdlabel~LEelabel~LFflabel~  
DAaarange~ABbbrange~ACccrange~ADddrange~AEeerange~AFffrange~  
AQGBSYAQCQ
```

The first of these sets global column width to 15. It cannot be subdivided. The second sets up a graph. It is longer and can be subdivided. Although it contains considerably fewer than 500 characters, it does illustrate how a single command sequence can become rather long. To subdivide it, pick a point where it is convenient to return to READY mode, move out of the menu, then move right back in. For example, taking the end of the second line as a break point, you can exit by inserting two Quits, and reenter with /Graph Options. The original command sequence now appears as two complete command sequences:

```
/GTLXxrange~Aarange~Bbrange~Ccrange~Ddrange~E~erange~Ffrange~  
OLAalabel~LBblabel~LCclabel~LDdlabel~LEelabel~LFflabel~—BQQ  
/GO~DAaarange~ABbbrange~ACccrange~ADddrange~AEeerange~AFffrange~  
AQGBSYAQCQ
```

The macro is a little longer, but each of the new complete command sequences is significantly shorter than the original command sequence.

—BOutput command too long~

CAUSE: MCA cannot convert 1-2-3 command sequences if the translation requires more than about 1000 characters, or if it won't fit in the space occupied by the original command sequence. The cell pointer

highlights the cell where conversion stopped.

SOLUTION: Shorten the original command sequence that begins in the highlighted cell, or subdivide it into two or more command sequences. Convert the remainder of the macro.

See item 1 above for the definition of a complete 1-2-3 command sequence and an example of how to subdivide one.

If the error recurs in the same cell, you must redistribute the commands in that cell over two or more cells to give MCA room to place the translation. Each cell can hold up to 240 characters.

|@HEADER

—U

Error messages during macro execution

~

|@TEXT

1. —BUnrecognized key/range name {...}~

CAUSE: Symphony encountered an unrecognized key or range name in curly braces. You may have failed to remove a message which MCA has put in the macro identifying an untranslated command or key name.

SOLUTION: Edit the macro to replace this messages with valid Symphony commands. See —U1-2-3 commands and key names that MCA cannot translate~ above.

|::

--

—BSyntax error in macro key/range {...}~

CAUSE: You may have included quotes (") in the prompt-string of a /XL or /XN command.

MCA automatically places quotes around the prompt-string when it translates a /XL or /XN command, so that strings including argument separators (such as comma or semicolon) will be legal in Symphony.

If the original string includes quotes, however, then the conversion--
{GETLABEL...} or {GETNUMBER...}--will contain too many quotes.

For example, MCA converts

```
/XNenter "amount paid">~location~
```

into

```
{GETNUMBER "enter "amount paid">",location}
```

Symphony tries to interpret the string following the second quotation
mark as the location in which to enter the user's response.

SOLUTION: Remove the extra quotes:

```
{GETNUMBER "enter amount paid>",location}
```

—BToo many nesting levels in macro calls~

CAUSE: The worksheet may contain a range name corresponding to a key
name or a command keyword which did not appear in the original macro
but does appear in the converted macro. The possibilities are:

—UKey names~ —UCommand Keywords~

MENU	IF
SERVICES	BRANCH
DELETE	RETURN
	MENUBRANCH
	GETLABEL
	GETNUMBER

When this happens, Symphony interprets the key name or command keyword
in the macro as a call to a subroutine beginning in the cell
identified by that range name. See the description of
{routine-name...} in the section on command keywords in the Symphony
—BReference Manual~.

If you execute such a converted macro, the results are unpredictable, but if the macro calls or branches to this range, which in turn calls itself, you may get the message listed above.

|::

Look, for example, at the following fragment of a 1-2-3 macro. The /XG command branches to a range called "menu," which contains instructions to format a range:

```
/XGmenu~
```

```
MENU /RFF1~c20.d45~
```

When converted, the fragment appears as follows:

```
{BRANCH menu}
```

```
MENU {MENU}FF1~c20.d45~
```

If you execute this fragment, {BRANCH menu} branches to the range "menu." This range, however, contains {MENU}, which Symphony interprets not as the beginning of a Format command, but as a call to a subroutine beginning at range "menu." {MENU} then continues calling itself until Symphony displays "Too many nesting levels in macro calls," and execution halts.

SOLUTION: Rename the range, and adjust any references to it in your macro accordingly.

It is a good idea not to use Symphony key names or command keywords as range names. For a complete list of both, see the [Symphony —Reference Manual](#).

|::

|@HEADER

—B
PART II -- RANGE INPUT COMMAND

What Range Input Command Does
~

|@TEXT

Range Input Command (INPUT.APP) is a Symphony add-in application. When attached to Symphony, it adds the 1-2-3 /Range Input command to the Symphony SHEET menu. You can use the Range Input command manually or include it in Symphony macros.

The Range Input command limits the movement of the cell pointer to unprotected cells within a specified range. Its principal use in 1-2-3 is to provide forms-oriented data entry into a worksheet.

Symphony provides data-entry forms, so the primary function of Range Input Command is to replace the 1-2-3 /Range Input command in macros that you convert with MCA. You can, of course, redesign and greatly simplify these macros, or avoid the need of a macro altogether, by using Symphony's data-entry forms.

See the Symphony documentation for information on data-entry forms and database management.

|@HEADER

—B
How To Use Range Input
~

|@TEXT

You can use the Range Input command:

In macros that you convert with MCA.

- o As a new Symphony command.

|::

|@HEADER

—U

The Range Input command in macros that you convert with MCA

|@TEXT

To use the Range Input command in a macro that you convert from 1-2-3 to Symphony:

Before you use MCA to convert the macro, replace each occurrence of /RI with {MENU}RI. MCA will ignore {MENU}RI during conversion because it is not a valid 1-2-3 command.

OR

After you convert the macro with MCA, replace each occurrence of {**'RI' COMMAND DOESN'T EXIST IN SYMPHONY**} with {MENU}RI.

You must attach INPUT.APP before you execute the command in the macro.

If you have a two-diskette system, INPUT.APP should be on your Help and Tutorial Disk. If you have a hard disk system, INPUT.APP should be in the Symphony directory on your hard disk.

Press [SERVICES], select Application Attach, and specify INPUT.APP.

You can attach INPUT.APP in the macro. The macro sequence to

attach INPUT.APP is {SERVICES}AAinput~Q.

You can have Symphony automatically attach INPUT.APP at the beginning of each work session. See the section on the Configuration settings sheet in the Symphony —BReference Manual.

|@NOTE

NOTES: If the original 1-2-3 macro defines which cells are unprotected, the converted macro does so also. MCA converts /RU (Range Unprotect) to {MENU}RPA (Range Protect Allow-Changes). The "U" setting, which indicates unprotected cells in 1-2-3, becomes "A" in Symphony. Any 1-2-3 worksheet that you load in Symphony with File Retrieve retains its protection settings.

During execution of the Range Input command, it makes no difference whether Global-Protection is set to Yes or No on the Services settings sheet.

|::

|@HEADER

—U

The Range Input command as a new Symphony command

~

|@TEXT

To use the Range Input command in Symphony:

--
Load INPUT.APP into memory.

If you have a two-diskette system, INPUT.APP should be on your Help and Tutorial Disk. If you have a hard disk system, INPUT.APP should be in the Symphony directory on your hard disk.

Press [SERVICES], select Application Attach, and specify INPUT.APP.

Specify all cells that you want the user to be able to move to and make entries in during execution of Range Input.

Press [MENU], select Range Protect Allow-Changes, specify a cell or range of cells, and press [RETURN]. Repeat this for all the cells in which you want to allow changes.

3. Execute the Range Input command.

Press [MENU] in a SHEET window and select Range Input.

(Alternatively, you can press [SERVICES], select Application Invoke, and specify INPUT.APP. You must be in a SHEET window when you invoke INPUT.APP.)

Symphony prompts you for an Input range. Type in a range name or address, or use the pointer movement keys to highlight a range, and press [RETURN].

You cannot move the cell pointer outside this range while the Range Input command is in effect.

|@NOTE

NOTE: If all cells in the range that you specify are protected, Symphony displays the error message "No unprotected cells in input range." Press [ESCAPE] to clear the error, and set Range Protect to Allow-Changes for the cells to which you want access.

|@TEXT

Symphony positions the beginning of the Input range at the top left corner of the screen.

|::

During execution of Range Input, you can use the pointer movement

keys to move to any unprotected cell in the Input range, and you can enter or edit data in these cells. [HOME] moves the cell pointer to the first unprotected cell in the Input range. [END] moves the cell pointer to the last unprotected cell in the range. [UP], [DOWN], [LEFT], and [RIGHT] move the cell pointer in the appropriate direction to the next unprotected cell in the range. If you attempt to move the cell pointer beyond the end of the Input range, it moves to the beginning of the range.

You can also use the [EDIT], [ABS], [CALC], and [ZOOM] keys. Other special keys are disabled.

To terminate the Range Input command, press [ESCAPE] or [RETURN].

|::

|@HEADER

—B
Range Input Reference
~

—U
Error Messages
~

|@TEXT

--
—BNo unprotected cells in input range~

CAUSE: The Range Input command doesn't work unless the Input range includes unprotected cells.

SOLUTION: Use the Range Protect Allow-Changes command to unprotect the appropriate cells in the Input range.

--
—BRange Input requires SHEET window~

CAUSE: The Range Input command works only in a SHEET work environment.

SOLUTION: Press [TYPE] and select SHEET.

|::