Cat No. I56E-EN-01

Software

Trajexia Studio

USER MANUAL

OMRON

Notice

OMRON products are manufactured for use by a trained operator and only for the purposes described in this manual.

The following conventions are used to classify and explain the precautions in this manual. Always heed the information provided with them.

⚠ **WARNING**
Indicates information that, if not heeded, could possibly result in serious injury or loss of life.

⚠ **Caution**
Indicates information that, if not heeded, could possibly result in minor or relatively serious injury, damage to the product or faulty operation.

Visual aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

Trademarks and copyrights

MECHATROLINK is a registered trademark of Yaskawa Corporation.
Trajexia is a registered trademark of OMRON.
All other product names, company names, logos or other designations mentioned herein are trademarks of their respective owners.

Copyright

Copyright © 2008 OMRON
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

TABLE OF CONTENTS

TABLE OF CONTENTS

SECTION 1
Introduction

1-1 Introducing Trajexia Studio

Trajexia Studio is a software tool designed to develop complete solutions for motion control devices, or devices for short. Trajexia Studio provides tools to develop motion applications, configure devices, communicate with devices and monitor and debug motion programs that are running on devices.

1-1-1 Highlights

- Microsoft Visual Studio-style IDE
  Trajexia Studio has one single user interface where the user can perform all actions to develop motion applications, execute them on a device and adjust and optimise them. This user interface is an IDE (integrated development environment) that has the look and feel of the Microsoft Visual Studio-style IDE. Users that know the Microsoft Visual Studio IDE can work with the Trajexia Studio IDE without problems.
- Solutions for multiple devices
  Trajexia Studio supports the development for several devices at the same time, integrated in one single solution.
- Flexible development
  Trajexia Studio offers flexible options for the development of motion applications. The hardware and software characteristics of motion axes are separate, which means that your motion application can be developed hardware-independent.
- Support for existing motion applications
  Trajexia Studio can import Motion Perfect 2 projects. Therefore, you can use your existing motion applications in Trajexia Studio.
- Communicate with online devices
  Trajexia Studio can communicate with online devices, which are devices connected to the computer on which Trajexia Studio runs.

1-1-2 Intended audience

This documentation is intended for designers, developers and testers of motion applications. They must have a good knowledge of the concepts of motion control, devices and the BASIC programming language.

1-2 This documentation

This documentation contains the following sections.

Section	Contents
1	What is Trajexia Studio, who is it for, what do you need to use it
2	The Trajexia Studio user interface, its windows, how to use it, and how to adjust it to your preferences
3	The items to develop a motion application
4	How to execute motion applications, how to monitor their execution
5	How to solve problems in a motion application, how to optimise the motion application
A	An overview of all keyboard shortcuts in the code editor

1-3 Installation

1-3-1 System requirements

Trajexia Studio is a Microsoft Windows application. The minimum versions of Microsoft Windows that Trajexia Studio supports are:

- Windows 2000 SP3
- Windows XP SP2
- Windows Vista Business Edition

The table below lists the system requirements for the computer on which you install Trajexia Studio.

Item	Windows 2000 or XP		Windows Vista	
	Minimum	Recommended	Minimum	Recommended
Processor	333 MHz Pentium II or compatible	1 GHz Pentium III or compatible	As recommended by Microsoft	
RAM	256 MB	512 MB	512 MB	1 GB
Hard disk space	512 MB (including CX-Server)			
Display	SVGA 800 x 600 256 colours	XGA 1024 x 768 16 mill. colours	SVGA 800 x 600 256 colours	XGA 1024 x 768 16 mill. colours
Media drive	CD-ROM drive			
Communication ports	• Ethernet, for connection to TJ1-MC04, TJ1-MC16 or CJ1 or CJ2 PLC with CJ1W-MCH72 • RS232, for connection to CJ1 or CJ2 PLC with CJ1W-MCH72 • Other, for connection to CJ1 or CJ2 PLC with CJ1W-MCH72 (refer to the CJ1/CJ2 hardware manual)			

1-3-2 Installing Trajexia Studio

The Trajexia Studio installer is a standard InstallShield wizard. To start the installer, execute setup.exe, which is located in the Trajexia Studio installer folder. The InstallShield wizard guides you through the installation process.

The following remarks are important when you install Trajexia Studio:

- You must have administrator privileges on the computer where you want to install Trajexia Studio.
- If you have a Trajexia Studio licence code, you can enter it when you install Trajexia Studio. If you do not have a licence code, you can still install Trajexia Studio. For more information on the registration of Trajexia Studio and licence codes, refer to section 1-3-3.
- You can select which applications you want to install in the **Select Features** window:
  - CX-Drive
    CX-Drive is a configuration tool for drives. Trajexia Studio needs CX-Drive to perform drive tuning (see section 3-5-4-1).
  - CX-Server
    CX-Server is middleware to communicate with other OMRON components. Trajexia Studio needs CX-Server to communicate with an online device (see section 3-2-3).
  - Trajexia Studio

Trajexia Studio is the main application.

- If CX-Server is installed, it can be required to restart your computer after the installation. This depends on several factors, such as the operating system. If a restart is required, the Trajexia Studio installer indicates that.

1-3-3 Registering Trajexia Studio

The Trajexia Studio software can run in 2 modes:

- Trial mode
  In trial mode, Trajexia Studio runs 30 days after the installation. After 30 days, it stops working.
- Full mode
  In full mode, Trajexia Studio has no duration limitation.

You can register Trajexia Studio with a licence code. The licence code consists of 4 groups of 4 digits. The following registrations are available:

Licence code	Trajexia Studio mode
None	Trial mode
Trial licence	Trial mode
Full licence	Full mode

You can register Trajexia Studio during the installation, or in Trajexia Studio: select **About...** on the **Help** menu, and click **Upgrade Licence...**

1-4 Document conventions

This documentation uses the following typographical conventions:

Typography	Description

`FOR i=0 to 10`
`    SPEED = i`
`NEXT i`
	This font is used for BASIC program code

GLOBAL "ratio", 23	Bold words in syntax descriptions are words that must be typed literally
Solution Explorer	Bold words in other text are used to indicate the names of user interface elements
solution-item-1	Italic words in syntax descriptions or names of user interface elements are placeholders for other text
CTRL	Capital letters are used for the names of keys
CTRL+TAB	A plus sign (+) between 2 keys indicates a combination of keys that are pressed at the same time

1-5 Getting help

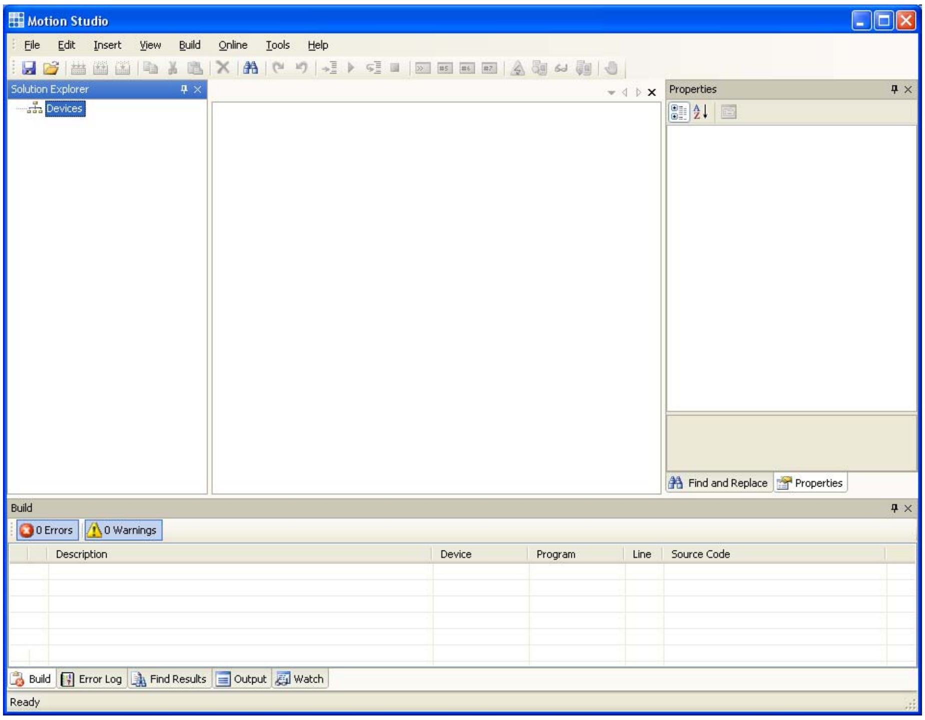
To view this documentation in Trajexia Studio, you can press F1 or select **Help...** on the **Help** menu.

The BASIC code editor offers context-sensitive help on BASIC keywords. To open the context-sensitive help on a BASIC keyword, place the cursor on a BASIC keyword in the code editor and press F1.

SECTION 2
Integrated Development Environment

2-1 Introduction

The Trajexia Studio IDE provides the tools you need to develop a motion application, such as a code editor, data entry windows, a compiler and a debugger.



Trajexia Studio IDE

The behaviour and appearance of the Trajexia Studio IDE can be customised to your personal preferences.

2-1-1 Window types

The Trajexia Studio IDE contains 2 types of windows:

- Document windows
  Document windows are dynamically created when you edit a solution item in Trajexia Studio.
- Docking windows
  The docking windows are predefined and statically created. They contain tools and feedback on the solution items and on the document windows. Docking windows are described in section 2-4.

Document windows and docking windows behave in a different way. Therefore, they cannot be grouped together in the IDE.

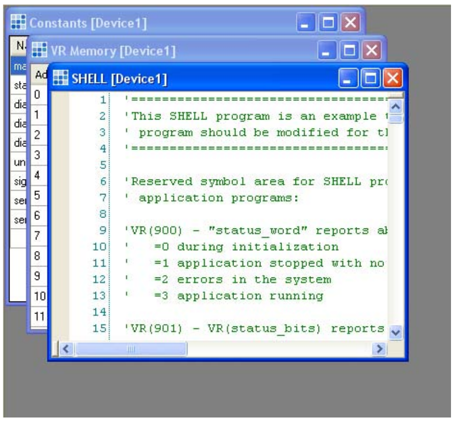
Some windows or user interface elements in Trajexia Studio have specific features. These are:

- Code editor
  The code editor is a text editor with advanced features such as code completion and syntax colouring. It is used to edit BASIC program code. The code editor is described in section 2-2.
- Tabular views
  A tabular view is a table of values in rows and columns, with special sorting and filtering features. Tabular views are described in section 2-3.

2-1-2 Layout

The Trajexia Studio IDE supports 2 different layouts:

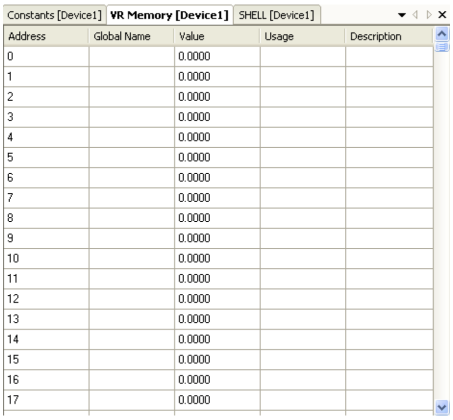
- Multiple-document view
  In the multiple-document view, the Trajexia Studio main window is the parent window and the document windows are child windows. In this view, the Trajexia Studio IDE has a **Window** menu that lists all open child windows. The child windows can be tiled and cascaded. You can use CTRL+F6 to move the focus to the next document window.



```
'=================================
'This SHELL program is an example t
' program should be modified for tl
'=================================

'Reserved symbol area for SHELL pro
' application programs:

'VR(900) - "status_word" reports al
'    =0 during initialization
'    =1 application stopped with no
'    =2 errors in the system
'    =3 application running

'VR(901) - VR(status_bits) reports
```

Multiple-document view

- Tabbed view
  In the tabbed view, all open document windows are grouped together in one single child window, which has a tab for each document window. In this view, there is no **Window** menu. You can use CTRL+TAB to move the focus to the next document window.



Address	Global Name	Value	Usage	Description
0		0.0000		
1		0.0000		
2		0.0000		
3		0.0000		
4		0.0000		
5		0.0000		
6		0.0000		
7		0.0000		
8		0.0000		
9		0.0000		
10		0.0000		
11		0.0000		
12		0.0000		
13		0.0000		
14		0.0000		
15		0.0000		
16		0.0000		
17		0.0000		

Tabbed view

To change the layout of the Trajexia Studio IDE, select **Multiple Document View** on the **View** menu. This toggles the layout between the multiple-document view and the tabbed view. The check mark on this menu item indicates that the current layout is the tabbed view. This is the default layout of the Trajexia Studio IDE.

Note The layout of the Trajexia Studio IDE has no influence on the behaviour of the docking windows.

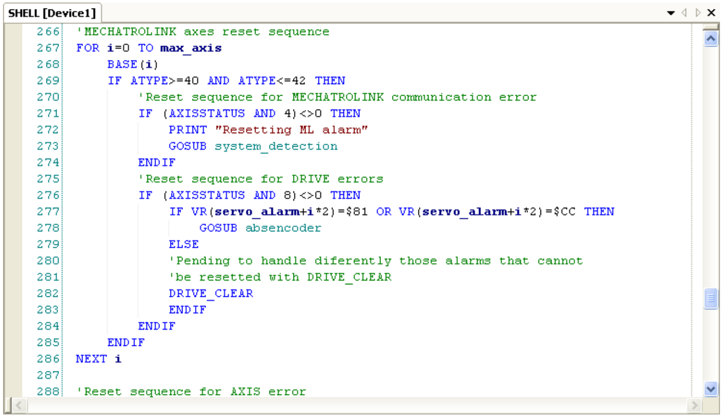
2-2 Code editor

The code editor is a text editor to write BASIC program code. It consists of the following areas:

- Margin bar
  This is the gray vertical bar at the left of the code editor. It contains the icons that indicate a bookmark or a breakpoint.
- Line number area
  The line number area is a column of white space right of the margin bar. It contains the line numbers.

Note If you click on a line number, the entire line is selected.

- Left margin indicator
  This is a dotted vertical line at the right side of the line number area. It indicates the left margin of the text area. Also, it indicates the parts of the program code that are built (with a green bar) or changed (with a yellow bar). Refer to section 2-2-5-2.
- Text area
  This is the area where you type the text.
- Right margin indicator
  This is a dotted vertical line at the right side of the text area. It indicates the right margin of the text area.



```
'MECHATROLINK axes reset sequence
FOR i=0 TO max_axis
    BASE(i)
    IF ATYPE>=40 AND ATYPE<=42 THEN
        'Reset sequence for MECHATROLINK communication error
        IF (AXISSTATUS AND 4)<>0 THEN
            PRINT "Resetting ML alarm"
            GOSUB system_detection
        ENDIF
        'Reset sequence for DRIVE errors
        IF (AXISSTATUS AND 8)<>0 THEN
            IF VR(servo_alarm+i*2)=$81 OR VR(servo_alarm+i*2)=$CC THEN
                GOSUB absencoder
            ELSE
            'Pending to handle diferently those alarms that cannot
            'be resetted with DRIVE_CLEAR
            DRIVE_CLEAR
            ENDIF
        ENDIF
    ENDIF
NEXT i

'Reset sequence for AXIS error
```

Code editor

The code editor has several features that make writing and editing program code very easy. These features are described in the following subsections.

2-2-1 Editing text

2-2-1-1 Standard operations

The code editor supports the standard text editor operations. The table below lists the standard operations. The second column lists the corresponding commands on the **Edit** menu.

Operation	Command
Undo or redo the last operation	**Undo**, **Redo**

Operation	Command
Cut, copy or paste the selected text	**Cut**, **Copy**, **Paste**
Delete the character right of the cursor or the selected text	**Delete**
Select all text in the code editor	**Select All**

2-2-1-2 Indenting text

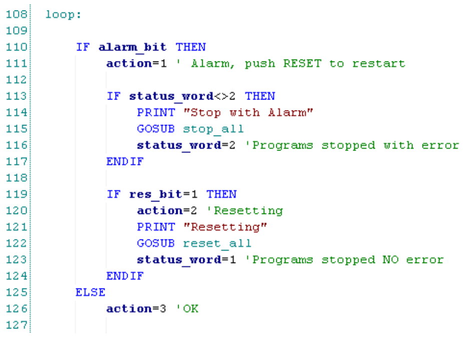
You can increase or decrease the indent of one or more lines of code with the **Increase Line Indent** and **Decrease Line Indent** commands under **Advanced** on the **Programs** menu. These commands apply to the selected lines in the code editor, or to the current line[1] if no text is selected.

Note If you increase a line indent, Trajexia Studio inserts 4 space characters to the beginning of the line. It does not insert a tab character.

You can also increase the indent with the TAB key, and decrease the indent with the SHIFT+TAB keys.

Note The TAB and SHIFT+TAB keys only increase or decrease the indent of the current line if the cursor is before the first non-space character on the line.

Trajexia Studio gives feedback about the indentation with gray vertical lines in the text area that connect lines of code with the same indentation.



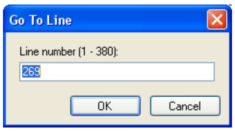
```
loop:

    IF alarm_bit THEN
        action=1 ' Alarm, push RESET to restart

        IF status_word<>2 THEN
            PRINT "Stop with Alarm"
            GOSUB stop_all
            status_word=2 'Programs stopped with error
        ENDIF

        IF res_bit=1 THEN
            action=2 'Resetting
            PRINT "Resetting"
            GOSUB reset_all
            status_word=1 'Programs stopped NO error
        ENDIF
    ELSE
        action=3 'OK

```

Indentation lines

2-2-2 Navigating

2-2-2-1 Line number

You can navigate directly to a specific line in the code editor. To do this, select **Go To** on the **Edit** menu, enter the line number to navigate to in the **Go To Line** window and click **OK**.



Go To Line

Line number (1 - 380): 269

Go To Line window

2-2-2-2 Bookmarks

A bookmark is a marker that specifies a line in the code editor. It is indicated with a bookmark icon ▭ in the margin bar of the code editor. Trajexia Studio supports navigation between bookmarked lines in the code editor.

[1] The current line is the line where the cursor is.

The table below lists the actions that Trajexia Studio supports for bookmarks in the code editor. The second column lists the corresponding commands on the **Programs** menu, under **Bookmarks**.

Action	Command
Set a bookmark on the current line	**Toggle Bookmark**
Clear the bookmark from the current line	**Toggle Bookmark**
Clear all bookmarks	**Clear Bookmarks**
Navigate to the next bookmark	**Next Bookmark**
Navigate to the previous bookmark	**Previous Bookmark**

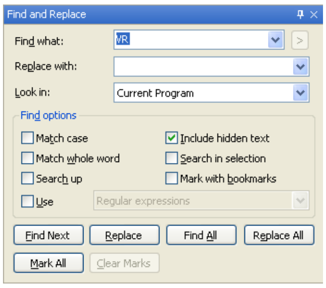
2-2-2-3 Breakpoints

Breakpoints are used to debug program code. For information about navigation between breakpoints in the program code and breakpoints in general, refer to section 5-2.

2-2-3 Searching text

2-2-3-1 Search and replace

Trajexia Studio supports search and replace operations for text in programs in code editor windows. The incremental search operation works directly in the code editor. It is described in the next section (2-2-3-2). The standard search and replace operations are accessible from the **Find and Replace** window.



Find what: VR
Replace with:
Look in: Current Program

Find options

□ Match case
☑ Include hidden text
□ Match whole word
□ Search in selection
□ Search up
□ Mark with bookmarks
□ Use Regular expressions

Find and Replace window

To show the **Find and Replace** window, select **Find and Replace** on the **Edit** menu, and then **Find and Replace**. The **Find and Replace** window is a docking window. For more information on docking windows, refer to section 2-4.

The 3 basic elements of a search and replace operation are the find string, the replace string and the search scope:

- The find string is entered in the **Find what** field. This field has a drop-down list with the recent find strings.
- The replace string is entered in the **Replace with** field. This field has a drop-down list with the recent replace strings. If you want to do a search without a replace, leave this field empty.
- The search scope is selected in the **Look in** list. Select **Current Program** to search only in the code editor window that is currently active. Select **All Programs** to search in all open code editor windows.

Note If you select **All Programs**, you cannot do a replace. The **Replace with** field and the **Find Next**, **Replace** and **Replace All** buttons are disabled. You can only list all occurrences of the find string in the **Find Results** window.

The table below lists the search actions that Trajexia Studio supports. The second column lists the corresponding buttons in the **Find and Replace** window.

Action	Button
Find the next occurrence of the find string	**Find Next**[1]
Replace the found occurrence of the find string with the replace string	**Replace**
Show a list of all occurrences of the find string in the **Find Results** window	**Find All**
Replace all occurrences of the find string in the current code editor window with the replace string	**Replace All**
Show a list of all occurrences of the find string in the **Find Results** window, and mark all occurrences of the find string with a green squiggle and optionally a bookmark	**Mark All**
Clear all green squiggles that mark the occurrences of the find string[2]	**Clear Marks**

1. You can also select **Find Next** under **Find and Replace** on the **Edit** menu. This menu is also available outside the **Find and Replace** window.
2. To clear all bookmarks, select **Bookmarks** on the **Edit** menu and then **Clear Bookmark**.

Trajexia Studio supports many search options. The table below lists the search options. The second column lists the corresponding check boxes in the **Find and Replace** window.

Search option	Check box
Search for occurrences of the find string that match the uppercase and lowercase characters in the find string	**Match case**
Search for whole words only and ignore occurrences of the find string within a word	**Match whole word**
Change the search direction to up from the cursor position up	**Search up**
Also search in hidden outlines in the text	**Search hidden text**
Search only in the selected text in the code editor	**Search in selection**
Mark all occurrences of the find string with a bookmark	**Mark with bookmarks**
Search for regular expressions or wildcards (see below)	**Use**

To search for a regular expression or a wildcard, select the **Use** check box and select **Regular expressions** or **Wildcards** in the list. If you search for a regular expression, certain characters in the find string represent notations for text patterns. If you search for wildcards, certain characters in the find string represent a class or a sequence of characters. Click the **>** button next to the **Find what** edit field to insert a regular expression character or a wildcard character.

2-2-3-2 Incremental search

The incremental search operation searches the text while you type the find string. Incremental search works directly in the text editor. You cannot do normal text editing in the code editor when you do an incremental search.

The table below lists the actions that Trajexia Studio supports for incremental search. The second column lists the keys to perform these actions.

Action	Keys
Start an incremental search	CTRL+I
Find a string	Type the string
Find the next occurrence of the find string	CTRL+I
Find the previous occurrence of the find string	CTRL+SHIFT+I
Cancel the incremental search	ESC

Note When you perform an incremental search, the mouse pointer changes to ⇩ or ⇧ (the arrow indicates the search direction).

2-2-4 Inserting and editing code

2-2-4-1 Commenting code

Comments are texts in the code editor that are not part of the program code. They are ignored when the program is built. Comments are useful when you want to add an explanation to your program code.

In BASIC, a comment is indicated by a comment mark. This is a single quotation mark ('). All the text from the comment mark to the end of the line is comment.

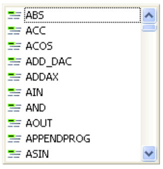
You can add or remove comment marks to one or more lines with the **Comment Selection** and **Uncomment Selection** commands under **Advanced** on the **Programs** menu. These commands apply to the selected lines in the code editor, or to the current line if no text is selected.

Note The command **Uncomment Selection** only removes the comment marks that are at the start of the line. Comment marks that are not at the start of the line are not removed.

2-2-4-2 Inserting a command

Trajexia Studio can insert a BASIC command in the program. This is useful if you do not know the correct name or syntax of a command.

You can insert BASIC command with the **Insert Command...** command on the **Programs** menu. This shows the list of available BASIC commands. This list supports incremental search, which means that you can type the first letters of the command, and the selection in the list scrolls to that command. To insert the selected command, double-click it or press ENTER or TAB.



ABS
ACC
ACOS
ADD_DAC
ADDAX
AIN
AND
AOUT
APPENDPROG
ASIN

Insert command

Note The shortcut key for **Insert Command...** is CTRL+SPACE.

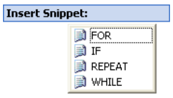
2-2-4-3 Inserting a code snippet

A code snippet is a piece of predefined code that you can insert in your program. The advantage of a code snippet is that you do not have to type the code manually, which saves you work and avoids syntax errors in the code.

Trajexia Studio supports code snippets for these conditional commands and loop commands:

Code snippet	BASIC command
FOR	FOR..TO..STEP..NEXT
IF	IF..THEN..ELSEIF..THEN..ELSE..ENDIF
REPEAT	REPEAT..UNTIL
WHILE	WHILE..WEND

You can insert a code snippet with the **Insert Code Snippet...** command on the **Programs** menu. This shows the list of available code snippets. To insert the selected code snippet, double-click it or press ENTER or TAB.



Insert snippet

A code snippet consists of BASIC keywords and code snippet fields. The code snippet fields are placeholders for the expressions, variables and statements that must be inserted to make the code syntactically correct. For example, the REPEAT code snippet inserts the following code:

```
REPEAT
  statement
UNTIL expression
```

In this code snippet, **statement** and **expression** are code snippet fields. They are placeholders for one or more BASIC statements and a BASIC expression respectively. A code snippet field shows in a specific colour, and it has an associated tooltip that shows if the mouse pointer hovers over the code snippet field.

2-2-4-4 Generating variable declarations

Trajexia Studio requires that all local variables in a program are declared before they are used. If a local variable is not declared, Trajexia Studio generates a warning when the program is built.

Motion Perfect 2 and the Trajexia devices do not require that local variables are declared. Thus, if you import a Motion Perfect 2 project in Trajexia Studio, or if you transfer a program from a Trajexia device to Trajexia Studio, you can have a program that has many undeclared local variables.

Trajexia Studio has a function that automatically adds a declaration for each local variable to a program. You can choose this function with the **Auto Variable Update** command in the context menu of the code editor. This adds the declaration of all local variables to a collapsible code section in the code editor.

Note BASIC does not have variable types. Thus, to declare a variable in a BASIC program you only have to initialise it, that is, assign it a value. This must be a numeric value, because all BASIC variables are numeric.
To declare the variable **var**, Trajexia Studio adds the line **var = 0** to the program.

2-2-4-5 Generating SHELL code

Trajexia Studio can generate program code for the assignment of axis parameters, drive parameters, Table data, VR variables and CAM table data in the SHELL program. For more information, refer to section 3-4-5.

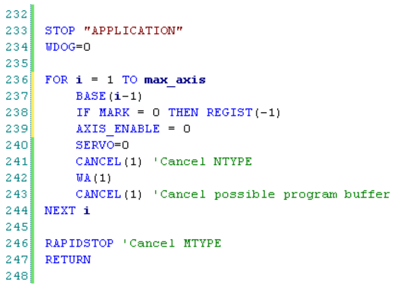
2-2-5 Visual feedback

2-2-5-1 Right margin

The right margin indicator indicates the right margin of the text area. The text area has a width of 80 characters. You can enter lines in the text area that are longer than 80 characters. However, this generates an error when you build this program, because BASIC code lines cannot be longer than 80 characters.

2-2-5-2 Built code and changed code

Trajexia Studio gives feedback in the code editor about which code is built, and which code is changed after the last build. Code that is built is indicated with a green vertical line. Code that is changed after the last build is indicated with a yellow vertical line.



```

STOP "APPLICATION"
WDOG=0

FOR i = 1 TO max_axis
    BASE(i-1)
    IF MARK = 0 THEN REGIST(-1)
    AXIS_ENABLE = 0
    SERVO=0
    CANCEL(1) 'Cancel NTYPE
    WA(1)
    CANCEL(1) 'Cancel possible program buffer
NEXT i

RAPIDSTOP 'Cancel MTYPE
RETURN

```

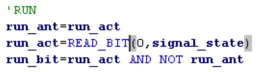
Green and yellow lines

2-2-5-3 Matching parentheses

Trajexia Studio recognises pairs of parentheses that match. This gives you feedback on misplaced or missing parentheses, which cause syntax errors.

If 2 parentheses match, Trajexia Studio highlights the 2 parentheses in the following cases:

- If the cursor is left of the opening parenthesis
- If the cursor is right of the closing parenthesis



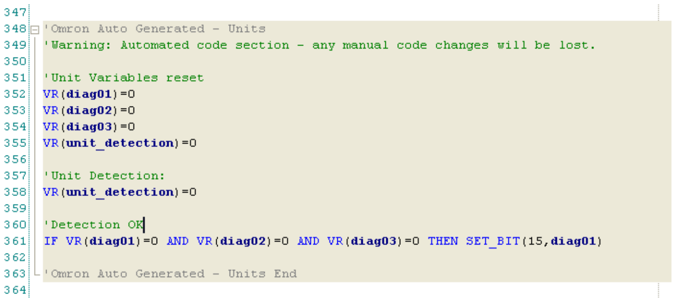
```
'RUN
run_ant=run_act
run_act=READ_BIT(0,signal_state)
run_bit=run_act AND NOT run_ant
```

Matching parentheses

Note If the cursor is left of an opening parenthesis or right of a closing parenthesis, you can press CTRL+] to navigate to the matching parenthesis.

2-2-5-4 Read-only sections

Trajexia Studio can generate program code. Some sections of generated code are read-only and cannot be edited. The read-only sections of code have a gray background colour.



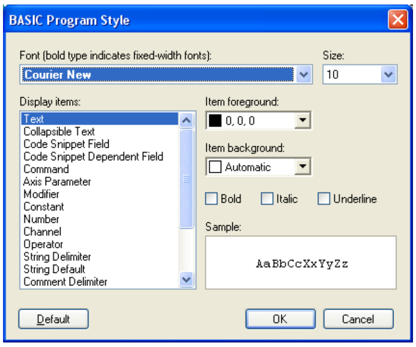
```
'Omron Auto Generated - Units
'Warning: Automated code section - any manual code changes will be lost.

'Unit Variables reset
VR(diag01)=0
VR(diag02)=0
VR(diag03)=0
VR(unit_detection)=0

'Unit Detection:
VR(unit_detection)=0

'Detection OK
IF VR(diag01)=0 AND VR(diag02)=0 AND VR(diag03)=0 THEN SET_BIT(15,diag01)

'Omron Auto Generated - Units End
```

Read-only section

Note You can copy text from a read-only section.

2-2-5-5 Syntax colouring

The BASIC program code consists of syntactical elements. The code editor recognises these syntactical elements when you enter the code.
You can change the colours and fonts of the syntactical elements in the **BASIC Program Style** window. To open this window, select **BASIC Program Style** under **Options** on the **Tools** menu.



BASIC Program Style

Font (bold type indicates fixed-width fonts): Courier New

Size: 10

Display items: Text

Item foreground: 0, 0, 0

Item background: Automatic

□ Bold □ Italic □ Underline

Sample: AaBbCcXxYyZz

BASIC Program Style window

In the **BASIC Program Style** window, you can set the font, the font size, the foreground and background colours and the character style (bold, italic and underlined) for each syntactical element, which are called items in this window.

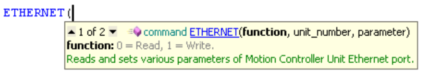
Note To restore the default colours and fonts of the syntactical elements, click the **Default** button in the **BASIC Program Style** window.

2-2-5-6 Command information

Trajexia Studio can show the following information about a BASIC command and the arguments when you type the BASIC command in the code editor:

- The number of overloads of the command (if the command has overloads)
- The arguments
- The possible values of an argument
- A description of the command

The information shows in a tooltip. The command in the tooltip is a hyperlink. If you click this hyperlink, the associated help information for this BASIC command shows.
The next argument that you must insert shows in bold.



```
ETHERNET(
▲ 1 of 2 ▼  command ETHERNET(function, unit_number, parameter)
function: 0 = Read, 1 = Write.
Reads and sets various parameters of Motion Controller Unit Ethernet port.
```

Command information

The table below lists the actions that Trajexia Studio supports for argument information. The second column lists the steps that you must perform to do these actions.

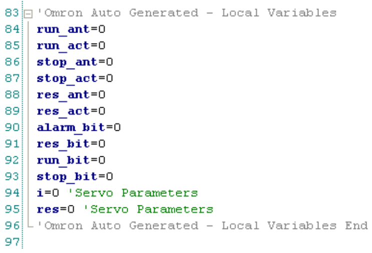
Action	Steps
View the argument information for a command	Type the command, and type the opening parenthesis, or Put the cursor between the argument parentheses, and press CTRL+SHIFT+SPACE
View the argument information for another overload of the command	Press the ARROW UP or ARROW DOWN key to scroll through the list of overloads
Show information about the next argument in the argument list	Type a value for the current argument, and type a comma
Close the argument information tooltip	Press ESC, or complete the command with all arguments and type the closing parenthesis

2-2-6 Viewing

2-2-6-1 Outlining

Trajexia Studio supports outlining of the text in the code editor. An outline is a section of the text that can be visible or hidden in the code editor.

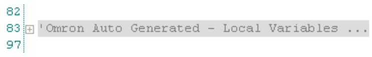
- A visible outline is indicated with a minus sign in a small square to the right of the left margin indicator, and a vertical line from the small square up to the last line of the outline.



```
⊟'Omron Auto Generated - Local Variables
 run_ant=0
 run_act=0
 stop_ant=0
 stop_act=0
 res_ant=0
 res_act=0
 alarm_bit=0
 res_bit=0
 run_bit=0
 stop_bit=0
 i=0 'Servo Parameters
 res=0 'Servo Parameters
└'Omron Auto Generated - Local Variables End
```

Visible outline

- A hidden outline is indicated with a plus sign in a small square to the right of the left margin indicator. Instead of the outlined text, an outline placeholder shows. It consists of the first line of the outlined text followed by an ellipsis.



```
⊞'Omron Auto Generated - Local Variables ...
```

Hidden outline

Note Outlined text is not deleted when you hide it, it is only invisible in the code editor.

Note If you hover the mouse pointer over a hidden outline, a tooltip with the hidden text shows. If the hidden text is very large, only the first part of the hidden text shows in the tooltip.

All outlines are automatically defined by Trajexia Studio. You cannot define your own outlines, or clear existing outlines.

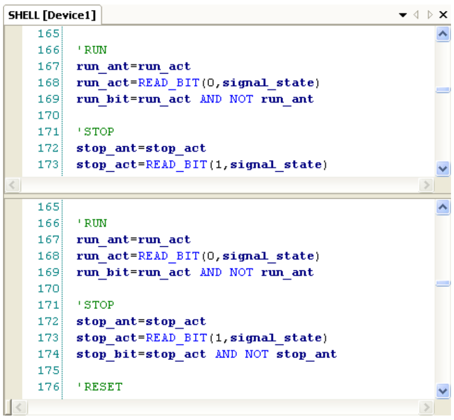
The table below lists the actions that Trajexia Studio supports to show or hide outlines. The second column lists the corresponding commands on the **Programs** menu, under **Outlining**.

Action	Command
Show a hidden outline	**Toggle Outlining** (Or click the plus sign in the small square of the outline)
Hide a visible outline	**Toggle Outlining** (Or click the minus sign in the small square of the outline)
Show all outlines	**Expand All Outlining**
Hide all outlines	**Collapse All Outlining**

2-2-6-2 Splitting the window

You can split the code editor window. If you split the window, you can look at different parts of the same window simultaneously. This is useful if you want to compare different parts of the code in the same program.

You can split the window horizontally and vertically. To split the window, drag or double-click the horizontal or the vertical splitter. The horizontal splitter is located left of the horizontal scroll bar. The vertical splitter is located above the vertical scroll bar.



```

'RUN
run_ant=run_act
run_act=READ_BIT(0,signal_state)
run_bit=run_act AND NOT run_ant

'STOP
stop_ant=stop_act
stop_act=READ_BIT(1,signal_state)
```

```

'RUN
run_ant=run_act
run_act=READ_BIT(0,signal_state)
run_bit=run_act AND NOT run_ant

'STOP
stop_ant=stop_act
stop_act=READ_BIT(1,signal_state)
stop_bit=stop_act AND NOT stop_ant

'RESET
```

Split code editor

2-2-7 Shortcut keys

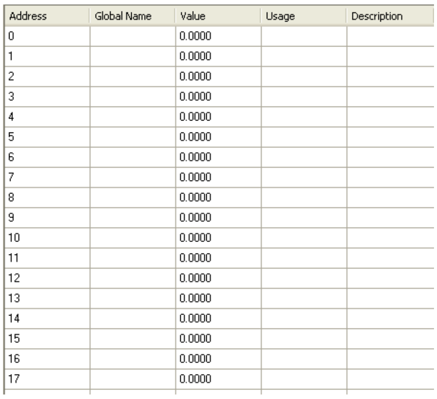
The code editor supports many shortcut keys to edit the text and navigate through the text. Refer to appendix A for an overview of all shortcut keys in the code editor.

2-3 Tabular views

A tabular view is a user interface element that shows a table of values arranged in rows and columns. Trajexia Studio uses tabular views to show the following items:

- Constants
- I/O memory

- CAM tables
- VR memory
- Table memory
- Watches
- Memory areas in the **Memory Manager**



Address	Global Name	Value	Usage	Description
0		0.0000		
1		0.0000		
2		0.0000		
3		0.0000		
4		0.0000		
5		0.0000		
6		0.0000		
7		0.0000		
8		0.0000		
9		0.0000		
10		0.0000		
11		0.0000		
12		0.0000		
13		0.0000		
14		0.0000		
15		0.0000		
16		0.0000		
17		0.0000		

Tabular view

The tabular views have the same layout, and they have a set of features, which are described in this section. Not all tabular views have the same features. For each feature, we list to which tabular views it applies.

2-3-1 Change a value in a cell

There are 2 ways to change the value in a cell: you can edit a cell, or paste on a cell.

Edit cell

To edit a standard cell, perform the following steps:

1 Select the cell that you want to edit.
2 Click in the cell, or press F2.
3 Type a new value in the cell.
4 Press ENTER.

Note If you press ESC instead of ENTER in this step, the new value is discarded and the old value restored.

To edit a drop-down cell, perform the following steps:

1 Select the cell that you want to edit.
2 Click the drop-down arrow in the cell.
3 Select a new value in the drop-down list.

Paste cells

To paste one or more table cells, perform the following steps:

1 Copy one or more cells to the clipboard.
2 Select a cell in the tabular view where you want to paste the cell(s) on the clipboard.
3 On the **Edit** menu, select **Paste**.

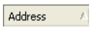
The table below lists the columns in which the cells can be edited.

Tabular view	Columns
Constants	**Name** **Value**
I/O	**Description**

Tabular view	Columns
CAM table	**Value**
VR memory	**Global Name** **Usage**
Table memory	All columns except **Address**

2-3-2 Sorting a column

Some columns can be sorted. If you click on the column header, the column is sorted in ascending order. A triangle shows at the right side of the column header that indicates the sort order.



Column header with ascending sort order triangle

If you click on the header of a column that is sorted in ascending order, the column is sorted in descending order.

Note Not all columns can be sorted. If you click on the header of a column that cannot be sorted, the column is selected.

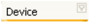
The table below lists the columns that can be sorted.

Tabular view	Columns
Constants	**Name** **Value**
I/O memory	**Address** **Type** **Description** **Device**
VR memory	**Address** **Global Name** **Usage** **Value**
Watch window	All

2-3-3 Filtering rows

Some columns can have a filter. If you set this filter, the table only shows the rows where the value of the cell in the filtered column is equal to the column filter.

If a column can have a filter, a filter button shows when the mouse pointer is over the column header.



Column header with filter button

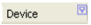
Set filter To set a filter, perform the following steps:

1 Click the filter button on the column header.

2 Select a value from the list of filter values.
Some filters have special filter values that can be selected. The table below lists these special filter values.

Special filter value	Description
(Custom)	A custom filter (see the procedure **Set custom filter** below)
(Blanks)	Blank characters (spaces and tabs)
(Non blanks)	Non-blank characters (all characters except blank characters)
(All)	All values (used to clear a filter, see the procedure **Clear filter** below)

If a column has a filter set, the filter button on the column header shows permanently, in purple.



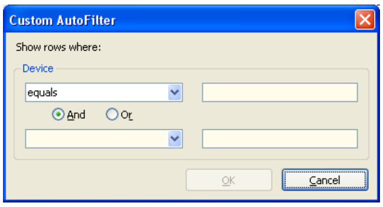
Filter button in purple

Set custom filter

To set a custom filter, perform the following steps:

1 Click the filter button on the column header.

2 Select the value **(Custom)** from the list of filter values.



Custom AutoFilter window

3 In the **Custom AutoFilter** window, enter a relation and a value for the filter.

4 Optionally, enter another relation and value for the filter, and a boolean relation (AND or OR) between the first relation and value and the second relation and value.

5 Click **OK**.

Clear filter

To clear a filter, perform the following steps:

1 Click the filter button on the column header.

2 Select **(All)** from the list of filter values.

The table below lists the columns that can have a filter.

Tabular view	Columns
I/O memory	**Type** **Device**
VR memory	**Usage**
Watch window	**Type** **Device** **Group**

2-3-4 Importing and exporting data

The data in some tables can be imported from a file and exported to a file. To do this, right-click in the tabular view, and select **Import *data*...**, or select **Export *data*...**

The file to import from or export to is in CSV format. If the size and values of the data in the CSV file do not match the size and value formats of the data in the tabular view, Trajexia Studio shows an error message.

The table below lists the tabular views of which the data can be imported and exported.

Tabular view
Constants
CAM table
VR memory
Table memory

2-4 Docking windows

The docking windows are windows that show information about the solution, the solution items and the contents of the document windows, and they provide tools and options to support working with Trajexia Studio.
Docking windows do not behave like document windows, but they have different display characteristics. These are described in the next section.

2-4-1 Arranging docking windows

The docking windows are listed on the **View** menu, under **Docking Windows**. A check mark on the menu item indicates that the corresponding docking window is visible. You can hide or show a docking window with these menu commands.
In addition, Trajexia Studio supports the following layout operations on docking windows:

- Show or hide the docking window automatically
- Dock a docking window against an edge of the IDE, or float a docking window over the IDE
- Group docking windows together, or detach grouped docking windows

2-4-1-1 Showing or hiding a docking window automatically

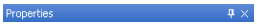
Trajexia Studio supports the auto-hide feature for docking windows. If this feature is on, the window is minimised to a tab at the edge of the IDE if it does not have the focus. The window shows when the mouse pointer is over the tab or in the window.

Note The auto-hide feature only works if the window is docked.

Set auto-hide

To set the auto-hide feature of a docking window, perform the following steps:

1 If the window floats, dock it.
2 Click the pushpin in the title bar.



Pushpin

Clear auto-hide

To clear the auto-hide feature of a docking window, perform the following steps:

1 If the window does not show, click on the tab at the edge of the IDE to make it visible.
2 Click the pushpin in the title bar.

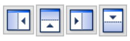
Pushpin

2-4-1-2 Docking or floating a docking window

Dock window

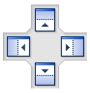
To dock a docking window against an edge of the IDE, perform the following steps:

1 Drag the window away from its current location.
Four single-arrow guide shows, at the 4 edges of the IDE.



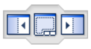
Four single-arrow guides

If the mouse pointer is over a document window, an additional four-arrow guide shows in the center of the document window. The 4 arrows point to the 4 edges of the document window.



Four-arrow guide for document windows

If the mouse pointer is over a docking window, an additional two-arrow guide shows in the center of the docking window. The 2 arrows point to the 2 sides of the docking window.



Two-arrow guide for docking windows

2 Drag the window to the arrow that points to the edge where you want to dock it.
An outline of the window location shows in the IDE at the proposed location.

3 Release the mouse button.
The window is docked at the proposed location.

Note If you double-click in the title bar of a floating docking window, the window is docked at its previous docking location.

Note If the auto-hide feature is set, you cannot dock the window.

Float window

To float a docking window over the IDE, perform the following steps:

1 Drag the window to the desired location.
2 Make sure that the mouse pointer is not over an arrow in an arrow guide.
3 Release the mouse button.

Note If you double-click in the title bar of a docked docking window, the window floats at its previous location.

Note A window that floats always shows on top of the IDE and the document windows.

Note If the auto-hide feature is set, you cannot float the window.

2-4-1-3 Grouping or detaching docking windows

Group windows

To group a docking window together with other docking windows, perform the following steps:

1 Drag the window to another docking window with which you want to group it.

The two-arrow guide shows in the center of the docking window. Between the 2 arrows is an area that represents the grouping option.

Two-arrow guide with grouping option area between the 2 arrows

2 Drag the window to the grouping option area.

3 Release the mouse button.
The window is added as a tab to the docking window.

Note If the auto-hide feature is set, you cannot group the window.

Detach window

To detach a docking window from a group of docking windows, perform the following steps:

1 Drag the tab that represents the window away from the docking window.

2 To dock or float the window, perform the corresponding docking or floating procedure described above.

2-4-2 All docking windows

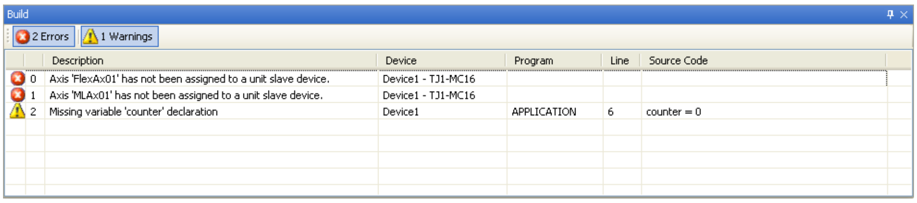
Trajexia Studio has 8 docking windows:

- **Build**
- **Error Log**
- **Find Results**
- **Find and Replace**
- **Output**
- **Solution Explorer**
- **Properties**
- **Watch**

This section describes the docking windows.

2-4-2-1 Build window

The **Build** window lists the errors and warnings that Trajexia Studio generates when you build a program, a device or the solution. The difference between an error and a warning is that an error is more severe. You can synchronise a device with build warnings and build errors, but is recommended not to synchronise a device with build errors. For more information on synchronisation, refer to section 4-5.



		Description	Device	Program	Line	Source Code
⊗	0	Axis 'FlexAx01' has not been assigned to a unit slave device.	Device1 - TJ1-MC16			
⊗	1	Axis 'MLAx01' has not been assigned to a unit slave device.	Device1 - TJ1-MC16			
⚠	2	Missing variable 'counter' declaration	Device1	APPLICATION	6	counter = 0

Build window

The **Build** window has a list that shows the following columns:

Column	Description
First column (unlabelled)	An icon that indicates an error ⊗ or a warning ⚠
Second column (unlabelled)	The order in which the errors and warnings are generated
Description	An explanation of the error or warning
Device	The device where the error or warning occurred
Program	The program where the error or warning occurred (only applicable to syntax errors and warnings in program code)
Line	The line number in the program where the error or warning occurred (only applicable to syntax errors and warnings in program code)
Source Code	The code at the line number in the program where the error or warning occurred (only applicable to syntax errors and warnings in program code)

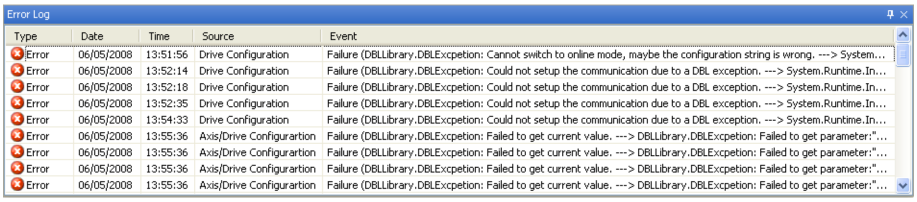
The table below lists the actions that the **Build** window supports. The second column lists the steps that you must perform to do these actions.

Action	Steps
Navigate to the syntax error or warning in the code editor	Double-click the row in the list
Navigate to the previous or next syntax error or warning in the code editor	Right-click the row in the list and select **Previous Error** or **Next Error**
Show or hide a column in the list	RIght-click on the list, select **Show Columns** and then the column you want to show or hide
Copy the contents of the list to the clipboard	Right-click on the list and select **Copy**
Filter the list to show or hide errors or warnings	Click the **Errors** or **Warnings** toggle button to set or clear the filter

2-4-2-2 Error Log window

The **Error Log** window lists all errors that occur in the online device. These errors are communication errors, synchronisation errors, or other errors from the device.

Note When an error occurs, an indicator flashes on the status bar. If you double-click this indicator, the **Error Log** window shows.



Type	Date	Time	Source	Event
⊗ Error	06/05/2008	13:51:56	Drive Configuration	Failure (DBLLibrary.DBLExcpetion: Cannot switch to online mode, maybe the configuration string is wrong. ---> System...
⊗ Error	06/05/2008	13:52:14	Drive Configuration	Failure (DBLLibrary.DBLExcpetion: Could not setup the communication due to a DBL exception. ---> System.Runtime.In...
⊗ Error	06/05/2008	13:52:18	Drive Configuration	Failure (DBLLibrary.DBLExcpetion: Could not setup the communication due to a DBL exception. ---> System.Runtime.In...
⊗ Error	06/05/2008	13:52:35	Drive Configuration	Failure (DBLLibrary.DBLExcpetion: Could not setup the communication due to a DBL exception. ---> System.Runtime.In...
⊗ Error	06/05/2008	13:54:33	Drive Configuration	Failure (DBLLibrary.DBLExcpetion: Could not setup the communication due to a DBL exception. ---> System.Runtime.In...
⊗ Error	06/05/2008	13:55:36	Axis/Drive Configurartion	Failure (DBLLibrary.DBLExcpetion: Failed to get current value. ---> DBLLibrary.DBLExcpetion: Failed to get parameter:"...
⊗ Error	06/05/2008	13:55:36	Axis/Drive Configurartion	Failure (DBLLibrary.DBLExcpetion: Failed to get current value. ---> DBLLibrary.DBLExcpetion: Failed to get parameter:"...
⊗ Error	06/05/2008	13:55:36	Axis/Drive Configurartion	Failure (DBLLibrary.DBLExcpetion: Failed to get current value. ---> DBLLibrary.DBLExcpetion: Failed to get parameter:"...
⊗ Error	06/05/2008	13:55:36	Axis/Drive Configurartion	Failure (DBLLibrary.DBLExcpetion: Failed to get current value. ---> DBLLibrary.DBLExcpetion: Failed to get parameter:"...

Error Log window

The **Error Log** window has a list that shows the following columns:

Column	Description
Type	Error or warning
Date	The date that the error occurred
Time	The time that the error occurred
Source	The operation or the service that generated the error
Event	The error message from the source

Note The errors that show in the **Error Log** window are also appended to a file called ErrorLog.log. When ErrorLog.log is larger than 100 kB, it is copied to a backup file called ErrorLogBackup.log (which overwrites the previous version of ErrorLogBackup.log if it exists), and the contents of ErrorLog.log is cleared.

The table below lists the actions that the **Error Log** window supports. The second column lists the steps that you must perform to do these actions.

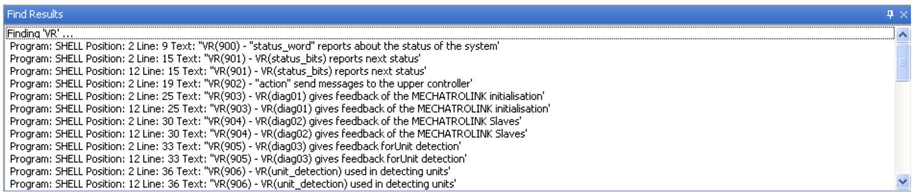
Action	Steps
Show or hide a column in the list	Right-click on the list, select **Show Columns** and then the column you want to show or hide
Copy the contents of the list to the clipboard	Right-click on the list and select **Copy**
Remove all errors from the list and clear the contents of the file ErrorLog.log	Right-click on the list and select **Clear**

Note When you close Trajexia Studio:

- The contents of the **Error Log** window is cleared
- The contents of ErrorLog.log is not cleared

2-4-2-3 Find Results window

The **Find Results** window lists the results of the last **Find All** or **Mark All** action in the **Find and Replace** window (refer to section 2-2-3-1).



```
Finding 'VR' ...
 Program: SHELL Position: 2 Line: 9 Text: "VR(900) - "status_word" reports about the status of the system'
 Program: SHELL Position: 2 Line: 15 Text: "VR(901) - VR(status_bits) reports next status'
 Program: SHELL Position: 12 Line: 15 Text: "VR(901) - VR(status_bits) reports next status'
 Program: SHELL Position: 2 Line: 19 Text: "VR(902) - "action" send messages to the upper controller'
 Program: SHELL Position: 2 Line: 25 Text: "VR(903) - VR(diag01) gives feedback of the MECHATROLINK initialisation'
 Program: SHELL Position: 12 Line: 25 Text: "VR(903) - VR(diag01) gives feedback of the MECHATROLINK initialisation'
 Program: SHELL Position: 2 Line: 30 Text: "VR(904) - VR(diag02) gives feedback of the MECHATROLINK Slaves'
 Program: SHELL Position: 12 Line: 30 Text: "VR(904) - VR(diag02) gives feedback of the MECHATROLINK Slaves'
 Program: SHELL Position: 2 Line: 33 Text: "VR(905) - VR(diag03) gives feedback forUnit detection'
 Program: SHELL Position: 12 Line: 33 Text: "VR(905) - VR(diag03) gives feedback forUnit detection'
 Program: SHELL Position: 2 Line: 36 Text: "VR(906) - VR(unit_detection) used in detecting units'
 Program: SHELL Position: 12 Line: 36 Text: "VR(906) - VR(unit_detection) used in detecting units'
```

Find Results window

The table below lists the actions that the **Find Results** window supports. The second column lists the steps that you must perform to do these actions.

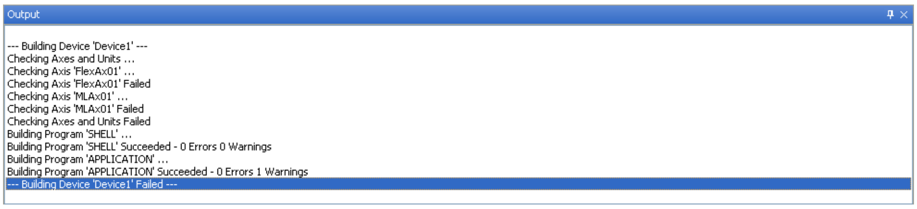
Action	Steps
Navigate to the find result in the code editor window	Double-click the line in the window
Copy the contents of the list to the clipboard	Right-click on the list and select **Copy**
Remove all find results from the list	Right-click on the list and select **Clear**

2-4-2-4 Find and Replace window

The **Find and Replace** window provides tools to search for text in the code editor windows and replace it. Refer to section 2-2-3-1 for information on how to search and replace text.

2-4-2-5 Output window

The **Output** window shows the status information when Trajexia Studio performs a process. For example, when Trajexia Studio builds a program, a device or a solution, the status of the build process shows in the **Output** window.



```
--- Building Device 'Device1' ---
Checking Axes and Units ...
Checking Axis 'FlexAx01' ...
Checking Axis 'FlexAx01' Failed
Checking Axis 'MLAx01' ...
Checking Axis 'MLAx01' Failed
Checking Axes and Units Failed
Building Program 'SHELL' ...
Building Program 'SHELL' Succeeded - 0 Errors 0 Warnings
Building Program 'APPLICATION' ...
Building Program 'APPLICATION' Succeeded - 0 Errors 1 Warnings
--- Building Device 'Device1' Failed ---
```

Output window

The table below lists the actions that the **Output** window supports. The second column lists the steps that you must perform to do these actions.

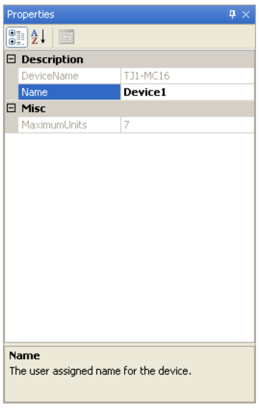
Action	Steps
Copy the contents of the list to the clipboard	Right-click on the list and select **Copy**
Remove all lines from the list	Right-click on the list and select **Clear**

2-4-2-6 Solution Explorer window

The **Solution Explorer** window shows the structure of the solution. For more information on the solution and the **Solution Explorer**, refer to section 3.

2-4-2-7 Properties window

The **Properties** window shows the design-time properties of the objects selected in the document windows or in the **Solution Explorer**. The properties can also be changed in the **Properties** window.



Description	
DeviceName	TJ1-MC16
Name	**Device1**
Misc	
MaximumUnits	7

Name
The user assigned name for the device.

Properties window

The **Properties** window has a list with 2 columns: the property name, and the value of the property. The window supports 2 views:

- Categorised view
  This view lists the properties and their values grouped together by category. Categories can be collapsed and expanded, to reduce or increase the number of visible properties. To collapse or expand a category, click the minus or plus sign left of the category description.
  The categories are listed alphabetically. The properties in a category are also listed alphabetically
  To show the properties in categorised view, click the ☰ button.
- Alphabetical view
  This view lists all properties in alphabetical order.
  To show the properties in alphabetical view, click the A↓Z button.

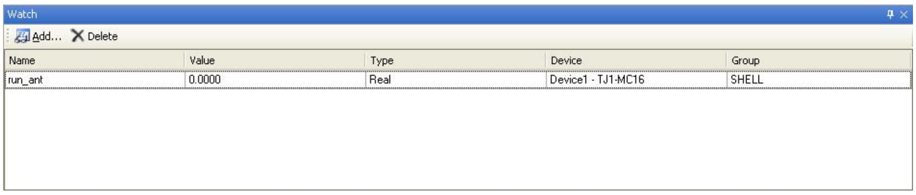
The **Properties** window has different types of edit fields, depending on the value type and the list of possible values of the property. The table below lists the types of edit fields and the ways to enter a value in these edit fields.

Edit field	How to enter a value in the edit field
Standard edit field	Type the value in the edit field
Drop-down combo box	• Type the value in the edit field • Open the drop-down list and select a value • Double-click in the edit field to select the next value from the list of values
Edit field with ellipsis button	• Type the value in the edit field • Click the ellipsis button to open a dialog box in which you can select a value to enter in the edit field

Properties that show in gray are read-only and cannot be edited.

2-4-2-8 Watch window

A watch is a variable and its value. You can define watches to view the values of the variables when a program is running. The **Watch** window lists the defined watches.



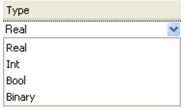
Name	Value	Type	Device	Group
run_ant	0.0000	Real	Device1 - TJ1-MC16	SHELL

Watch window

The **Watch** window has 2 buttons, **Add...** and **Delete**, to add and delete watches from the window, and it has a tabular view with the following columns:

Column	Description
Name	The name of the watched variable
Value	The value of the watched variable
Type	The numeric format in which the value in the **Value** column shows
Device	The device where the watched variable is defined
Group	The program where the watched variable is defined and the process ID of the running program (if applicable)

The cells in the **Type** column have a drop-down list, where you can select the numeric format of the value in the **Value** column.



Type

Real

Real
Int
Bool
Binary

Type column in **Watch** window

The table below lists the values in the **Type** column and their corresponding numeric formats.

Type value	Numeric format
Real	Floating point (4 decimals)
Int	Integer
Bool	Boolean (0 equals **False**, all other values equal **True**)
Binary	Binary representation of 24-bit integer values[1]

[1] The **Binary** type does not show all 32 bits used in IEEE floating-point values.

The list in the Watch window is a tabular view. Refer to section 2-3 for more information on the specific features of tabular views. For more information on watches, refer to section 5-4.

2-4-3 Resetting the layout

The docking window layout can be restored to the default settings.

Reset layout

To reset the window layout, perform the following steps:

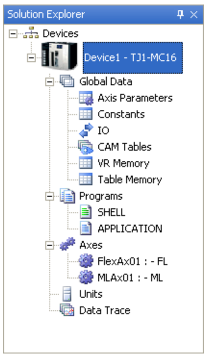
1 On the **Tools** menu, select **Options**.
2 Select **Reset Docking Window Layout**.

SECTION 3
Solutions

3-1 Solution items

To manage the items in your motion application, Trajexia Studio provides a container: the solution. The solution contains the devices in your motion application.

The **Solution Explorer** is the interface to view and manage the solution. The **Solution Explorer** shows a tree view of the items in the solution.



Solution Explorer

Note The items under a device in the solution can show in black, red and green. The meaning of these colours is explained in section 4-5-2.

3-1-1 Managing solutions

Trajexia Studio always contains one solution. Trajexia Studio cannot contain more than one solution, and it cannot contain zero solutions.

When you start Trajexia Studio, it automatically creates an empty solution. An empty solution is a solution without devices. When you open an existing solution, Trajexia Studio closes the current solution.

New solution To close the current solution and create a new solution, perform the following steps:

1 On the **File** menu, select **New**.

Open solution To open a solution, perform the following steps:

1 On the **File** menu, select **Open**.
2 In the **Open** dialog box, select the solution file.
3 Click **Open**.

Save solution To save an existing solution, perform the following steps:

1 On the **File** menu, select **Save**.

To save a new solution or a copy of an existing solution, perform the following steps:

1 On the **File** menu, select **Save As...**
2 In the **Save As** dialog box, specify the solution file name and the location to save the file.
3 Click **Save**.

Note A solution file has the extension .tjs. The contents of the file is in XML format.

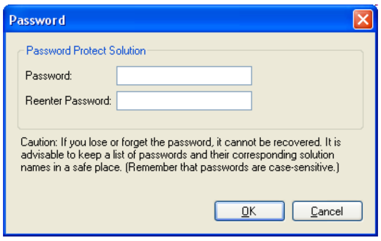
Protect solution

You can protect the solution with a password. A protected solution is encrypted when it is saved. You can only open a protected solution if you have the password.

To protect the solution, perform the following steps:

1 On the **Tools** menu, select **Solution Password...**

2 In the **Password** window, type the password in the **Password** text field.



Password

Password Protect Solution

Password:

Reenter Password:

Caution: If you lose or forget the password, it cannot be recovered. It is advisable to keep a list of passwords and their corresponding solution names in a safe place. (Remember that passwords are case-sensitive.)

Password window

3 Type the password again in the **Reenter Password** text field to confirm the password.

4 Click **OK**.

Caution

If you forget the password, you cannot open the solution.

Note Passwords are case-sensitive.

Note To clear the password of a protected solution, set the password to the empty string in the **Password** window.

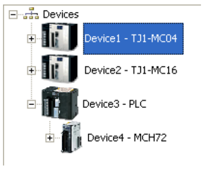
3-2 Devices

A solution contains devices. A device can contain data, BASIC programs, units and axes. Trajexia Studio supports 2 kinds of devices:

- Motion devices
- Host devices, which contain motion devices

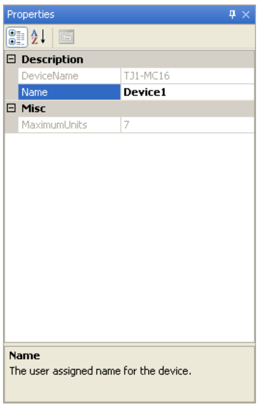
Trajexia Studio supports the following devices:

Trajexia Studio name	Device	Motion device/host device
TJ1-MC04	Trajexia device with 4 axes	Motion device
TJ1-MC16	Trajexia device with 16 axes	Motion device
PLC	PLC (CJ1-series, CJ2-series or CP1H-series)	Host device for the MCH72
MCH72	CJ1W-MCH72 device	Motion device



Motion devices (MCH72, TJ1-MC04 and TJ1-MC16) and host devices (PLC)

A device can be renamed (refer to section 3-8). The name of a device corresponds to the property **Name**, and not to the property **DeviceName**. The property **DeviceName** contains the identifier of the device (such as PLC and TJ1-MC16). **DeviceName** is read-only.



Description

DeviceName	TJ1-MC16
Name	Device1

Misc

MaximumUnits	7

Name
The user assigned name for the device.

DeviceName and **Name** properties of a device

3-2-1 Adding a device

There are 3 ways to add a device to the solution:

- Add a new device
- Import a Motion Perfect 2 project
- Import a resource

New devices are always added after the existing devices in the solution.

Add new device

To add a new device to the solution, perform the following steps:
1 On the **Insert** menu, select **Add *device-to-add***.

Note If you select **Add MCH72** on the **Insert** menu, a PLC host device and an MCH72 motion device under the PLC device are added to the solution.

To add a new motion device to a host device, perform the following steps:
1 In the **Solution Explorer**, select a host device.
2 On the **Insert** menu, select **Add *motion-device-to-add***.

Import Motion Perfect 2 project

Trajexia Studio can import Motion Perfect 2 projects into the solution. A Motion Perfect 2 project contains the representation of a device and its programs, parameters and data. Trajexia Studio can convert a Motion Perfect 2 project to a device.
To import a Motion Perfect 2 project as a device in the solution, perform the following steps:
1 On the **File** menu, select **Import Motion Perfect Project...**
2 In the **Open** dialog box, select the Motion Perfect 2 project file.
3 Click **Open**.

Load resource

A resource is an XML file that contains a saved Trajexia Studio device.
To load a resource, perform the following steps:

1. In the **Solution Explorer**, right-click the **Devices** item.
2. Select **Import Device...**
3. In the **Open** dialog box, select the resource file.
4. Click **Open**.

Note If the resource file does not contain a valid Trajexia Studio device, an error message shows.

To load a motion device from a resource and add it to a host device, perform the following steps.

1. In the **Solution Explorer**, right-click a host device.
2. Select **Import Device...**
3. In the **Open** dialog box, select the resource file.
4. Click **Open**.

Note If the resource file does not contain a motion device that can be added to the selected host device, an error message shows.

Note The device in a resource file has a name. Trajexia Studio ignores this name when it loads the device from the resource file. Instead, it gives the device a new name.

3-2-2 Saving a device

You can save a device to a resource. This enables you to use the same device in different solutions, or use multiple instances of a device in one solution.

Save resource

To save a device to a resource file, perform the following steps:

1. In the **Solution Explorer**, right-click the device.
2. Select **Export Device...**
3. In the **Save As** dialog box, specify the resource file name and the location to save the file.
4. Click **Save**.

Note You can save a device and then load it to add a copy of the device to the solution. However, it is more practical to copy the device with the clipboard or with drag-and-drop (refer to section 3-8).

3-2-3 Configuring a device

The configuration of a device determines the following settings of the device:

- Communication settings
- General settings
- Date/time settings
- Memory settings
- PLC Memory Mapping settings (MCH72 devices only)

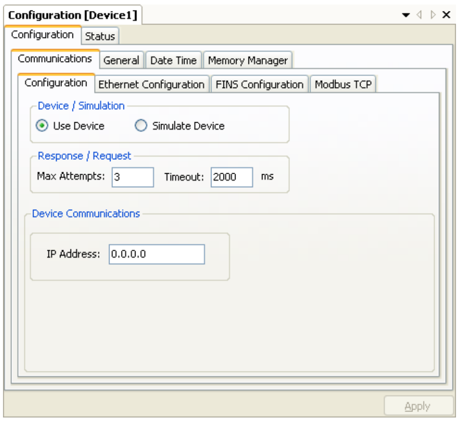
You can change the configuration settings in the **Configuration** window. To open the **Configuration** window of a device, right-click the device and select **Configure...**
If you change one or more configuration settings, you must click **Apply** in the lower right corner of the **Configuration** window to save the settings.

3-2-3-1 Communication settings

The communication settings of a motion device determine the communication between Trajexia Studio and an online device. For more information about online devices, refer to section 4-3.

Trajexia Studio supports 3 communication methods with an online device. You can set these communication methods on the **Configuration** tab on the **Communications** tab on the **Configuration** tab of the **Configuration** window of the device.

- Direct communication
  In this case, Trajexia Studio communicates directly with an online device.
  To use direct communication, select the **Use Device** option button, and enter the IP address of the device in the **IP Address** text field. For the IP address of the device, refer to the documentation of the device.
- Communication via CX-Server
  CX-Server is OMRON software that acts as common communication middleware for PLCs and online devices. MCH72 devices communicate via CX-Server. To configure this communication, you must configure the communication settings of the PLC device, and then configure the communication settings of the MCH72 device.
  To configure the CX-Server settings, click the **Configure** button. For more information on CX-Server, refer to the CX-Server documentation.
- Simulation
  You can set up a virtual device and simulate the communication with this device.
  To use simulation, select the **Simulate Device** option button.



Device / Simulation
● Use Device ○ Simulate Device

Response / Request
Max Attempts: 3 Timeout: 2000 ms

Device Communications
IP Address: 0.0.0.0

Configuration window, **Configuration** tab

Note Not all devices support all communication settings:

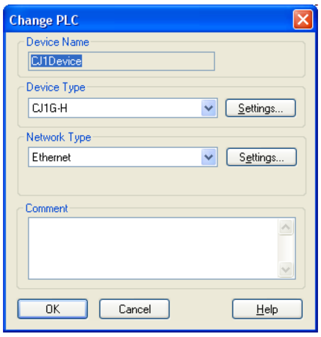
- The TJ1-MC04 and TJ1-MC16 devices only support direct communication and simulation.
- The MCH72 device only supports communication via CX-Server and simulation.

In the **Max Attempts** and **Timeout** text fields, you can enter the number of connection attempts and the timeout period in milliseconds that Trajexia Studio uses to establish the communication to the online device. If Trajexia Studio cannot establish the communication after the entered number of attempts or does not receive a response within the entered timeout period, the communication has failed.

Note A slow communication or an action that takes long (for example, the command **MECHATROLINK(*unit*,0)** on an ML unit with many axes) can cause a timeout. With a larger **Timeout** value, you can resolve this problem

The other tabs in the **Communications** tab, **Ethernet Configuration**, **FINS Configuration** and **Modbus TCP**, show the Ethernet, FINS and Modbus/TCP settings. Refer to the documentation of the device for more information on these settings.

The communication settings of a PLC host device determine the PLC type and the communication network type. To change the PLC communication settings, click **Configure** in the **Configuration** window of the PLC device. The **Change PLC** window, which is part of CX-Server, shows.



Change PLC

Device Name: CJ1Device

Device Type: CJ1G-H

Network Type: Ethernet

Comment:

Change PLC window

In the **Change PLC** window, you can change the PLC type and the network type.

3-2-3-2 General settings

There are 2 general settings for the device:

- Firmware version
  The firmware of a device determines the supported version of the BASIC programming language and the supported units, axes and drives. Trajexia Studio comes with a set of integrated firmware versions. When you go online to a device that has a different firmware version, Trajexia Studio uses the firmware version from the device.

Note Trajexia Studio stores different firmware versions from online devices locally and keeps them for future use.

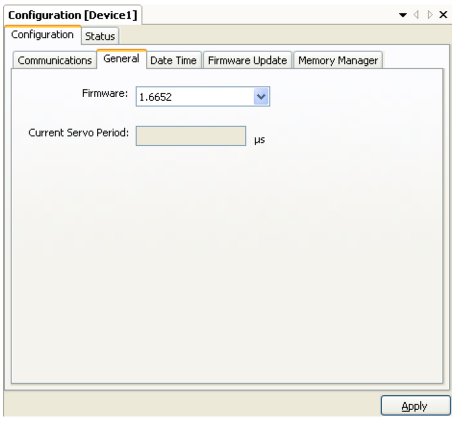
You can select a different firmware version for the device in the **Firmware** list.

Note The **Firmware** list is disabled when the device is online.

- Servo period
  The servo period is the duration in µs of one motion cycle in the device. The number of connected axes determines the minimum servo period of the device.
  You can change the servo period in the **Current Servo Period** text field.

Note The **Current Servo Period** text field is only enabled when the device is online.

Note The **Current Servo Period** text field only shows the current servo period. You can change the servo period in this text field, but the change takes effect only after the device is reset. Until the device is reset, the value that shows in the **Current Servo Period** field is the current servo period, and not the changed value. Trajexia Studio cannot show the new servo period value, because the new servo period can only be written to a device and it cannot be read.



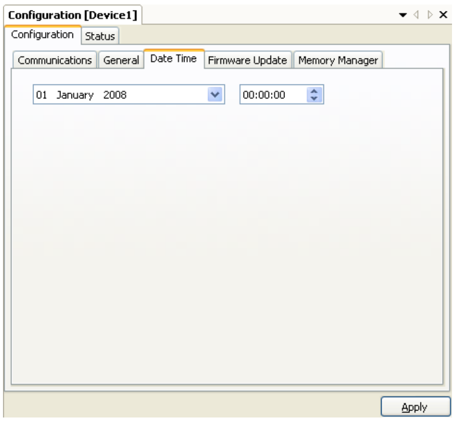
Firmware: 1.6652

Current Servo Period: μs

Configuration window, **General** tab

3-2-3-3 Date/time settings

When the device is not online, the **Date Time** tab shows the current date and time of the computer. When the device is online, the **Date Time** tab shows the date and time of the online device. You can change the date and time and transfer them to the device.



01 January 2008 00:00:00

Configuration window, **Date Time** tab

3-2-3-4 Memory settings

The Table memory of a motion device is a large memory for global data. It is used to store large amounts of data such as motion profiles or logging data. It can be used to create CAM tables, and data traces use it to store data real-time.

The Table memory can be addressed with indexes. The size of the Table memory is 64,000 slots, and the slot indexes range from 0 to 63,999.

The Table memory is divided in the following areas:

Memory area	For use by
Shell	SHELL program
User Table memory	SHELL program and application programs

Memory area	For use by
CAM tables	CAM tables
Oscilloscope	Data traces

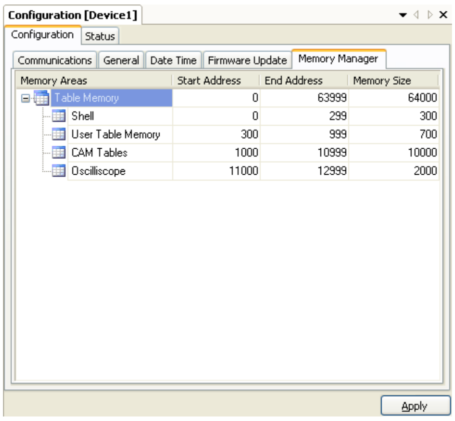
When you allocate Table memory to these areas, it is important to observe the following:

- A device has one memory, which is used for the programs and the Table memory. The larger the Table memory is, the less memory is available for programs.
  Therefore, it is recommended to keep the size of the Table memory to a minimum.
- The Table memory is created from 0 to the largest end address. Thus, if you set the Shell memory area from 10,000 to 10,999, the User Table memory area from 20,000 to 20,999, the CAM tables memory area from 30,000 to 30,999 and the Oscilloscope memory area from 40,000 to 40,999, 41,000 memory slots are allocated for the Table memory, but only 4,000 are used.
  Therefore, it is recommended to allocate the 4 Table memory areas contiguously, and to set the start address of the first Table memory area to 0.

The default Table memory allocation of a device, which is given in the table below, supports these recommendations.

Memory area	Start address	End address	Memory size
Shell	0	299	300
User Table memory	300	999	700
CAM tables	1,000	10,999	10
Oscilloscope	11,000	12,999	2,000

You can change the Table memory assignment of a device in the **Memory Manager** tab on the **Configuration** tab.



Memory Areas	Start Address	End Address	Memory Size
Table Memory	0	63999	64000
Shell	0	299	300
User Table Memory	300	999	700
CAM Tables	1000	10999	10000
Oscilliscope	11000	12999	2000

Configuration window, **Memory Manager** tab

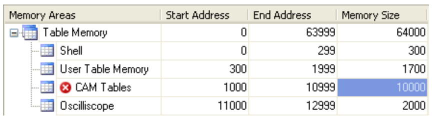
The **Memory Manager** tab has a tabular view of the memory assignment. Refer to section 2-3-1 for information on how to edit a value in a tabular view.

Note The start address, the end address and the memory size of a memory area are interdependent. If you change one of these values, and the cell with this value loses the focus, Trajexia Studio changes another value:

- If you change the start address, Trajexia Studio changes the end address.
- If you change the end address, Trajexia Studio changes the memory size.
- If you change the memory size, Trajexia Studio changes the end address.

Caution
If the addresses of a memory area is changed, the data is not automatically moved. If you want to move the data, you must do this manually.

Note If the values entered for the memory allocation are such that two memory areas overlap, Trajexia Studio shows an error message when you click **Apply**. An icon ⊗ indicates the memory area that is overlapped.

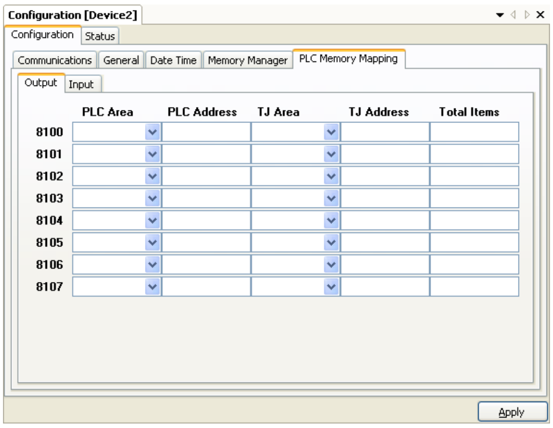


Memory Areas	Start Address	End Address	Memory Size
Table Memory	0	63999	64000
Shell	0	299	300
User Table Memory	300	1999	1700
⊗ CAM Tables	1000	10999	10000
Oscilliscope	11000	12999	2000

The **User Table Memory** area overlaps the **CAM Tables** memory area

3-2-3-5 PLC Memory Mapping settings

The PLC Memory Mapping settings determine how the MCH72 device exchanges configurable data[1] with its host device (PLC). For more information on cyclic data exchange, refer to the CJ1W-MCH72 Motion Control Unit Operation Manual (I55E).



	PLC Area	PLC Address	TJ Area	TJ Address	Total Items
8100					
8101					
8102					
8103					
8104					
8105					
8106					
8107					

Configuration window, **PLC Memory Mapping** tab

The PLC Memory Mapping tab has 2 tabs: one for the 8 output areas (8100..8107), and one for the 8 input areas (100..107)[2]. The table below lists the edit fields on the 2 tabs, the description and the possible values.

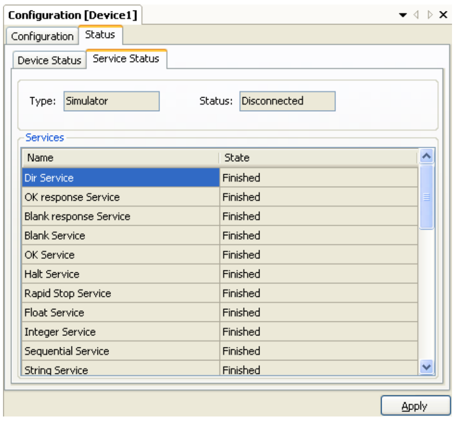
1. The configurable data is all data that is exchanged except control and status data.
2. The words *input* and *output* indicate the direction starting from the MCH72 device. Thus, *output* means *from MCH72 to PLC*, and *input* means *from PLC to MCH72*.

Edit field	Description	Possible values	
PLC Area	The PLC memory area used for data exchange	01	CIO memory
		01	CIO memory
		03	DM memory
		04	WR memory
		05	HR memory
		08..14	EM memory bank 0 hex..C hex
PLC Address	The start address in the PLC memory	The validity of this value depends on the value of **PLC Area**	
TJ Area	The MCH72 memory area used for data exchange	**VR 16bit Int**	VR memory 16-bit signed integer
		VR 32bit Float	VR memory 32-bit floating point
		Digital IO	IN or OP array (depending on direction)
		Analogue IO	AIN or AOUT array (depending on direction)
		Axis Status	Axis Status array (only valid if direction is MCH72 output)
TJ Address	The start address in the MCH72 memory	The validity of this value depends on the value of **TJ Area**	
Total Items	The total number of items (words and dwords) to exchange	The validity of this value depends on the values of **PLC Area** and **TJ Area**	

Note If the PLC type of the PLC host device is not set, you cannot set the data exchange values of the MCH72 device.

3-2-3-6 Status information

The **Status** tab in the **Configuration** window shows the status of the Trajexia Studio communication services that are running (on the **Service Status** tab), the status of the PLC host device (on the **PLC Status** tab; only for MCH72 devices) and the status of the online device (on the **Device Status** tab).



Type: Simulator

Status: Disconnected

Services

Name	State
Dir Service	Finished
OK response Service	Finished
Blank response Service	Finished
Blank Service	Finished
OK Service	Finished
Halt Service	Finished
Rapid Stop Service	Finished
Float Service	Finished
Integer Service	Finished
Sequential Service	Finished
String Service	Finished

Configuration window, **Service Status** tab

3-2-4 Building a device

When you build a device, Trajexia Studio performs the following actions:

- Generate program code for the definitions of axis parameters, drive parameters, Table data, VR variables and CAM table data, and add it to the SHELL program (see section 3-4-5)
- Build all BASIC programs in the device (see section 3-4-3)
- Check that all axes are associated with a drive
- Check that all drives are associated with an axis

Build device

To build a device, perform the following steps:

1 In the **Solution Explorer**, select the device.

2 On the **Build** menu, select **Build Device**.

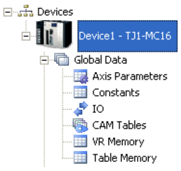
Trajexia Studio builds the device. If the device has errors or warnings, they show in the **Build** window. Refer to section 2-4-2-1 for more information on the **Build** window, build errors and build warnings.

Note To build all devices in the solution, select **Build Solution** on the **Build** menu.

3-3 Global data

Global data is data that is accessible to all programs of a motion device. This means that all BASIC programs of that motion device can access that data. Trajexia Studio devices have 6 types of global data:

- Axis parameters
- Constants
- I/O memory
- CAM tables
- VR memory
- Table memory



Motion device with global data

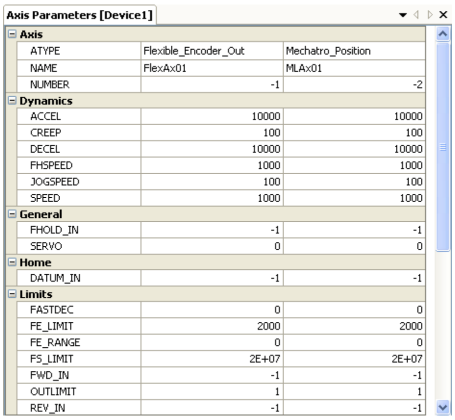
3-3-1 Axis parameters

The axis parameters are the motion parameters for the axes connected to the device. They set and monitor the behaviour of the axes. The axis parameters are unique to each axis, thus all axes can work independently.

Edit axis parameters

To view and edit the axis parameters of all axes connected to a device, perform the following steps:
You can view and edit the axis parameters in the **Axis Parameters** window. To open the **Axis Parameters** window, perform the following steps:

1 In the **Solution Explorer**, right-click **Axis Parameters** under **Global Data**.
2 Select **Open**.



Axis Parameters [Device1]

Axis		
ATYPE	Flexible_Encoder_Out	Mechatro_Position
NAME	FlexAx01	MLAx01
NUMBER	-1	-2
Dynamics		
ACCEL	10000	10000
CREEP	100	100
DECEL	10000	10000
FHSPEED	1000	1000
JOGSPEED	100	100
SPEED	1000	1000
General		
FHOLD_IN	-1	-1
SERVO	0	0
Home		
DATUM_IN	-1	-1
Limits		
FASTDEC	0	0
FE_LIMIT	2000	2000
FE_RANGE	0	0
FS_LIMIT	2E+07	2E+07
FWD_IN	-1	-1
OUTLIMIT	1	1
REV_IN	-1	-1

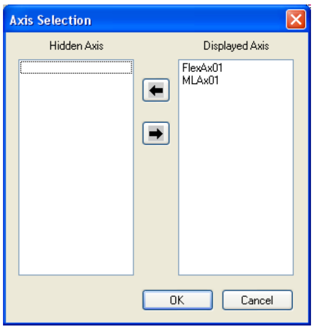
Axis Parameters window

The **Axis Parameters** window has categorised list of commonly used axis parameters for all axes in the device. The rows show the axis parameters, the columns show the axes connected to the device. The axis parameters are grouped together in categories. The values of read-only axis parameters are shown in gray. The values of axis parameters that have errors are shown with a red background colour. Editing a cell in the categorised list in the **Axis Parameters** window is similar to editing a cell in a tabular view. Refer to section 2-3-1 for information on how to edit a value in a tabular view.

You can hide or show the parameters of axes in the **Axis Parameters** window. The table below lists the actions to show or hide one or more axes in the **Axis Parameters** window. The second column lists the corresponding commands on the shortcut menu of the **Axis Parameters** window.

Action	Command
Hide an axis	**Hide Axis**
Show all axes	**Show All Axes**
Select which axes to show or hide	**Select Axis...**[1]

[1] This command shows the **Axis Selection** window, where you can move an axis to the **Hidden Axis** or the **Displayed Axis** list with the arrow buttons.



Axis Selection window

3-3-2 Constants

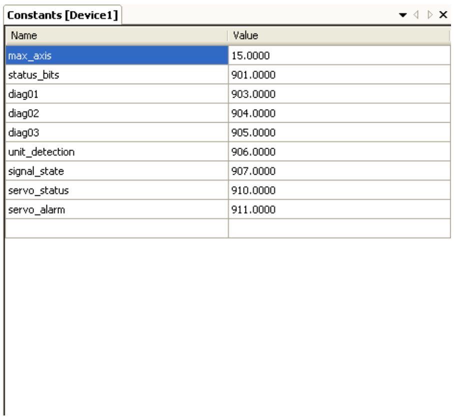
A constant is a variable that is assigned a specific value that cannot be changed. The maximum number of constants in all BASIC programs in a device is 128.

Edit constants

You can view and edit the constants in the **Constants** window. To open the **Constants** window, perform the following steps:

1 In the **Solution Explorer**, right-click **Constants** under **Global Data**.

2 Select **Open**.



Name	Value
max_axis	15.0000
status_bits	901.0000
diag01	903.0000
diag02	904.0000
diag03	905.0000
unit_detection	906.0000
signal_state	907.0000
servo_status	910.0000
servo_alarm	911.0000

Constants window

The **Constants** window has a tabular view of all constants and their values. Refer to section 2-3-1 for information on how to edit a value in the **Constants** window.

You can change the display precision of the values in the **Value** column with the **Precision** property in the **Properties** window.

Note If you change the display precision, Trajexia Studio does not change the value, but only the display format of the value.

A constant definition corresponds to the BASIC command **CONSTANT**. When the device or the solution is built, a **CONSTANT** command is added to a read-only section of the SHELL program for each constant definition in the **Constants** window.

Note If you add a constant definition to an application program with the **CONSTANT** command, a warning is generated when the program is built.

Add constant

To add a constant, perform the following steps:

1 In the last row in the **Constants** window, which is empty, type a constant name in the **Name** column.

2 Type a value in the **Value** column in the same row.

Delete constant

To delete a constant, perform the following steps:

1 In the **Constants** window, select the constant to delete.

2 Select **Delete** on the **Edit** menu.

3-3-3 I/O memory

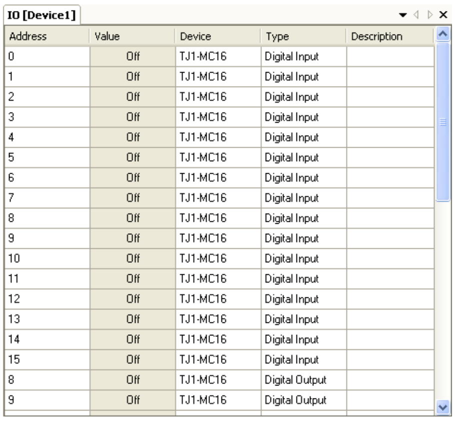
The I/O memory holds the status of the inputs and outputs of the device. The inputs and outputs can have the status ON or OFF.

Edit I/O memory

You can view and edit the I/O memory in the **IO** window. To open the **IO** window, perform the following steps:

1 In the **Solution Explorer**, right-click **IO** under **Global Data**.

2 Select **Open**.



Address	Value	Device	Type	Description
0	Off	TJ1-MC16	Digital Input	
1	Off	TJ1-MC16	Digital Input	
2	Off	TJ1-MC16	Digital Input	
3	Off	TJ1-MC16	Digital Input	
4	Off	TJ1-MC16	Digital Input	
5	Off	TJ1-MC16	Digital Input	
6	Off	TJ1-MC16	Digital Input	
7	Off	TJ1-MC16	Digital Input	
8	Off	TJ1-MC16	Digital Input	
9	Off	TJ1-MC16	Digital Input	
10	Off	TJ1-MC16	Digital Input	
11	Off	TJ1-MC16	Digital Input	
12	Off	TJ1-MC16	Digital Input	
13	Off	TJ1-MC16	Digital Input	
14	Off	TJ1-MC16	Digital Input	
15	Off	TJ1-MC16	Digital Input	
8	Off	TJ1-MC16	Digital Output	
9	Off	TJ1-MC16	Digital Output	

IO window

The **IO** window has a tabular view of all inputs and outputs and their status. You can only edit the values in the **Description** column. Refer to section 2-3-1 for information on how to edit a value in the **IO** window.

Note The value in the **Description** column is not stored in the device. It is only stored in the solution.

3-3-4 CAM tables

A CAM table is an area of the Table memory. It is used to read or write a set of positions that define a movement profile. CAM tables are used by the BASIC commands **CAM** and **CAMBOX**.

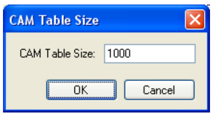
Add new CAM table

To add a CAM table to a device, perform the following steps:

1 In the **Solution Explorer**, right-click **CAM Tables** under **Global Data**.

2 Select **Add Cam Table**.

3 In the **CAM Table Size** window, enter the memory size of the new CAM table.



CAM Table Size: 1000

CAM Table Size window

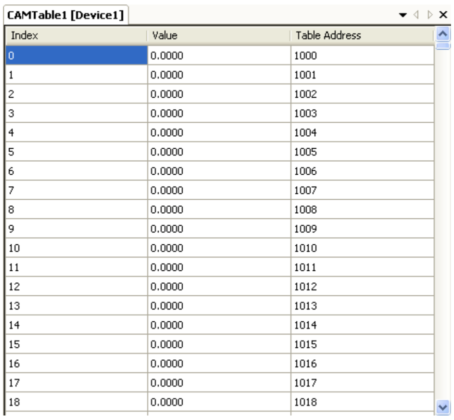
4 Click **OK**.

A new CAM Table has the default name CAMTable*n*, where *n* is a sequence number.

Edit CAM table You can view and edit a CAM table in the CAM table window. To open the CAM table window, perform the following steps:

1 In the **Solution Explorer**, right-click the CAM table to edit under **CAM Tables** under **Global Data**.

2 Select **Open**.


CAMTable1 [Device1]

Index	Value	Table Address
0	0.0000	1000
1	0.0000	1001
2	0.0000	1002
3	0.0000	1003
4	0.0000	1004
5	0.0000	1005
6	0.0000	1006
7	0.0000	1007
8	0.0000	1008
9	0.0000	1009
10	0.0000	1010
11	0.0000	1011
12	0.0000	1012
13	0.0000	1013
14	0.0000	1014
15	0.0000	1015
16	0.0000	1016
17	0.0000	1017
18	0.0000	1018

CAM table window

The CAM table window has a tabular view that shows the values of the CAM table. Refer to section 2-3-1 for information on how to edit a value in a CAM table window.

You can change the display precision of the values in the **Value** column with the **Precision** property in the **Properties** window.

Note If you change the display precision, Trajexia Studio does not change the value, but only the display format of the value.

The order of CAM tables in the **Solution Explorer** can be changed (refer to section 3-8). The order of the CAM tables does not have an influence on the Table memory allocation.

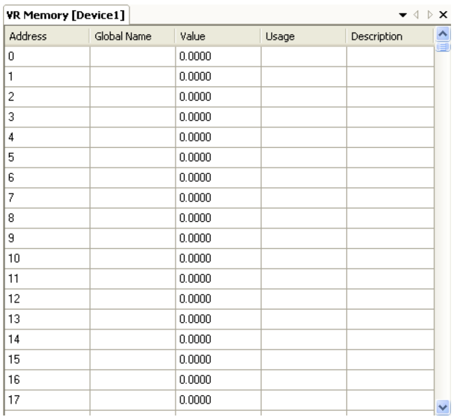
3-3-5 VR memory

The VR memory is an array of global variables. These variables are called VR variables. The VR memory array is an indexed array. Its size is 1,024, and the indexes range from 0 to 1,023. The VR memory is used for small amounts of global data.

Edit VR memory You can view and edit the VR memory in the **VR Memory** window. To open the **VR Memory** window, perform the following steps:

1 In the **Solution Explorer**, right-click **VR Memory** under **Global Data**.

2 Select **Open**.



VR Memory [Device1]

Address	Global Name	Value	Usage	Description
0		0.0000		
1		0.0000		
2		0.0000		
3		0.0000		
4		0.0000		
5		0.0000		
6		0.0000		
7		0.0000		
8		0.0000		
9		0.0000		
10		0.0000		
11		0.0000		
12		0.0000		
13		0.0000		
14		0.0000		
15		0.0000		
16		0.0000		
17		0.0000		

VR Memory window

The **VR Memory** window has a tabular view that has the following columns:

Column	Description
Address	The index of the variable in the VR array
Global Name	The alias for the index of the VR variable
Value	The value of the VR variable
Usage	The item that uses the variable (Possible values are the SHELL program, the PRT unit and the DRT unit)
Description	A description of the VR variable

The columns **Global Name**, **Value** and **Description** can be edited. The columns **Address** and **Usage** are read-only. Refer to section 2-3-1 for information on how to edit a value in the **VR Memory** window.

Note The value in the **Description** column is not stored in the online device if it is synchronised. It is only stored in the solution.

You can change the display precision of the values in the **Value** column with the **Precision** property in the **Properties** window.

Note If you change the display precision, Trajexia Studio does not change the value, but only the display format of the value.

You can limit the number of rows that show in the tabular view to a user-defined area of the VR memory with the **Offset** and **Size** properties in the **Properties** window.

The definition of the global name alias is equivalent to the BASIC command **GLOBAL**. The command **GLOBAL "ratio",23** defines the variable **ratio** as an alias for **VR(23)**. In a similar way, the definition of an alias in the **VR Memory** window is reflected in the SHELL program. If you add the variable *var* in the **Global Name** column in the row where the **Address** column has value *n*, the line **GLOBAL "*var*",*n*** is added to the read-only section of the SHELL program. For more information on generating SHELL code, refer to section 3-4-5.

A VR variable can have an alias. This alias is a reference to the index of the variable in the VR array.

3-3-6 Table memory

The Table memory contains the Table variables of the device. It is used for large amounts of global data, such as CAM tables and data traces.

Edit Table memory

You can view and edit the Table memory in the **Table Memory** window. To open the **Table Memory** window, perform the following steps:

1 In the **Solution Explorer**, right-click **Table Memory** under **Global Data**.

2 Select **Open**.

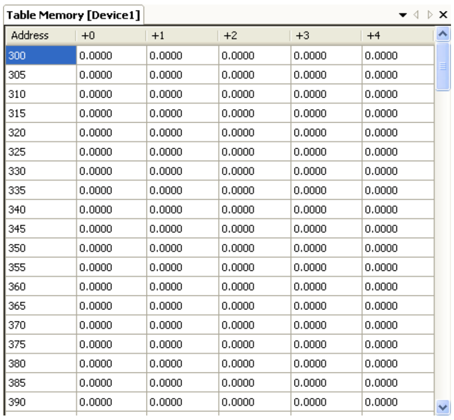


Table Memory [Device1]

Address	+0	+1	+2	+3	+4
300	0.0000	0.0000	0.0000	0.0000	0.0000
305	0.0000	0.0000	0.0000	0.0000	0.0000
310	0.0000	0.0000	0.0000	0.0000	0.0000
315	0.0000	0.0000	0.0000	0.0000	0.0000
320	0.0000	0.0000	0.0000	0.0000	0.0000
325	0.0000	0.0000	0.0000	0.0000	0.0000
330	0.0000	0.0000	0.0000	0.0000	0.0000
335	0.0000	0.0000	0.0000	0.0000	0.0000
340	0.0000	0.0000	0.0000	0.0000	0.0000
345	0.0000	0.0000	0.0000	0.0000	0.0000
350	0.0000	0.0000	0.0000	0.0000	0.0000
355	0.0000	0.0000	0.0000	0.0000	0.0000
360	0.0000	0.0000	0.0000	0.0000	0.0000
365	0.0000	0.0000	0.0000	0.0000	0.0000
370	0.0000	0.0000	0.0000	0.0000	0.0000
375	0.0000	0.0000	0.0000	0.0000	0.0000
380	0.0000	0.0000	0.0000	0.0000	0.0000
385	0.0000	0.0000	0.0000	0.0000	0.0000
390	0.0000	0.0000	0.0000	0.0000	0.0000

Table Memory window

The **Table Memory** window has a tabular view that shows the values in the Table memory. Refer to section 2-3-1 for information on how to edit a value in the **Table Memory** window.

You can change the display precision of the values in the tabular view with the **Precision** property in the **Properties** window.

Note If you change the display precision, Trajexia Studio does not change the value, but only the display format of the value.

Because the Table memory can be large, the **Table Memory** window has some features to change the layout of the tabular view. The table below lists the features. The second column lists the properties in the **Properties** window that you must change to enable the features.

Feature	Property
Change the number of columns in the tabular view	**Columns**
Limit the values that show to a specific Table memory area (Shell, User Table memory, CAM tables or Oscilloscope)	**Table Memory Area**
Limit the number of values that show to a user-defined area of the Table memory	**Offset** and **Size**

3-4 Programs

Programs are the essential elements of devices: they drive the behaviour of the device and the connected axes. The programs are written in the BASIC programming language.

3-4-1 Program concepts

A motion application can consist of several BASIC programs. A program can have an influence on the execution of the other programs. For example, a program can start another program with the command **RUN**.

The results of a motion application where the programs have an influence on the other programs can be unpredictable and even dangerous. Therefore it is recommended to minimise the influence that the programs can have on other programs. This can be done with the following 2 recommendations:

- There is one program that can have an influence on other programs. This is the *control* program. The control program does the following:
  - It starts, stops and resets the execution of other programs.
  - It monitors the execution of other programs.
- All other programs are separate. They do not have an influence on other programs. They do not do the following:
  - They do not start, stop or reset the execution of other programs.
  - They do not know about the other programs except the control program.

To support these recommendations, Trajexia Studio provides 2 kinds of programs: the SHELL program and the application programs.

- The SHELL program is intended as the control program. It does the following:
  - The SHELL program starts, stops and resets the execution of the application programs.
  - The SHELL program monitors the execution of the application programs.
  - The SHELL program monitors the drive statuses.
  - The SHELL program initialises the axis parameters, Table data, VR variables and CAM table data.
- The application programs must not have an influence on other programs. It is recommended that they do not do the following:
  - They do not start, stop or reset the execution of application programs or the SHELL program.
  - They do not know about the application programs. They only know about the SHELL program.

The SHELL program and the application programs have different characteristics in Trajexia Studio:

- A motion device can have one SHELL program. It can have several application programs.
- The auto-generated code for a new SHELL program is very extensive. The auto-generated code for a new application program does only contain some comments, but no BASIC commands.
- Trajexia Studio provides tools to add the code for the definitions of axis parameters, drive parameters, Table data, VR variables and CAM table data to the SHELL program. These tools are not available for the application programs. Refer to section 3-4-5.
- The **Solution Explorer** shows the icon for the SHELL program in a different colour.

With these characteristics, the SHELL program provides a framework to make sure that your motion application follows the 2 recommendations. However, Trajexia Studio does not enforce the recommendations. This means that the following actions are possible in Trajexia Studio:

- The SHELL program can be deleted (refer to section 3-8). Thus, it is possible to have a device without a SHELL program.
- It is possible to start an application program from another application program.

It is recommended not to do the actions described above.

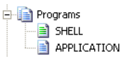
The reason that Trajexia Studio does not enforce the recommendations is to provide compatibility with legacy programs and motion applications. However, it is strongly recommended to follow the recommendations.

Note Trajexia Studio identifies the SHELL program of a motion device only by its name. Therefore, if you rename the SHELL program (refer to section 3-8), it has the characteristics of an application program (for example, the icon colour changes to the default colour). Similarly, if you change the name of an application program to 'SHELL', it has the characteristics of the SHELL program (for example, Trajexia Studio provides tools to add the code for the definition of Table data to the program).

It is recommended not to rename the SHELL program, and not to rename an application program to SHELL.

3-4-2 Adding a program

When you add a new motion device to the solution, 2 programs are created by default: the SHELL program, and an application program called APPLICATION.



Default programs SHELL and APPLICATION

Add program To add a new BASIC program to a motion device, perform the following steps.

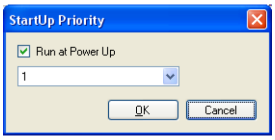
1 In the **Solution Explorer**, right-click **Programs** under a motion device.

2 Select **Add SHELL Program** to add the SHELL program, or select **Add Basic Program** to add an application program.

Note The **Add SHELL Program** menu item is disabled if the motion device already contains the SHELL program.

A device can execute a program at startup: when the device is switched on, it executes the program.

You can set the startup priority for a program in Trajexia Studio with the **Priority** property in the **Properties** window. If you click the ellipsis button in the edit field of this property, the **StartUp Priority** window shows.


StartUp Priority

☑ Run at Power Up

1

StartUp Priority window

To set the program to run at power up, select the **Run at Power Up** check box and select a priority in the list. Possible priority values are **Default** or 1 (lowest priority) to 14 (highest priority).

To set the program not to run at startup, clear the **Run at Power Up** check box.

Note The SHELL program by default runs at startup at priority 1.

3-4-3 Building programs

Build program To build a BASIC program, perform the following steps:

1 Go to the code editor that contains the program code to build.

2 On the **Build** menu, select **Build Program**.

Trajexia Studio compiles and builds the program. If the program has errors or warnings, they show in the **Build** window. Refer to section 2-4-2-1 for more information on the **Build** window, build errors and build warnings.

3-4-4 Printing a program

Print program

To print a BASIC program, perform the following steps:

1 In the **Solution Explorer**, right-click the program that you want to print.

2 Select **Print Program**.

3-4-5 Generating SHELL code

Trajexia Studio can add the code for the definitions of axis parameters, drive parameters, Table data, VR variables and CAM table data to the SHELL program.

Note Drive parameters can only be added to the SHELL program from the **Axis Configuration** window or the **Drive Configuration** window. Refer to section 3-6-6-5.

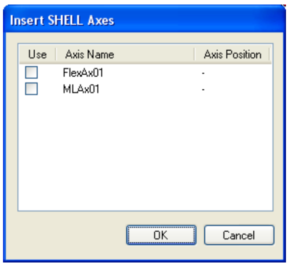
Add axis parameters

To add the axis parameters of one or more axes to the SHELL program, perform the following steps:

1 In the **Solution Explorer**, select the SHELL program.

2 Select **Insert Axes...** under **Modify SHELL** on the **Programs** menu.

3 In the **Insert SHELL Axes** window, select the check boxes for the axes of which you want to add the axis parameters to the SHELL code.



Use	Axis Name	Axis Position
□	FlexAx01	-
□	MLAx01	-

Insert SHELL Axes window

4 Click **OK**.

For each selected axis, the code for the assignment of the axis parameters is added under **' Axis Parameters** in the **' Start Standard Section** outline of the SHELL program.

Note

- Trajexia Studio does not check if the code that you want to generate is already in the SHELL program. If you add code for the same axis parameters twice, the same axis parameter assignments are in the SHELL code twice.
- You can change the added code.

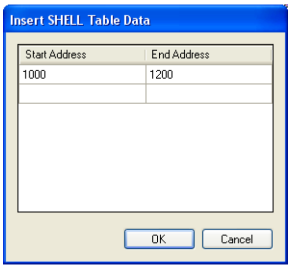
Add Table data

To add Table data to the SHELL program, perform the following steps:

1 In the **Solution Explorer**, select the SHELL program.

2 Select **Insert Table Data...** under **Modify SHELL** on the **Programs** menu.

3 In the **Insert SHELL Table Data** window, enter the start and end addresses for the Table memory areas for which you want to generate SHELL code.



Start Address	End Address
1000	1200

Insert SHELL Table Data window

4 Click **OK.**

For each selected Table memory area, one or more lines **TABLE(*start-address*,*value-1*,..*value-n*)** are added under **' TABLE DATA** in the **' Start Standard Section** outline of the SHELL program. To make the **TABLE** statements not too long, one **TABLE** statement is generated per 10 memory addresses.

Note

- Trajexia Studio does not check if the code that you want to generate is already in the SHELL program. If you add code for the same Table memory area twice, the same **TABLE** statements are in the SHELL code twice.
- You can change the added code.

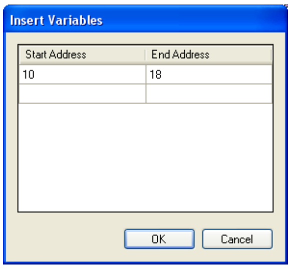
Add VR variables

To add VR variables to the SHELL program, perform the following steps:

1 In the **Solution Explorer**, select the SHELL program.

2 Select **Insert VR Variables...** under **Modify SHELL** on the **Programs** menu.

3 In the **Insert Variables** window, enter the start and end addresses for the memory areas for which you want to generate SHELL code.



Start Address	End Address
10	18

Insert Variables window

4 Click **OK.**

For each VR variable in the selected VR memory areas, the line **VR(*index*) = *value*** is added under **' Variables** in the **' Start Standard Section** outline of the SHELL program.

Note

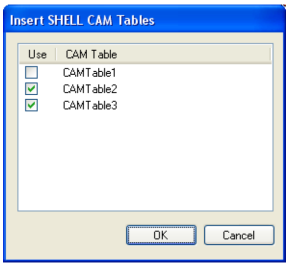
- Trajexia Studio does not check if the code that you want to generate is already in the SHELL program. If you add code for the same VR memory area twice, the same **VR** statements are in the SHELL code twice.
- You can change the added code.

Add/remove CAM table data

The code generation for CAM table data works differently than the other code generations, because the code for CAM table data is added to a read-only section of the SHELL code. This means that Trajexia Studio can check if the code already exists, and it can remove the code from the SHELL program. The user cannot edit the CAM table data code.

To add or remove CAM table data of one or more CAM tables to the SHELL program, perform the following steps:

1 In the **Solution Explorer**, select the SHELL program.
2 Select **Insert CAM Table...** under **Modify SHELL** on the **Programs** menu. The **Insert SHELL CAM Tables** window shows.



Use	CAM Table
☐	CAMTable1
☑	CAMTable2
☑	CAMTable3

Insert SHELL CAM Tables window

If the check boxes of a CAM table has a check mark, the SHELL program already has the code for this CAM table. If it does not have a check mark, the SHELL program does not yet have the code for this CAM table.

3 Select the check boxes for the CAM tables for which you want to add the code to the SHELL program.
4 Clear the check boxes for the CAM tables for which you want to remove the code from the SHELL program.
5 Click **OK**.

For each selected CAM table, one or more lines **TABLE(*start-address*,*value-1*,..*value-n***) are added under **'Omron Auto Generated - CAM TABLE** in the **'Omron Auto Generated - Symbols** outline (which is read-only). To make the **TABLE** statements not too long, one **TABLE** statement is generated per 10 memory addresses.

3-5 Axes

An axis is a machine or a motor that controls the motion. Trajexia Studio supports 3 types of axes:

- Flexible axis
  A flexible axis is an axis controlled by an FL unit.
- MECHATROLINK-II axis
  A MECHATROLINK-II axis is an axis controlled by a MECHATROLINK-II Master Unit.
- Virtual axis
  A virtual axis is an axis that does not correspond to a MECHATROLINK-II axis or a flexible axis.

3-5-1 Axis concepts

Trajexia Studio makes a difference between the functional characteristics and the hardware characteristics of axes. This makes the development of a solution very flexible, because it is possible to write programs that use axes and at the same time postpone the decision on which specific axis hardware you use. For example, you can add a MECHATROLINK-II servo axis to a motion device and write a BASIC program that uses this axis without knowing the hardware characteristics of this servo axis, such as the model, the address of the axis, and the MECHATROLINK-II unit that it is connected to. Later you can choose the specific axis hardware and link it to the servo axis in Trajexia Studio.

To reflect the distinction between the functional characteristics and the hardware characteristics of an axis, Trajexia Studio supports axes at two levels:

- Axes
  When Trajexia Studio uses the word axis, it refers to the functional characteristics of an axis. There characteristics are:
  - Axis type (MECHATROLINK-II, flexible or virtual)
  - Functionality (position, speed or torque, encoder or pulse output)

  These axes show under **Axes** in the **Solution Explorer**.
- Drives connected to units
  This level supports the hardware-related characteristics of an axis. These are:
  - Hardware (servo or inverter, model, firmware version)
  - The unit to which it is connected, and the port or address it uses on this unit

  Drives are connected to units, which show under **Units** in the **Solution Explorer**.

Axes are described in this section. Units and drives are described in section 3-6 (Units).

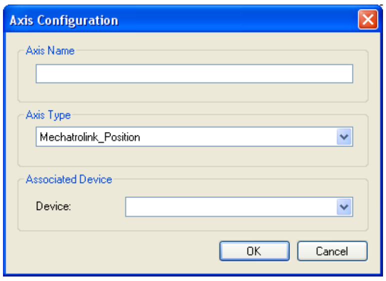
3-5-2 Adding an axis

Add axis To add an axis to a motion device, perform the following steps:

1 In the **Solution Explorer**, right-click **Axes** under a motion device.

2 Select **Add *axis-type-to-add***.

3 In the **Axis Configuration** window, type a name for the axis in the **Axis Name** text field.



Axis Type: Mechatrolink_Position

Axis Configuration window

4 In the **Axis Type** list, select an axis type.

Note You can change the axis type. For example, if you selected **Add Flexible Axis** in step 2, you can change the axis type in this step to a MECHATROLINK-II axis or a virtual axis.

Note The axis type corresponds to the BASIC command **ATYPE**.

5 In the **Device** list, select a drive to associate with the axis. If there are no available drives, or you do not want to associate the axis with a drive, you can skip this step.
Refer to section 3-5-3 for more information on associating axes.

Note You cannot associate a virtual axis with a drive. Thus, if the **Axis Type** is **Virtual**, the **Device** list is disabled.

For more information on associating a drive with an axis, refer to section 3-5-3.

6 Click **OK**.

3-5-3 Associating an axis with a drive

An axis in Trajexia Studio represents only the functional characteristics of a motion axis. To make the axis in Trajexia Studio represent a complete motion axis, it must be linked to a drive that is connected to a unit. This is called associating.

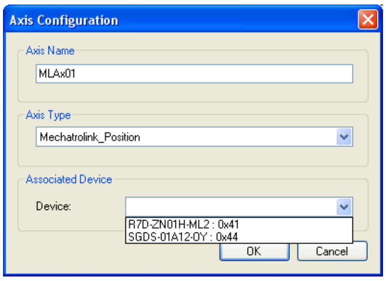
Note If an axis in Trajexia Studio is not associated with a drive, Trajexia Studio generates an error when the device or the solution is built.
This does not hold for a virtual axis, because a virtual axis cannot be associated with a drive.

Associate axis

1 In the **Solution Explorer**, right-click the axis that you want to associate with a drive.
2 Select **Assign Physical Device**.
3 In the **Axis Configuration** window, select a drive in the **Device** list to associate with the axis.

Note The list of drives in the **Device** list is limited to drives that can support the axis type selected in the **Axis Type** list. Also, drives that are already associated to another axis are not in the list.

Note If you change the selected axis type in the **Axis Type** list, the selected drive in the **Device** list is cleared.



Axis Name: MLAx01
Axis Type: Mechatrolink_Position

Axis Configuration window

4 Click **OK**.

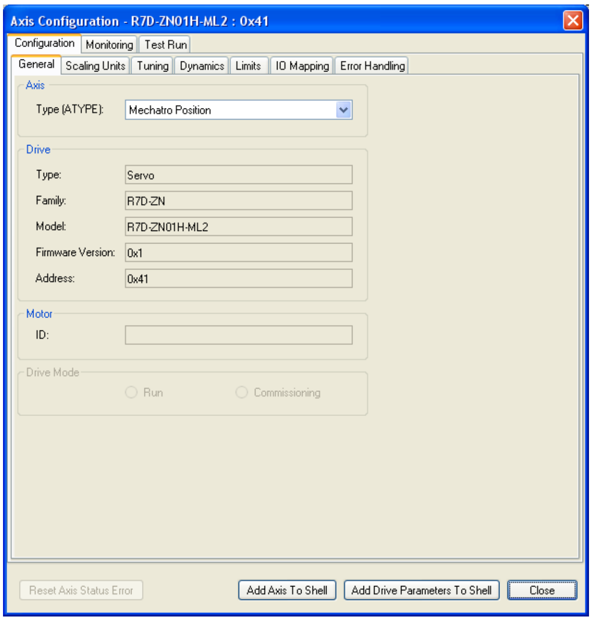
Note If you associate a drive with an axis, Trajexia Studio assigns an axis number to the axis. The axis number is determined by the drive address. Because drive addresses must be unique, the axis numbers are also unique, except in the following case: If a servo and an inverter have the same last digit in their address (for example, 41 hex 21 hex), and the inverter is associated to an axis as a MECHATROLINK-II inverter, Trajexia Studio assigns the same axis number to both associated axes. This gives a conflict, and results in a build error.

3-5-4 Using an axis

Before you use an axis, you must configure it. To configure it properly, it is useful to monitor the axis and test the axis. The **Axis Configuration** window groups together these actions. To show the **Axis Configuration** window, right-click an axis in the **Solution Explorer** and select **Configure...**

3-5-4-1 Configuring an axis

The configuration of an axis is determined by the values of the axis parameters and the drive mappings of the associated drive. The **Configuration** tab of the **Axis Configuration** window shows the real-time values of commonly used axis parameters if the device is online. If the device is offline, you can change the value of the axis parameters and add the code to initialise them in the SHELL program.



Type (ATYPE): Mechatro Position

Type: Servo
Family: R7D-ZN
Model: R7D-ZN01H-ML2
Firmware Version: 0x1
Address: 0x41

ID:

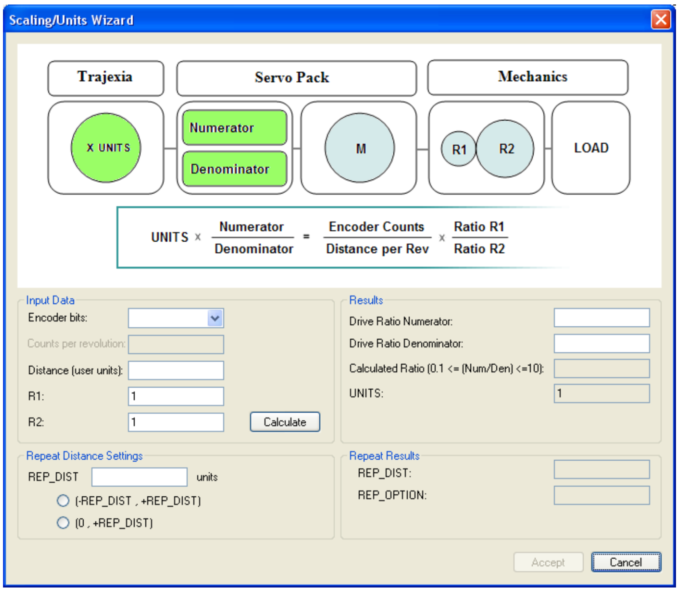
Axis Configuration window, **Configuration** tab

The table below lists the tabs on the **Configuration** tab and the axis parameters and information they show.

Tab	Axis parameters/information
General	• **ATYPE** • The associated drive information
Scaling Units	• **UNITS**, **REP_DIST**, **REP_OPTION** • The gear ratio numerator and denominator of the associated drive • Scaling/Units wizard (see below) • Encoder numerator and denominator (flexible axis only) • **ENCODER_BITS** and **ENCODER_CONTROL** (EnDat and SSI flexible axes only)
Tuning	**P_GAIN**, **I_GAIN**, **D_GAIN**, **OV_GAIN**, **VFF_GAIN**
Dynamics[1]	**SPEED**, **ACCEL**, **DECEL**, **CREEP**
Limits	**FE_LIMIT**, **RS_LIMIT**, **FS_LIMIT**, **FASTDEC**
IO Mapping	• **FWD_IN**, **REV_IN**, **DATUM_IN** • The drive mappings of the associated drive • Intelligent assignment, which automatically sets the drive mappings and restricts the possible drive mapping options. To set the drive mappings manually, clear the **Enable Intelligent Assignment** check box. Intelligent assignment is automatically cleared if the axis is transferred to the online device and the values of the drive mappings are not possible according to the rules of the firmware of the online device.
Error Handling	**ERRORMASK**

1. The values on this tab are not the current axis parameter values. They are default values, which can be added to the SHELL program with the **Add Axis To Shell** button to initialise the axis parameters. The values on this tab can only be changed when the device is offline.

The Scaling/Units wizard calculates the axis parameters **UNITS**, **REP_DIST** and **REP_OPTION** and the gear ratio numerator and denominator of the drive. To open the wizard, click **Wizard...** on the **Scaling Units** tab.


Scaling/Units Wizard
Trajexia
Servo Pack
Mechanics
X UNITS
Numerator
Denominator
M
R1
R2
LOAD
UNITS x Numerator / Denominator = Encoder Counts / Distance per Rev x Ratio R1 / Ratio R2

Input Data

Encoder bits:	
Counts per revolution:	
Distance (user units):	
R1:	1
R2:	1

Calculate

Results

Drive Ratio Numerator:	
Drive Ratio Denominator:	
Calculated Ratio (0.1 <= (Num/Den) <=10):	
UNITS:	1

Repeat Distance Settings

REP_DIST units

○ (-REP_DIST , +REP_DIST)

○ (0 , +REP_DIST)

Repeat Results

REP_DIST:

REP_OPTION:

Accept Cancel

Scaling/Units Wizard window

To calculate the axis parameters and the drive ratio, select a value in the **Encoder bits** field, type a value in the fields **Distance (user untis)**, **R1**, **R2**, **REP_DIST**, select a repeat distance option with the option buttons and click **Calculate**. To do a new calculation, click **Reset**.

Note If the calculated values are invalid, they show with a red background.

To copy the results to the **Scaling Units** tab and close the **Scaling/Units Wizard** window, click **Accept**.

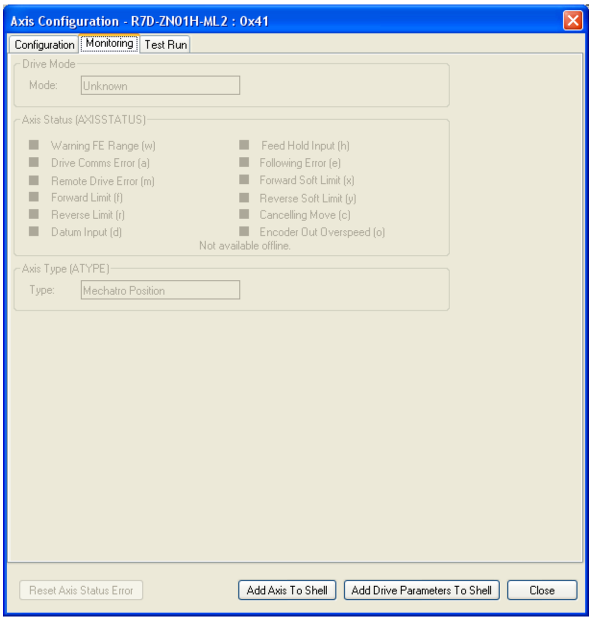
The table below lists the actions that Trajexia Studio supports to configure the axis. The second column in the table lists the corresponding buttons on the **Axis Configuration** window.

Action	Button
Generate code for the axis parameters and add it to the SHELL program[1]	**Add Axis To Shell**
Generate code for the drive parameters and add it to the SHELL program	**Add Drive Parameters To Shell**
Execute CX-Drive to tune the drive	**Launch CX-Drive** on the **Tuning** tab

[1] See also section 3-4-5.

3-5-4-2 Monitoring an axis

The **Monitoring** tab of the **Axis Configuration** window gives feedback about the axis status when the device is online.



Drive Mode

Mode: Unknown

Axis Status (AXISSTATUS)

■ Warning FE Range (w)	■ Feed Hold Input (h)
■ Drive Comms Error (a)	■ Following Error (e)
■ Remote Drive Error (m)	■ Forward Soft Limit (x)
■ Forward Limit (f)	■ Reverse Soft Limit (y)
■ Reverse Limit (r)	■ Cancelling Move (c)
■ Datum Input (d)	■ Encoder Out Overspeed (o)

Not available offline.

Axis Type (ATYPE)

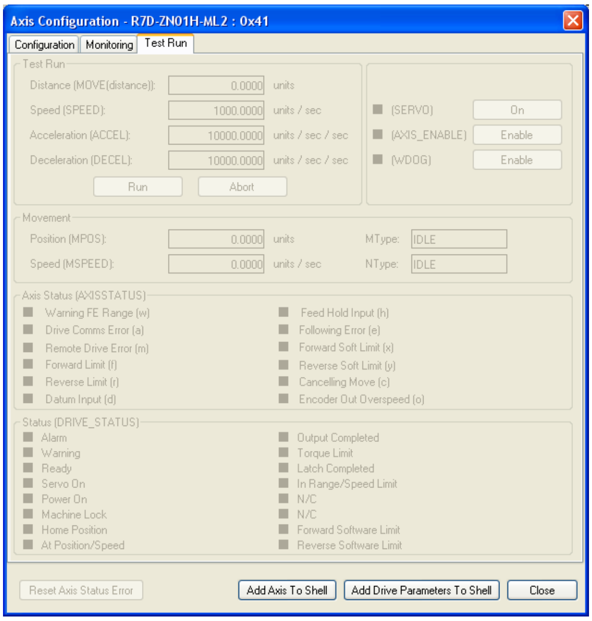
Type: Mechatro Position

Axis Configuration window, **Monitoring** tab

When an axis status bit is on, the corresponding square is red. When an axis status bit is off, the corresponding square is gray.

3-5-4-3 Testing an axis

On the **Test Run** tab of the **Axis Configuration** window, you can test an axis to check if it is properly connected and configured. To do this, the device must be online, and the axis must be associated to a drive.



Test Run

Distance (MOVE(distance)): 0.0000 units

Speed (SPEED): 1000.0000 units / sec

Acceleration (ACCEL): 10000.0000 units / sec / sec

Deceleration (DECEL): 10000.0000 units / sec / sec

■ (SERVO)

■ (AXIS_ENABLE)

■ (WDOG)

Movement

Position (MPOS): 0.0000 units MType: IDLE

Speed (MSPEED): 0.0000 units / sec NType: IDLE

Axis Status (AXISSTATUS)

■ Warning FE Range (w)	■ Feed Hold Input (h)
■ Drive Comms Error (a)	■ Following Error (e)
■ Remote Drive Error (m)	■ Forward Soft Limit (x)
■ Forward Limit (f)	■ Reverse Soft Limit (y)
■ Reverse Limit (r)	■ Cancelling Move (c)
■ Datum Input (d)	■ Encoder Out Overspeed (o)

Status (DRIVE_STATUS)

■ Alarm	■ Output Completed
■ Warning	■ Torque Limit
■ Ready	■ Latch Completed
■ Servo On	■ In Range/Speed Limit
■ Power On	■ N/C
■ Machine Lock	■ N/C
■ Home Position	■ Forward Software Limit
■ At Position/Speed	■ Reverse Software Limit

Axis Configuration window, **Test Run** tab

A test run is a simple move at a given speed with a given acceleration and deceleration. You can set the distance argument of the BASIC command **MOVE** and the **SPEED**, **ACCEL** and **DECEL** parameters. Also, you can enable the **SERVO**, **AXIS_ENABLE** and **WDOG** parameters for the test run with the respective **On** or **Enable** buttons.

When these parameters are set, you can click **Run** to execute the test run. To stop the test run, you can click the **Abort** button.

The **Movement**, **Axis Status (AXISSTATUS)** and **Status (DRIVE_STATUS)** groups on the **Test Run** tab give feedback about the movement of the axis, the axis status and the drive status respectively when the test run is executed.

3-5-5 Finding an axis

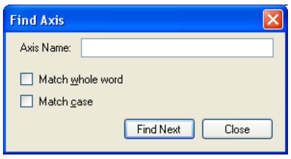
If the solution contains many axes, it can be difficult to find a specific axis in the **Solution Explorer**. Therefore, Trajexia Studio has search functionality to find an axis by its name.

Find axis

To find an axis in the solution, perform the following steps:

1 On the **Edit** menu, select **Find and Replace**, and then **Find Axis**.

2 In the **Find Axis** window, type the name or a part of the name of the axis you want to find in the **Axis Name** field.



Find Axis window

3 If you want to search for an axis by its whole name, select the **Match whole word** check box.

4 If you want to make the search case-sensitive, select the **Match case** check box.

5 Click **Find Next**.

If the axis is found, it is selected in the **Solution Explorer**.

3-6 Units

A device can have several connected units. These units perform specific tasks: they control axes, or they control the communication with other networks.

These units can be stand-alone units, such as in the modular Trajexia system. Or they can be embedded in the device, for example in the MCH72 device, which is a single unit with integrated MECHATROLINK-II functionality and flexible axis functionality.

Units in Trajexia Studio are the representation of stand-alone units or embedded units in a device. Trajexia Studio supports the following units for the TJ1-MC04 and the TJ1-MC16 motion devices:

Unit	Full name	Description
CORT	TJ1-CORT CANopen Master Unit	Interface with CANopen network
DRT	TJ1-DRT DeviceNet Slave Unit	Interface with DeviceNet network
FL	TJ1-FL02 Flexible Axis Unit	Analogue controller for 2 axes
ML04	TJ1-ML04 MECHATROLINK-II Master Unit	MECHATROLINK-II controller for 4 axes
ML16	TJ1-ML16 MECHATROLINK-II Master Unit	MECHATROLINK-II controller for 16 axes
PRT	TJ1-PRT PROFIBUS-DP Slave Unit	Interface with PROFIBUS network

Trajexia Studio supports the following embedded units for the MCH72 motion device:

Unit	Functionality	Description
FL	Flexible axis functionality	Controller for 1 flexible axis
ML	MECHATROLINK-II functionality	MECHATROLINK-II controller for 31 axes

3-6-1 Adding a unit

Add unit To add a unit to a motion device, perform the following steps:

1 In the **Solution Explorer**, right-click **Units** under the motion device.
2 Select **Add *unit-to-add***.

Note You can only add units to the Trajexia devices, that is, the TJ1-MC04 and the TJ1-MC16. The MCH72 only has embedded units, and you cannot add units to it.

The Trajexia devices have restrictions on the number and type of units that can be added to it. These restrictions are that the system can have:

- 4 ML04 or ML16 units
- 3 FL units (for a TJ1-MC04)
  7 FL units (for a TJ1-MC16)
- 1 PRT unit or 1 DRT unit (but not both)
- 1 CORT unit

If you cannot add a unit to a motion device because of these restrictions, the corresponding menu item is disabled.

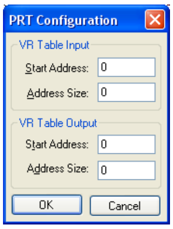
Note A Trajexia device must always have a TJ1-TER Terminator Unit as the last unit. This unit terminates the internal bus. Because the terminator unit does not control axes or communicate with other networks, it has no representation in Trajexia Studio.

3-6-2 DRT unit and PRT unit

The DRT unit and the PRT unit are slave units that exchange data between the device and other networks (the DeviceNet and the PROFIBUS networks respectively). To enable this data exchange, the device has two reserved areas of VR memory: one for data input from the slave unit, and one for data output to the slave unit. These VR memory areas must be set before the data can be exchanged. This is called the configuration of the DRT unit or the PRT unit.

Configure unit To configure the DRT unit or the PRT unit, perform the following steps:

1 In the **Solution Explorer**, right-click the DRT unit or the PRT unit that you want to configure.
2 Select **Configure**.
3 In the **DRT Configuration** window or the **PRT Configuration** window, enter the memory start address and the memory size for the VR input memory area and the VR output memory area in the text fields.



VR Table Input

Start Address: 0

Address Size: 0

VR Table Output

Start Address: 0

Address Size: 0

PRT Configuration window

4 Click **OK**.

When the device is built, Trajexia Studio adds code to the **'Omron Auto Generated - Units** section of the SHELL program to detect and configure the unit.

3-6-3 CORT unit

The CORT unit can be added to a motion device, but it has no functionality in Trajexia Studio. The CORT unit in a device can be supported with the BASIC command **CAN_CORT**.

3-6-4 FL unit

An FL unit can control 2 analogue drives. These drives are called A and B. When an FL unit is added to a motion device, the 2 drives A and B are available to associate with a flexible axis (refer to section 3-5-3).

3-6-5 MECHATROLINK-II slaves

MECHATROLINK-II is a motion network. It consists of a master unit and a number of slaves in serial connection to the master unit. A slave is an I/O or a drive. A drive is a servo or an inverter.
Trajexia Studio supports 3 MECHATROLINK-II Master Units: ML04, ML16 and embedded ML. You can add slaves to these ML units.

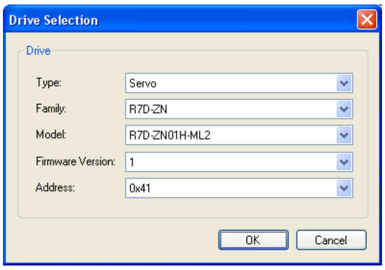
Add drive

To add a drive to an ML unit, perform the following steps:

1 In the **Solution Explorer**, right-click the ML unit to which you want to add a drive.

2 Select **Add Servo** or **Add Inverter**.

3 In the **Drive Selection** window, select the family, model and firmware version for the drive from the respective lists.



Drive

Type: Servo

Family: R7D-ZN

Model: R7D-ZN01H-ML2

Firmware Version: 1

Address: 0x41

Drive Selection window

4 In the **Address** list, select the MECHATROLINK-II address for the drive.

Note The addresses for servos range from 41 hex to 50 hex. The addresses for inverters range from 21 hex to 30 hex.
Trajexia Studio denotes hexadecimal addresses in the format 0x*nn*, where *nn* is the hexadecimal number.

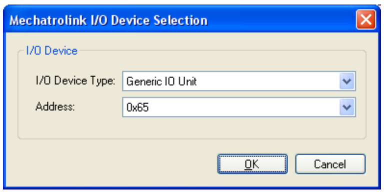
5 Click **OK**.

Add I/O To add an I/O to an ML unit, perform the following steps:

1 In the **Solution Explorer**, right-click the ML unit to which you want to add an I/O.

2 Select **Add I/O**.

3 In the **Mechatrolink I/O Device Selection** window, select an I/O type from the **I/O Device Type** list.



I/O Device Type: Generic IO Unit

Address: 0x65

Mechatrolink I/O Device Selection window

4 In the **Address** list, select the MECHATROLINK-II address for the I/O.

Note The addresses for servos range from 60 hex to 7F hex.

5 Click **OK**.

Reset network You can reset the MECHATROLINK-II network of an ML unit. This can be necessary when there is a communication problem. Resetting the network is done with a network initialisation, when the slaves in a MECHATROLINK-II network are detected and connected.

To reset the MECHATROLINK-II network, perform the following steps:

1 In the **Solution Explorer**, right-click the ML unit.

2 Select **Initialise MECHATROLINK-II Network**.

Note To reset the MECHATROLINK-II network, Trajexia Studio sends the BASIC command **MECHATROLINK(*unit-number*,0)** to the device.

3-6-6 Configuring a drive

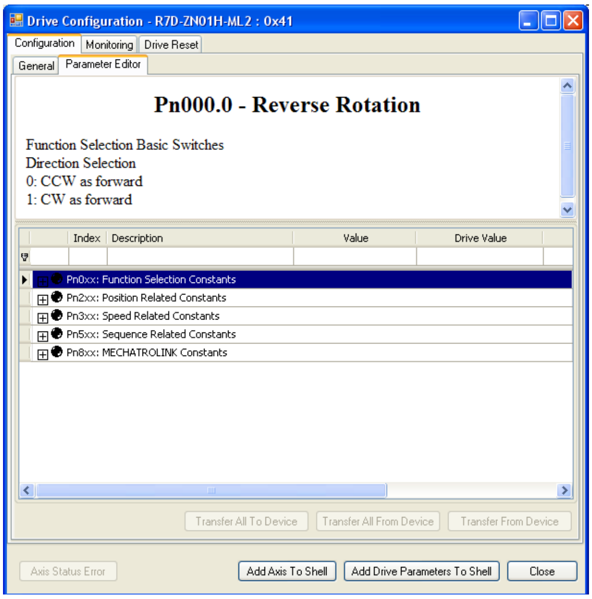
The **Drive Configuration** window groups together the following drive configuration operations and information:

- Change the drive parameters
- Transfer the drive parameters to and from the drive
- Monitor the drive
- Reset the drive
- Generate code for the SHELL program

To open the **Drive Configuration** window, right-click a drive in the Solution Explorer and select **Configure...**

3-6-6-1 Changing the drive parameters

The drive configuration consists of a set of drive parameters and their values. You can change the drive parameters in the parameter editor, which is on the **Parameter Editor** tab on the **Configuration** tab in the **Drive Configuration** window.



Pn000.0 - Reverse Rotation

Function Selection Basic Switches
Direction Selection
0: CCW as forward
1: CW as forward

Index	Description	Value	Drive Value
	Pn0xx: Function Selection Constants		
	Pn2xx: Position Related Constants		
	Pn3xx: Speed Related Constants		
	Pn5xx: Sequence Related Constants		
	Pn8xx: MECHATROLINK Constants		

Drive Configuration window, **Parameter Editor** tab

The parameter editor has the following elements:

- A text field, in the upper half, which shows the description of the selected parameter
- A filter row, marked with ▼, where you can set a filter on the parameter list
- The parameter list, in the lower half, which shows all parameters grouped together in categories

To change a parameter, enter a new value in the **Value** column of the selected parameter in the parameter list. If the value of the parameter is different from the default value, the circle in the first column shows in red ●.

3-6-6-2 Transferring the parameters

To transfer the drive parameters from Trajexia Studio to the drive or vice versa, the device must be online. The table below lists the transfer operations that Trajexia Studio supports for drive parameters. The third column in the table lists the steps to perform the transfer operation.

Parameters	Transfer direction	Steps
Selected	From Trajexia Studio to the drive	Change the value[1]
All	From Trajexia Studio to the drive	Click **Transfer All To Device**
Selected	From the drive to Trajexia Studio	Click **Transfer From Device**
All	From the drive to Trajexia Studio	Click **Transfer All From Device**

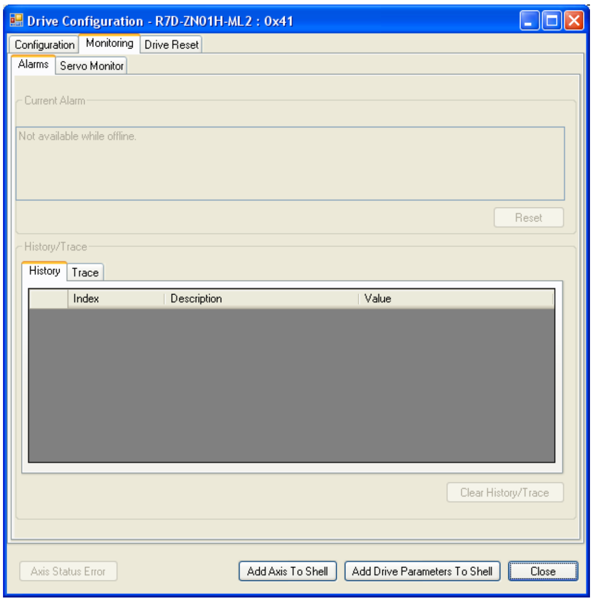
1. The changed value is transferred to the drive real-time.

3-6-6-3 Monitoring the drive

The **Monitoring** tab shows the real-time monitoring information of the drive when the device is online.

The **Alarms** tab shows the current alarm of the drive, if it has one. To clear the current alarm, click **Reset**.

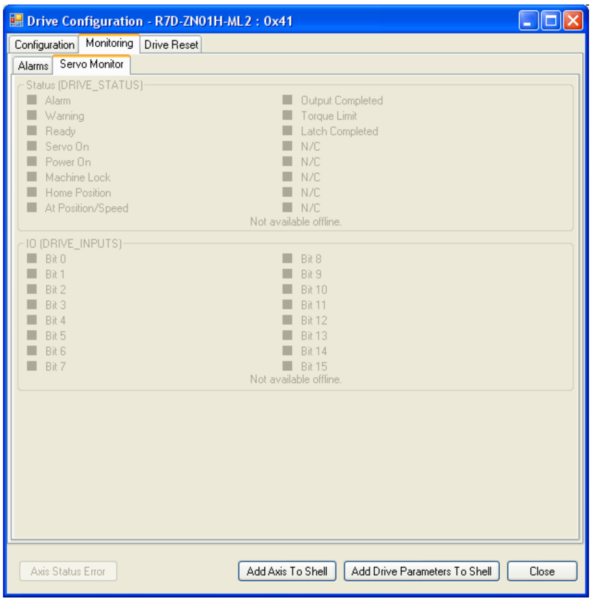
Previous alarms show in the lists in the **History** and **Trace** tabs. To clear the previous alarms, click **Clear History/Trace**.



Not available while offline.

Drive Configuration window, **Alarms** tab

The **Servo Monitor** tab shows the drive status and the I/O status. When an axis status bit or I/O bit is on, the corresponding square is red. When an axis status bit or I/O bit is off, the corresponding square is gray.



Not available offline.

Not available offline.

Drive Configuration window, **Servo Monitor** tab

Note If the axis that is associated with the drive has an error, the **Servo Monitor** tab shows when the user opens the **Drive Configuration** window.

3-6-6-4 Resetting the drive

Trajexia Studio supports 2 drive resets, which are listed in the table below. The second column in the table lists the corresponding buttons on the **Drive Reset** tab.

Drive reset	Button
Clear the alarm status of the drive	**Clear**

Drive reset	Button
Complete drive reset	**Reset**

3-6-6-5 Generating code for the drive parameters

You can add code to the SHELL program to initialise the drive parameters that have a non-default value, that is, drive parameters that are indicated with a red circle ● in the parameter list in the parameter editor. To add code for these drive parameters, click the **Add Drive Parameters To Shell** button in the **Drive Configuration** window.

The code for the initialisation of the drive parameter is added under **'Drive Parameters** in the **' Start Standard Section** outline of the SHELL program.

Note

- Trajexia Studio does not check if the code that you want to generate is already in the SHELL program. If you add code for the same drive parameters twice, the same drive parameter initialisations are in the SHELL code twice.
- You can change the added code.
- You can only generate code for the drive parameters when the device is offline.

3-7 Data traces

A data trace is a set of real-time values that are obtained by monitoring a parameter during a period of time and logging the parameter values. A parameter can be an axis parameters, a digital input or a digital output of a device.

Trajexia Studio can show the data trace values as a graph. This graph is useful to represent the motion in a device, and can help when you develop or debug a motion application. Also, Trajexia Studio can save the values in a file.

A data trace in Trajexia Studio consists of the following items:

- Four parameters that are monitored
- A sample period, which is the period of time during which the parameter values are logged
- A sample rate, which is the interval between logging 2 consecutive parameter values

Trajexia Studio can only log the data trace values when the device is online. For more information on online devices, refer to section 4-4.

Note

Trajexia Studio uses the Oscilloscope memory area to store the data trace values. You must make sure that the Oscilloscope memory area is large enough to store all values. The number of values is determined by the formula *sample-period* ÷ *sample-rate* × 4 (because a data trace monitors 4 parameters). For example, if the sample period is 10 s, and the sample rate is 2 ms (0.002 s), the size of the Oscilloscope memory area must be at least 10 ÷ 0.002 × 4 = 20,000. If it is less, Trajexia Studio shows an error message.

Refer to section 3-2-3-4 for information on changing the size of the Oscilloscope memory area.

3-7-1 Adding a data trace

Add data trace

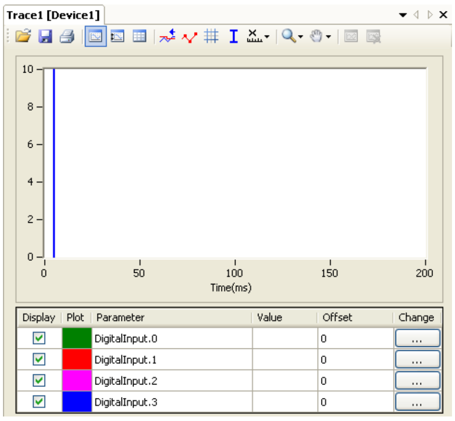
To add a data trace to a motion device, perform the following steps:

1 In the **Solution Explorer**, right-click **Data Traces** under the motion device.

2 Select **Add Data Trace**.
A new data trace is added under **Data Traces** in the **Solution Explorer**. By default, the 4 parameters are digital inputs 0 to 3.

3 Right-click the data trace in the **Solution Explorer**, and select **Open**.

The data trace window shows. This window has a toolbar, a graph view and a parameter list.

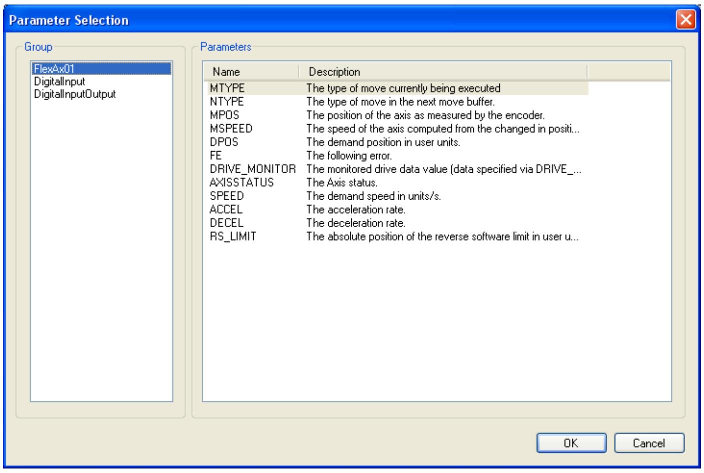


Display	Plot	Parameter	Value	Offset	Change
☑	🟩	DigitalInput.0		0	...
☑	🟥	DigitalInput.1		0	...
☑	🟪	DigitalInput.2		0	...
☑	🟦	DigitalInput.3		0	...

Data trace window

4 In the parameters list, click the ellipsis button in the **Change** column of the parameter that you want to change.

5 In the **Parameter Selection** window, select an axis or a set of digital inputs or outputs in the **Group** list.



FlexAx01
DigitalInput
DigitalInputOutput

Name	Description
MTYPE	The type of move currently being executed
NTYPE	The type of move in the next move buffer.
MPOS	The position of the axis as measured by the encoder.
MSPEED	The speed of the axis computed from the changed in positi...
DPOS	The demand position in user units.
FE	The following error.
DRIVE_MONITOR	The monitored drive data value (data specified via DRIVE_...
AXISSTATUS	The Axis status.
SPEED	The demand speed in units/s.
ACCEL	The acceleration rate.
DECEL	The deceleration rate.
RS_LIMIT	The absolute position of the reverse software limit in user u...

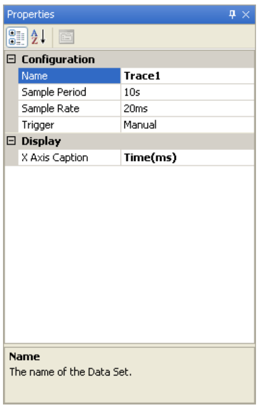
Parameter Selection window

6 In the **Parameters** list, select an axis parameter or a digital input or output.

7 Click **OK**.

8 Repeat the 4 previous steps for the other parameters that you want to change.

9 In the **Properties** window, set the sample period of the data trace with the **SamplePeriod** property.



Properties

Configuration

Name	**Trace1**
Sample Period	10s
Sample Rate	20ms
Trigger	Manual

Display

X Axis Caption	**Time(ms)**

Name
The name of the Data Set.

Data trace properties

10 Set the sample rate with the **SampleRate** property.

3-7-2 Logging the data trace values

The data trace starts logging the values when it is triggered to start. There are 2 methods to trigger a data trace to start logging the values:

- Manual
- Programmable

A programmable trigger is equivalent to the BASIC command **TRIGGER**. If you want to use a programmable trigger, you must include the **TRIGGER** command in a program, and set the **Trigger** property to **Programmable**.

Manual trigger

To trigger the data trace to start logging values manually, perform the following steps:

1 Set the value of the **Trigger** property to **Manual**.
2 Right-click the data trace in the **Solution Explorer**.
3 Select **Start Logging**.

The data trace stops logging the values when the sample period has elapsed. When you want the data trace to stop logging before the sample period has elapsed, you can right-click the data trace in the **Solution Explorer** and select **Stop Logging**.

3-7-3 Viewing the graph

When the data trace logs the values, Trajexia Studio can show a graph of these values. The graph shows in the graph view of the data trace window.
The data trace window has many layout features to change the layout of the graph view. The table below lists the layout features. The second column lists the corresponding steps.

Feature	Steps
Show the graph as a standard graph (that is, not a scaled graph)	Click ▦ on the toolbar
Show the graph as a scaled graph, in which the values on the Y axis are scaled to a selected axis	Click ▦ on the toolbar
Show or hide the graph for a parameter	Select or clear the check box in the **Display** column of the parameter in the parameter list

Feature	Steps
Move the graph for a parameter up or down along the Y axis (only for standard graphs)	Change the value in the **Offset** column of the parameter list
Show an axis for the Y axis scale (only for scaled graphs)	Select the checkbox in the **Y Axis** column in the parameter list
Change the minimum and maximum values for a scaled Y axis (only for scaled graphs)	Change the values in the **Y Axis Min** and **Y Axis Max** columns in the parameter list
Select or clear the automatic adjustment of the X axis scale as the graphs are drawn in the graph view	Click on the toolbar
Show or hide the markers that indicate points on the graph	Click :on the toolbar
Show or hide the grid lines	Click # on the toolbar
Show or hide the graph cursor	Click I on the toolbar
Change the X axis scale	Click on the toolbar and select a value for the X axis scale (which can be the time or one of the 4 parameters)
Zoom in	Click 🔍 on the toolbar, select **Zoom In** and drag the mouse pointer over the area in the graph view that you want to zoom in to
Zoom out	Click 🔍 on the toolbar and select **Zoom Out** to zoom out one step, or select **Zoom Reset** to return to the default zoom setting
Select the pan mode, in which only a portion of the graphs shows, which can be dragged across the graph view	Click ✋ on the toolbar and select **Pan On**
Clear the pan mode, so that the graph view shows the entire graphs	Click ✋ on the toolbar and select **Pan Off**

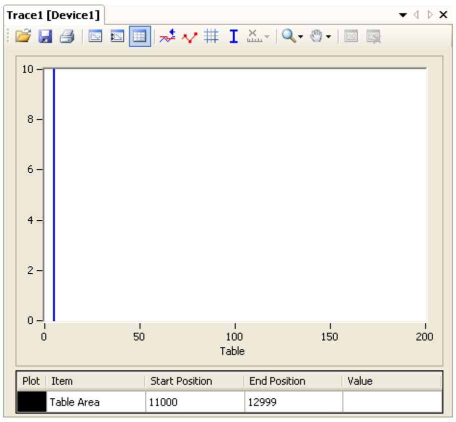
3-7-3-1 Load data

The graph that a data trace window shows is drawn from the data trace values. However, you can also load values from 2 other sources in the data trace window to draw a graph:

- From a CSV file that contains previously saved data trace values

  To load data trace values from a CSV file, click the 📂 button on the toolbar and select a CSV file.

- From a Table memory area
  To load values from a Table memory area, click the button on the toolbar. The parameter list in the data trace window is replaced by a Table list, in which you can enter the start address and the end address of the Table memory area.



Plot	Item	Start Position	End Position	Value
■	Table Area	11000	12999	

Data trace window with Table list

Note

To return to the standard view of a data trace graph, click the button on the toolbar. The Table list is replaced by the parameter list.

3-7-4 Saving the data trace values

You can save the data trace values in a CSV file, with the button on the data trace window toolbar.

Caution
A data trace is part of the solution, and thus it is saved when the solution is saved. However, the data trace values are not saved in the solution file. You must save the values separately in a CSV file.

3-8 General operations in the Solution Explorer

The **Solution Explorer** shows the items in the solution in a hierarchical way. The root item is **Devices**. Under **Devices**, the devices in the solution are listed.

Some items in the **Solution Explorer** are fixed. They cannot be added, changed or deleted. Some items are variable. They can be added, changed and deleted. The table below lists the variable items in the **Solution Explorer** and the parent items under which they show.

Parent item	Variable item
Devices	Device
CAM Tables	CAM table
Programs	Program
Axes	Axis
Units	Unit
Data Traces	Data trace
ML unit	MECHATROLINK-II slave

Trajexia Studio supports the following general operations on variable items in the **Solution Explorer**:

- Change order
- Delete
- Rename
- Cut, copy, paste
- Copy with drag-and-drop

Change order

You can change the order of solution items as follows: you can move a solution item in the **Solution Explorer** in front of another solution item.
To move *solution-item-1* in front of *solution-item-2*, perform the following steps:

1 In the **Solution Explorer**, select *solution-item-1*.
2 Drag *solution-item-1* on *solution-item-2* in the **Solution Explorer**.

To move *solution-item* to the back of the order, perform the following steps:

1 In the **Solution Explorer**, select *solution-item-1*.
2 Drag *solution-item-1* on its parent item in the **Solution Explorer**.

The order of all solution items except units in the solution explorer does not matter. You can change the order according to your preferences, but it does not have an influence on the solution.

The order of units is determined by their unit numbers. When you change the order of units, their unit numbers are also changed.

The order of axes is synchronised with the order of the addresses of the associated drives and the unit number of the units to which the drives are connected. If axes are associated with drives, you cannot change their order. And if you change the order of drives, the association with the axes is lost. For more information on associating an axis with a drive, refer to section 3-5-3.

Delete

To delete a solution item from the solution, perform the following steps:

1 In the **Solution Explorer**, select the solution item.
2 On the **Edit** menu, select **Delete**.

Rename

Some solution items have a name. Trajexia Studio assigns the name automatically when you add this solution item to the solution. This name is *item-n*, where *n* is a sequence number. You can change the name of a solution item. This makes it easy to identify and find a solution item in the solution.
To change the name of a solution item, perform the following steps:

1 Right-click the solution item in the **Solution Explorer**.
2 Select **Rename**.
3 Type the new name for the solution item.
4 Press ENTER.

Note

The following rules apply for the name of a solution item:

- A valid name consists of letters, digits and underscores. Other characters are not allowed.
- The first character cannot be a number.

Cut, copy, paste

Trajexia Studio supports the clipboard operations cut, copy and paste for solution items.
To cut or copy a solution item, perform the following steps:

1 Select the solution item in the **Solution Explorer** that you want to cut or copy.
2 On the **Edit** menu, select **Cut** or **Copy**.

To paste a solution item, perform the following steps:
1 Select the solution item in the **Solution Explorer** under which you want to insert the contents of the clipboard.
2 On the **Edit** menu, select **Paste**.

Note Trajexia Studio checks If the contents of the clipboard can be pasted under the selected solution item. If it cannot be pasted, the **Paste** menu item is disabled.
For example, if the clipboard contains a device, and the selected solution item in the **Solution Explorer** is **CAM Tables**, the **Paste** menu item is disabled, because a device cannot be added under **CAM Tables**.

Note When you copy and paste a MECHATROLINK-II slave, Trajexia Studio shows the **Drive Selection** window (for drives) or the **Mechatrolink I/O Device Selection** window (for I/Os), where you must select a new address for the copied slave. If you do not select a new address, Trajexia Studio cannot paste the slave, because all slaves must have a different address.

Note When you copy and paste an axis that is associated to a drive, the copied axis does not have a drive association, because 2 axes cannot be associated with the same drive.

Copy with drag-and-drop

With a drag-and-drop operation, you can copy a solution item in the **Solution Explorer** without using the clipboard.
To copy a solution item, perform the following steps:
1 Select the solution item in the **Solution Explorer** that you want to copy.
2 Press and hold down the CTRL key.
3 Drag the solution item to a solution item under which you want to insert the copy.

SECTION 4
Work online

4-1 Introduction

In Trajexia Studio, you can design and develop motion applications for one or more devices. If these devices are connected to the computer, Trajexia Studio can communicate real-time with them. It can transfer programs and data from and to the device, and it can execute programs in the device.

4-2 Hardware connection

A device can support several connection methods, such as Ethernet, serial or USB. Trajexia Studio supports all these connection methods for the supported devices.
Refer to the manual of the connected device for more information on the following topics:

- The supported connection methods
- The network settings of your computer if you use an Ethernet connection

4-3 Online concepts

When Trajexia Studio communicates real-time with a connected device, the device is called online. The device in the solution in Trajexia Studio is the interface via which Trajexia Studio communicates with the online device. A device that is not online is called offline.

When a device is online, Trajexia Studio can make the contents of the device in the solution match with the contents of the online device. This is called synchronisation. Synchronisation is discussed in section 4-5.

When a device is online, you can transfer the motion application you developed to the device, and you can view and edit the values of variables and memory areas in the device (refer to section 4-6). Also, you can execute and debug the programs in the device. For information on how to debug a program, refer to section 5.

Trajexia Studio can also perform the following actions on an online device:

- Perform a motion stop on all motion axes
- Lock the device
- View the communication channels
- Make a back-up of the memory in the device
- Update the firmware of the device
- Compare programs on the online device and in the solution
- Reset the device
- Debug a program
- Send commands with the Terminal window
- Monitor the communication channels

These actions are described in sections 4-7 and 5.

4-4 Going online

Go online To go online to a device, perform the following steps:

1 Make sure that the device is connected to the computer on which Trajexia Studio runs.

2 Set the communication settings for the device (see section 3-2-3).
3 Select the device in the **Solution Explorer**.
4 On the **Online** menu, select **Work Online/Offline**.

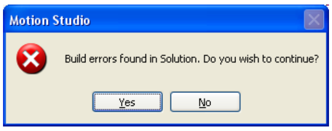
Trajexia Studio builds the device and synchronises it with the connected device (see section 4-5). When the build and the synchronisation are successful, the device is online.

When a device is online, an online icon shows on the device in the **Solution Explorer**. Also, the status bar indicates that the selected device is online.

To go offline, select the online device in the Solution Explorer and select **Work Online/Offline** on the **Online** menu.

4-5 Synchronisation

Before Trajexia Studio synchronises a device, it builds the device. If there are build errors, Trajexia Studio asks for your confirmation before it continues, because build errors can lead to unexpected behaviour of the device.



Motion Studio

Build errors found in Solution. Do you wish to continue?

Confirm build errors before synchronisation

You usually want to correct the build errors before you synchronise the device.

Trajexia Studio supports synchronisation in 2 directions:

- To device
  This synchronisation transfers the device in the solution to the online device.
- From device
  This synchronisation transfers the online device to the device in the solution.

Trajexia Studio supports 2 types of synchronisation: simple and advanced.

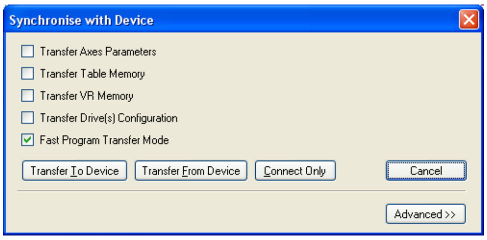
4-5-1 Simple synchronisation

The simple synchronisation transfers the following items:

- Programs
- Axes
- Units

You can also synchronise the following items, if you select the corresponding check boxes in the simple **Synchronise with Device** window:

- Axis parameters
- Table memory
- VR memory
- Configuration of drive(s)



□ Transfer Axes Parameters
□ Transfer Table Memory
□ Transfer VR Memory
□ Transfer Drive(s) Configuration
☑ Fast Program Transfer Mode

Simple **Synchronise with Device** window

The simple synchronisation supports the Fast Program Transfer mode. In this mode, all programs are deleted from the device in the solution or in the online device before all programs are transferred. To synchronise programs in the Fast Program Transfer mode, select the **Fast Program Transfer Mode** check box .

Click **Transfer To Device** to transfer from the device in the solution to the online device. Click **Transfer From Device** to transfer from the online device to the device in the solution. Click **Connect Only** to keep the device online without synchronising items.

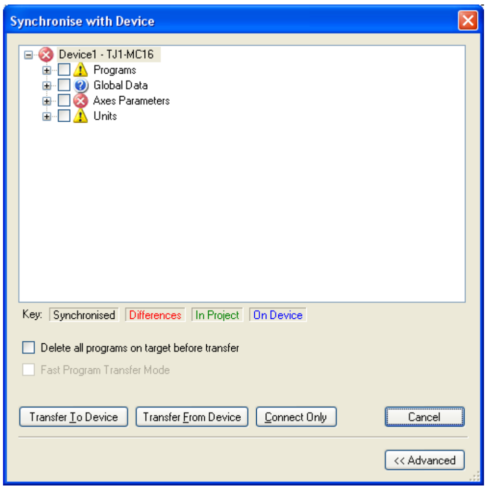
Note If you transfer the axes from an online device to the device in the solution, all axes are transferred, even the axes that are not used. Axes that are not used are virtual axes, and Trajexia Studio cannot distinguish between unused virtual axes and used virtual axes. Therefore, also unused virtual axes are transferred to the device in the solution. You can remove unused axes in the device in the solution when the device is offline.

4-5-2 Advanced synchronisation

The advanced synchronisation offers more flexibility and features than the simple synchronisation.

- The advanced synchronisation gives detailed feedback on the matches and differences between the device in the solution and the online device.
- You can specify which items to synchronise on a detailed level.

The advanced synchronisation is done from the advanced **Synchronise with Device** window. To show this window, click **Advanced >>** in the simple **Synchronise with Device** window.



□ Delete all programs on target before transfer
□ Fast Program Transfer Mode

Advanced **Synchronise with Device** window

The advanced **Synchronise with Device** window shows a tree view of the device and the programs, the VR memory and the Table memory (under **Global Data**), the axes, the units and the drives (under the units to which they are connected). The tree view uses colours to indicate the matches and the differences between the device in the solution and the online device, and icons to indicate the synchronisation status.

Colour	Label	Description
Black	Synchronised	The items in the device match
Red	Differences	The items in the device are different
Green	In Project	The item exists in the device in the solution, but not in the online device
Blue	On Device	The item exists in the online device, but not in the device in the solution

Note The **Solution Explorer** uses the colours black, red and green to indicate the matches and differences of the items under a device in the solution and the online device.

Note If a program in the device in the solution and the corresponding program on the online device are different, you can compare the programs with a viewer that shows the differences. Refer to section 4-7-6.

Icon	Synchronisation status	Description
✓	Synchronised	The items are synchronised
?	Not checked	Trajexia Studio has not checked if the items are synchronised
⚠	Warning	The item exists in the online device, but not in the device in the solution, or vice versa
✖	Error	The items are different

Each item has a check box. Select the check box to include the item in the synchronisation. If you select or clear the check box of an item in the tree that has child items, all the check boxes of the child items are selected or cleared.

The advanced synchronisation also supports the Fast Program Transfer mode, in which all programs are deleted from the device in the solution or in the online device before all programs are transferred. To synchronise programs in the Fast Program Transfer mode, perform one of the following actions:

- Select the **Fast Program Transfer Mode** check box.
- Select the **Programs** check box in the tree view, and select the **Delete all programs on target before transfer** check box.

Note The **Fast Program Transfer Mode** check box is enabled when the **Programs** check box is selected in the tree view and the **Delete all programs on target before transfer** check box is selected. In all other situations, the **Fast Program Transfer Mode** check box is disabled.

Click **Transfer To Device** to transfer from the device in the solution to the online device. Click **Transfer From Device** to transfer from the online device to the device in the solution. Click **Connect Only** to keep the device online without synchronising items.

Note If a program is running, you cannot transfer it to the online device.

Note Sometimes you cannot transfer a program to an online device when another program is running. This depends on the device. If you cannot transfer a program because another program is running, Trajexia Studio shows a message.

4-5-3 Transferring a program to the device

Trajexia Studio offers a quick way to transfer a program to the device when the device is online. This is useful when you debug a program: you can change some code in the program, and then quickly transfer it to the device without doing a full synchronisation.

To transfer a program to the device, select **Transfer Program To Device** on the **Online** menu.

Note The menu item **Transfer Program To Device** is disabled when the selected program is not changed.

4-6 Monitoring

Trajexia Studio supports the monitoring mode for online devices. When the monitoring mode is on, Trajexia Studio monitors the values of variables and memory areas in the online device. When the monitoring mode is off, Trajexia Studio shows only the values of variables and memory areas that are set in the solution. They can be different from the values in the online device, if the device is not synchronised.

On the other hand, if a device is online and the monitoring mode is on, Trajexia Studio does an automatic background synchronisation: the following items are transferred from the online device to the device in the solution:

- Table memory
- CAM tables
- VR memory
- Axis parameters

An important limitation of the automatic background synchronisation is the following: the items are only transferred from the online device to the device in the solution if they are visible in Trajexia Studio. For example, if a device is online, the monitoring mode is on, and the **VR Memory** window of the device is open and has the focus, only the VR variables that are visible in the **VR Memory** window are transferred from the online device to the device in the solution. When you scroll the **VR Memory** window up or down, another set of VR variables is visible and then these variables are automatically synchronised in the background.[1]

[1] The reason for this implementation is efficiency. Because synchronisation of large amounts of data (such as the complete Table memory) is slow, only the visible part is transferred.

Note When the user is entering text in an edit field, Trajexia Studio does not show the new values of variables that are updated by the background synchronisation.
When the user finishes or cancels entering text in an edit field, the updated values of the variables are shown in Trajexia Studio.

The automatic background synchronisation also works in the other direction, from the device in the solution to the online device. When you change a value in the Table memory, a CAM table, the VR memory or an axis parameter in the device in the solution, this value is automatically transferred to the online device.

Note The following values are read-only and cannot be changed when the device is online:

- Constants
- The global name of a VR variable

To set the monitoring mode on or off, select **Monitor** on the **Online** menu

4-7 Other online actions

4-7-1 Stopping motion

Trajexia Studio has a motion stop function. This function stops all motion on all axes connected to a device, and stops all programs that are running in the device.

⚠ **Caution**
The motion stop function in Trajexia Studio is not a safety function and it is not an emergency stop. When you must perform an emergency stop on a device, use the physical emergency stop on the device.

Stop motion To stop all motion on a device, perform the following steps:

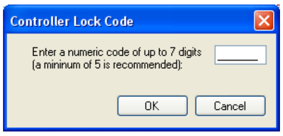
1. In the **Solution Explorer**, select the corresponding device.
2. On the **Online** menu, select **Stop Motion**.

4-7-2 Locking

Trajexia Studio can lock a device. If a device is locked, you cannot go online. This is useful when you want to disable access to the device for unauthorised users.

Lock device To lock a device, perform the following steps:

1. In the **Solution Explorer**, select the device.
2. On the **Online** menu, select **Lock Device**.
3. Click **Yes** in the **Lock Device** dialog box.
4. In the **Controller Lock Code** window, type a lock code in the text field.



Enter a numeric code of up to 7 digits
(a mininum of 5 is recommended):

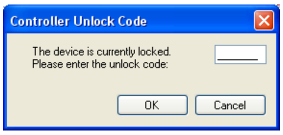
Controller Lock Code window

5. Click **OK**.
6. Enter the lock code again in the **Controller Lock Code** window to confirm the lock code.
7. Click **OK**.

Caution
If you forget the lock code, you cannot unlock the device. You must send it to the distributor or the manufacturer to remove the lock.

When you lock the device, it goes offline.

If you want to go online to a locked device, you must unlock it. Enter the lock code in the **Controller Unlock Code** window and click **OK**.



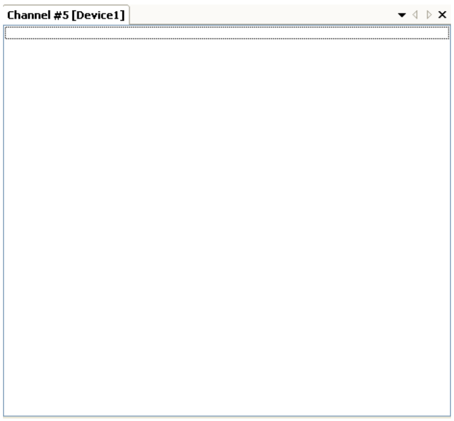
Controller Unlock Code

The device is currently locked.
Please enter the unlock code:

Controller Unlock Code window

4-7-3 Channels

A device has a number of communication channels. A communication channel is a direct communication connection to the device.
Channel 0 is reserved for the Terminal window (see section 5-5). Channels 5, 6 and 7 are user communication channels. You can use these channels to communicate with programs that are running in the device. Trajexia Studio can show the communication on these channels in a channel window. To open a channel window, select the online device in the **Solution Explorer** and select **Channel #*n*** under **Channels** on the **View** menu.



Channel #5 window

The channel window is a simple text output window that shows the communication. You can right-click on the channel window to copy the text to the clipboard or clear the contents of the window.

Note Trajexia Studio can only show the communication on the user channels 5,6 and 7. It cannot send commands via these channels. To send commands via the user channels, you need an input device connected to the device, such as a keypad.

4-7-4 Memory back-up

Programs and data are stored in the RAM memory of a device. If the device also has an EPROM memory, Trajexia Studio can copy the programs (but not the data) from the RAM memory to the EPROM memory and vice versa. This is useful because the device copies the programs in the EPROM memory to the RAM memory when it is switched on, to make sure that the RAM memory has the correct programs.

To copy the programs from the source memory of a device to the target memory, select the corresponding online device in the **Solution Explorer**, and select one of the following commands on the **Online** menu:

Command	Source memory	Target memory
Fix Into EPROM	RAM	EPROM
Recover From EPROM	EPROM	RAM

If you copy the programs, Trajexia Studio performs the following actions:

- All programs in the source memory are copied to the target memory.

Note Existing programs in the target memory are overwritten.

- The device is restarted.
- The connection between Trajexia Studio and the online device is broken.
- The copied programs with a startup priority are automatically started at power up or after a software reset of the device.

4-7-5 Firmware update

Trajexia Studio can update the firmware version of a connected device. This replaces the existing firmware of the device with new firmware.

Caution
If you update the firmware version of a device, the existing firmware is overwritten.

Caution
Before you update the firmware version of a device, make sure that you do the following:

- Stop the axes connected to the device.
  When the firmware update is in progress, control of the axes and I/Os is not possible.
- Save the programs in the device.
  The firmware update deletes all programs in the RAM and EPROM of the device.

Caution
Do not close Trajexia Studio when the firmware update is in progress. This can lead to unexpected results.

Note To update the firmware fo the connected device, this device must be offline. The device goes online automatically during the firmware update.

Update firmware

To update the firmware of a device, perform the following steps:

1. Select the device in the **Solution Explorer**.
2. On the **Online** menu, select **Download Firmware**.
3. In the **Load New Firmware** dialog box, click **Yes**.
4. In the **Select New Firmware File** dialog box, select a firmware file and click **OK**.

Note Firmware files for TJ1-MC04 and TJ1-MC16 devices have the file extension .out.
Firmware files for MCH72 devices have the file extension .bin.

The device goes online, and the firmware version of the device is updated with the firmware version from the selected firmware file.

Note The firmware update can take several minutes.

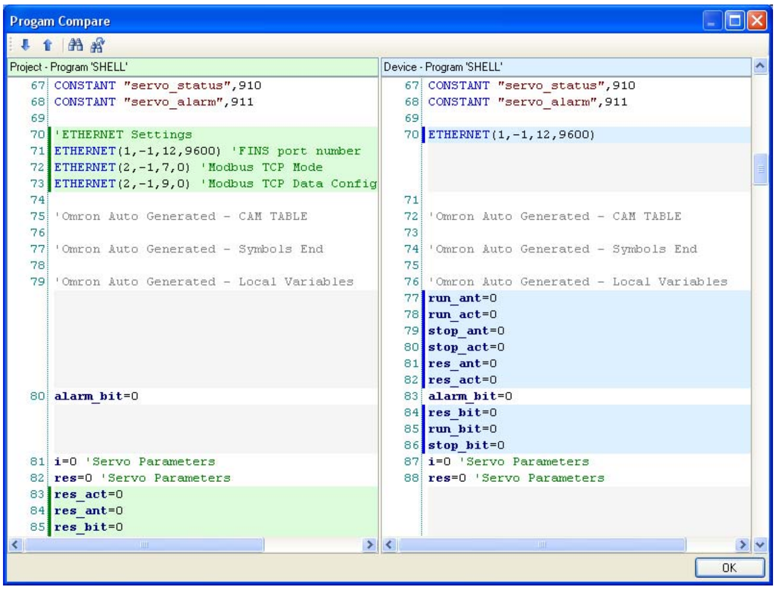
4-7-6 Comparing programs

Trajexia Studio has a viewer that shows the differences between a program in the device in the solution and the corresponding program on the online device: the **Program Compare** window.

Compare programs To open the **Program Compare** window, perform the following steps:

1 Select the program in the **Solution Explorer**.

2 On the **Online** menu, select **Program Compare**.



Project - Program 'SHELL'

```
CONSTANT "servo_status",910
CONSTANT "servo_alarm",911

'ETHERNET Settings
ETHERNET(1,-1,12,9600) 'FINS port number
ETHERNET(2,-1,7,0) 'Modbus TCP Mode
ETHERNET(2,-1,9,0) 'Modbus TCP Data Config

'Omron Auto Generated - CAM TABLE

'Omron Auto Generated - Symbols End

'Omron Auto Generated - Local Variables
alarm_bit=0
i=0 'Servo Parameters
res=0 'Servo Parameters
res_act=0
res_ant=0
res_bit=0
```

Device - Program 'SHELL'

```
CONSTANT "servo_status",910
CONSTANT "servo_alarm",911

ETHERNET(1,-1,12,9600)

'Omron Auto Generated - CAM TABLE

'Omron Auto Generated - Symbols End

'Omron Auto Generated - Local Variables
run_ant=0
run_act=0
stop_ant=0
stop_act=0
res_ant=0
res_act=0
alarm_bit=0
res_bit=0
run_bit=0
stop_bit=0
i=0 'Servo Parameters
res=0 'Servo Parameters
```

Program Compare window

Note The menu item **Program Compare** is disabled if the program in the device in the solution and the program on the online device are not different. If the programs are not different, they do not show in red.

Note The menu item **Program Compare** is also available in the shortcut menu of a program in the **Solution Explorer**, and in the shortcut menu of a program in the tree in the Advanced **Synchronise with Device** window.
Again, if the programs are not different, they do not show in red, and the **Program Compare** menu item is disabled.

The **Program Compare** window shows the 2 programs in 2 windows that are tiled horizontally, where matching lines are at the same horizontal position. This viewer makes it very easy to see the differences between the programs instantly. The program in the device in the solution is on the left, the program in the online device is on the right.

The table below lists the background colours that the **Program Compare** window uses to mark the differences.

Background colour	Difference
Green (left window) or blue (right window)	The code is present and different

Background colour	Difference
Gray	The code is not present

The table below lists the toolbar buttons of the **Program Compare** window and the corresponding description.

Toolbar button	Description
⬇	Navigate to the next difference
⬆	Navigate to the previous difference
🔍	Search for a find string in the left or the right window[1]
🔍↻	Repeat the last search

1. If you want to search for text in the left (or right) window, click in the left (or right) window, and then click the 🔍 or the 🔍↻ button. The title bar of the **Find** window shows in which window the search is performed. The search is performed from the cursor position forward.

To close the **Program Compare** window, click the **OK** button.

4-7-7 Reset

Trajexia Studio can reset an online device. This is equivalent to the BASIC command **EX** or **EX(0)**. The reset is a software reset of the device.
To reset an online device, select the device in the **Solution Explorer**, and select **Reset Device** on the **Online** menu.

SECTION 5 Debugging

5-1 Introduction

Trajexia Studio has an integrated debugger. With the debugger, you can observe the behaviour of BASIC programs when they are running and find logical or semantic errors in the programs.

Note The debugger functions are only available when Trajexia Studio is online.

The Trajexia Studio debugger has the following tools:

- Breakpoints
- Execution control
- Watches
- Terminal window

The debugger tools are only accessible for online devices.

5-2 Breakpoints

A breakpoint is a mark in the program code that causes the debugger to temporarily stop the execution of the program when it reaches the mark. When the execution of the program is stopped at a breakpoint, the program execution is said to break, and the program and the debugger are said to be in break mode. In break mode, the state of the program is frozen. The variables and I/O statuses remain in the memory, and their values can be examined. The program can resume execution from the break mode at any time.
A breakpoint is indicated with a dark red background colour in the code editor, and with a breakpoint icon in the margin bar of the code editor. A breakpoint can have 2 states: enabled or disabled.

Icon	State	Description
●	Enabled	An enabled breakpoint is an active breakpoint. It causes the program execution to break.
○	Disabled	A disabled breakpoint is inactive. The program execution does not break at an inactive breakpoint. However, the breakpoint mark is still in the program code. A disabled breakpoint is useful if you do not want the program execution to break, but you want to keep the breakpoint mark in the program code, to enable it later.

The table below lists the actions that Trajexia Studio supports for breakpoints in the code editor. The second column lists the corresponding commands on the **Programs** menu, under **Breakpoints**.

Action	Command
Set an enabled breakpoint on the current line	**Toggle Breakpoint**
Clear the breakpoint from the current line	**Toggle Breakpoint**
Clear all breakpoints in the current program	**Delete All Breakpoints**
Enable all breakpoints in the current program	**Enable All Breakpoints**
Disable all breakpoints in the current program	**Disable All Breakpoints**
Navigate to the next breakpoint in the current program	**Next Breakpoint**
Navigate to the previous breakpoint in the current program	**Previous Breakpoint**

5-3 Execution control

The Trajexia Studio debugger supports the following functions to control the execution of a BASIC program:

- Start execution
- Stop execution
- Break execution
- Step through the program

5-3-1 Starting execution

Start/restart execution

To start the execution of a program or to restart the execution of a program in break mode, perform the following steps:

1. In the **Solution Explorer**, select the program.
2. On the **Programs** menu, select **Run**.

The program runs until it reaches a breakpoint. After this, the program is in break mode. To resume the execution from the break mode, restart the execution.
If the program has no breakpoints, the execution stops at the end of the program.

5-3-2 Stopping execution

If you stop the execution of a program, you stop the debugger. This is different from the break mode, in which the debugger is still active and from which you can resume the execution.

Stop execution

To stop the execution of a program, perform the following steps:

1. In the **Solution Explorer**, select the program.
2. On the **Programs** menu, select **Stop**.

5-3-3 Stepping through a program

If you step through a program, you execute one line of code at a time, after which the program is in break mode. To execute the next line of code, step again.

Step through program

To step through a program, perform the following steps:

1. Open the code editor window of the program.
2. Make sure that the program is in break mode.
3. On the **Programs** menu, select **Step**.

After each step, the program is in break mode. In break mode, you can continue stepping, or you can restart the execution (see section 5-3-1).

5-3-4 Navigating to the execution location

If a program is running, it can be useful to know which line of the program is currently executed.

Navigate to current line

To navigate to the line in the program that is executed, perform the following steps:

1. Open the code editor window of the program.
2. On the **Programs** menu, select **Goto Current Line**.

The cursor in the code editor window moves to the line of code that is currently executed, and the line is marked with a yellow background colour.

5-4 Watches

A watch or a watched variable is a real-time view of the value of a variable in a program. Watches are useful to debug a program when the program does not behave as expected.

The **Watch** window lists the defined watches. For more information on the **Watch** window, refer to section 2-4-2-8.

5-4-1 Adding a watch

There are 2 ways to add a watch to the **Watch** window:

- Select a variable in a code editor window and drag it to the **Watch** window
- Select a variable in the **Add Watch Item** window

Drag variable

To drag a variable from the code editor to the **Watch** window, perform the following steps:

1 Select a variable name in the code editor.

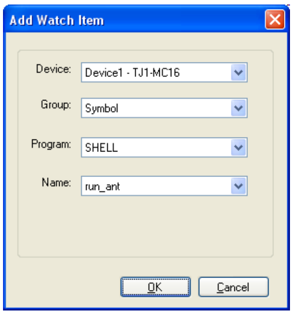
2 Drag the selected text from the code editor to the **Watch** window.

Note If the selected text is not a variable name, you cannot drop it in the **Watch** window.

Add watch

To select a variable in the **Add Watch Item** window, perform the following steps:

1 In the **Watch** window, click the **Add ...**



Device: Device1 - TJ1-MC16
Group: Symbol
Program: SHELL
Name: run_ant

Add Watch Item window

2 In the **Add Watch Item** window, select a device from the **Device** list.

3 Select the type of watch from the **Group** list. Select **Symbol** to add a watch for a variable in a program. Select **VR Table**, **Table**, **Constant**, **Global** or **IO** to add a watch for the corresponding global data item.

4 Select the program that contains the watch from the **Program** list. This list is only visible when **Symbol** is selected in the **Group** list.

5 Select the index of the variable to watch in the memory area from the **Index** list. This list is only visible when **VR Table**, **Table** or **IO** is selected in the **Group** list.

6 Select the I/O type of the watch (digital or analogue, input or output) from the **IO Type** list. This list is only visible when **IO** is selected in the **Group** list.

7 Select the name of the variable to watch from the **Name** list. This list is only visible when **Symbol**, **Constant** or **Global** is selected in the **Group** list.

8 Click **OK**.

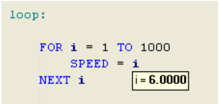
Note The values in the lists in the **Add Watch Item** window are refreshed after a build. For example, if you add a variable to a program, the variable does not show in the **Name** list in the **Add Watch Item** window until the program is built successfully.

5-4-2 Viewing a watch

To view the real-time value of a watch, Trajexia Studio must be in monitoring mode.

Trajexia Studio supports 2 ways to view the real-time value of a watched variable when a program is running:

- In the **Watch** window
  The **Value** column of the **Watch** window shows the value of the defined watches.
- In the code editor
  When you hover the mouse pointer over a variable in the code editor, a tooltip shows with the value of the variable.



```
loop:

    FOR i = 1 TO 1000
        SPEED = i
    NEXT i
```

View watch in the code editor

5-4-3 Changing the value of a variable

You can change the value of a variable in a running program. This is useful for debugging a program, because you do not have to change the program to have an influence on the program execution and results.

Change variable value

To change the value of a variable in a running program, perform the following steps:

1 Add a variable in the program to the **Watch** window (refer to section 5-4-1).
2 Execute the program.
3 Change the value in the **Value** column of the **Watch** window. Refer to section 2-3-1 for information on how to edit a value in a tabular view.

The value of the variable in the running program is changed. The program execution continues with the new variable value.

Note It is also possible to change the value in the Name column of a watch in the **Watch** window. However, this does not change the name of the watched variable.
If you change the name of a watch in the **Watch** window, and this name is not the name of a variable in the program, the watch is invalid. An invalid watch shows in red in the **Watch** window.

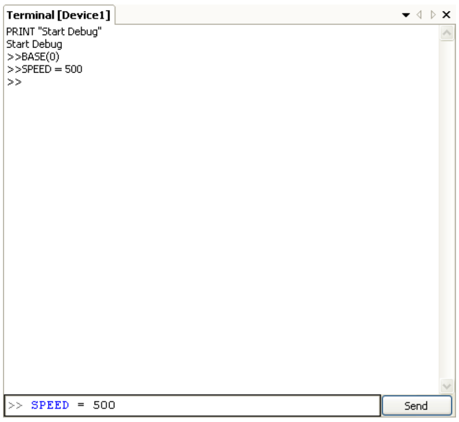
5-4-4 Deleting a watch

To delete a watch, select the corresponding row in the **Watch** window and click **Delete**.

5-5 Terminal window

The Terminal window is used for direct communication with the online device. In the Terminal window, you can execute BASIC commands in the device. This is useful for debugging, because you can execute commands immediately and do not have to write and transfer a BASIC program.

To open the Terminal window, select a device in the **Solution Explorer**, and select **Terminal Window** on the **View** menu.



```
PRINT "Start Debug"
Start Debug
>>BASE(0)
>>SPEED = 500
>>
```

```
>> SPEED = 500
```

Terminal window

The Terminal window has 2 parts:

- Output area
  This is the large text area that shows all commands executed in the device via the Terminal window (prefixed with >>), and the response from the device.
- Input field
  This is the text field at the bottom of the Terminal window where you type the commands to execute in the device. The input field supports some useful features, which are listed in the table below. The second column lists the corresponding keys.

Feature	Keys
Show a list of commands to insert	CTRL+SPACE (See section 2-2-4-2)
View the argument information for a command	CTRL+SHIFT+SPACE (See section 2-2-5-6)
Select the previous or the next command in the list of recent commands	ARROR UP or ARROW DOWN

To execute a command in the device, type the command in the input field, and click **Send** or press ENTER.

Caution
If you execute commands via the Terminal window, you can change the synchronisation status of the online device.

A Shortcut keys in the code editor

A-1 Navigation

Operation	Keys
Move the cursor to the first character of the current line	HOME
Move the cursor to the last character of the current line	END
Move the cursor to the first character in the code editor	CTRL+HOME
Move the cursor to the last character in the code editor	CTRL+END
Move the cursor to the previous token[1]	CTRL+ARROW LEFT
Move the cursor to the next token	CTRL+ARROW RIGHT
Move the cursor to the first visible line in the code editor	CTRL+PAGE UP
Move the cursor to the last visible line in the code editor	CTRL+PAGE DOWN
Move the cursor to the matching parenthesis	CTRL+]

[1] A token is a constructing element of the BASIC programming language, such as a command, an operator, a variable name or an expression.

A-2 Selecting text

To select text in the code editor, you can press SHIFT plus a navigation key (see section A-1). This operation selects the text from the current cursor position to the new cursor position.
The table below lists the other text selection operations.

Operation	Keys
Select all text in the code editor	CTRL+A
Select the current token	CTRL+SHIFT+W
Select a block of text from the cursor position up	SHIFT+ALT+ARROW UP
Select a block of text from the cursor position down	SHIFT+ALT+ARROW DOWN
Select a block of text from the cursor position to the left	SHIFT+ALT+ARROW LEFT
Select a block of text from the cursor position to the right	SHIFT+ALT+ARROW RIGHT
Select a block of text from the cursor position to the previous token	CTRL+SHIFT+ALT+ARROW LEFT
Select a block of text from the cursor position to the next token	CTRL+SHIFT+ALT+ARROW RIGHT

A-3 Finding text

Operation	Keys
Find and replace text	CTRL+F
Start an incremental search	CTRL+I
Do an incremental search backwards	CTRL+SHIFT+I

A-4 Changing text

Operation	Keys
Increase the line indent of the current line or the selected text	TAB
Decrease the line indent of the current line or the selected text	SHIFT+TAB

Operation	Keys
Change the character right of the cursor or the selected text to uppercase	CTRL+SHIFT+U
Change the character right of the cursor or the selected text to lowercase	CTRL+U
Toggle between inserting text or overwriting text	INSERT
Swap the characters left and right of the cursor	CTRL+T
Swap the token at the cursor position with the next token	CTRL+SHIFT+T
Swap the current line and the next line	CTRL+SHIFT+ALT+T

A-5 Deleting text

Operation	Keys
Delete the current line	CTRL+SHIFT+L
Delete the text from the cursor to the next token	CTRL+DEL
Delete the text from the previous token to the cursor	CTRL+BACKSPACE

A-6 Inserting a new line

Operation	Keys
Insert a new line above the current line	CTRL+ENTER
Insert a new line below the current line	CTRL+SHIFT+ENTER

A-7 BASIC commands and arguments

Operation	Keys
Insert a BASIC command	CTRL+SPACE
Show the argument information of a command	CTRL+SHIFT+SPACE

A-8 Clipboard and undo/redo operations

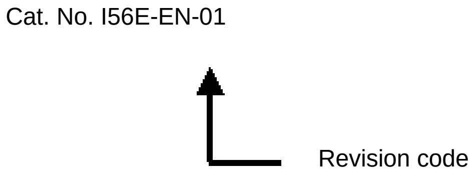
Operation	Keys
Undo the last operation	CTRL+Z
Redo the last operation	CTRL+Y
Cut the selected text to the clipboard	CTRL+X or SHIFT+DEL
Copy the selected text to the clipboard	CTRL+C or CTRL+INS
Paste the text on the clipboard	CTRL+V or SHIFT+INS
Cut the current line to the clipboard	CTRL+L

A-9 Scrolling

Operation	Keys
Scroll the contents of the code editor one line up	CTRL+ARROW UP
Scroll the contents of the code editor one line down	CTRL+ARROW DOWN

Revision history

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.



Cat. No. I56E-EN-01

Revision code

The following table lists the changes made to the manual during each revision. The page numbers of a revision refer to the previous version.

Revision code	Date	Revised content
01	August 2008	First version

OMRON

Authorized Distributor:

Note: Specifications subject to change without notice.

Printed in Europe