

JP1 Version 13

**Job Management: Getting Started (Client
Process Automation)**

3021-3-L55(E)

Notices

■ Relevant program products

For details on supported operating system versions and service packs, patches, and other prerequisites for the target product, see the *Release Notes*.

JP1/Client Process Automation (For Windows 11, Windows 10):

P-2A12-3BDL JP1/Client Process Automation version 13-00 or later

■ Trademarks

HITACHI and JP1 are either trademarks or registered trademarks of Hitachi, Ltd. in Japan and other countries.

Microsoft is a trademark of the Microsoft group of companies.

Microsoft, Windows are trademarks of the Microsoft group of companies.

Microsoft, Windows Server are trademarks of the Microsoft group of companies.

Other company and product names mentioned in this document may be the trademarks of their respective owners.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.

■ Microsoft product name abbreviations

This manual uses the following abbreviations for Microsoft product names.

Abbreviation	Full name or meaning
Windows 10	Windows(R) 10 Enterprise
	Windows(R) 10 Pro
Windows 11	Windows(R) 11 Enterprise
	Windows(R) 11 Pro

Windows is sometimes used generically, referring to Windows 11 and Windows 10.

■ Microsoft product screen shots

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

■ Restrictions

Information in this document is subject to change without notice and does not represent a commitment on the part of Hitachi. The software described in this manual is furnished according to a license agreement with Hitachi. The license agreement contains all of the terms and conditions governing your use of the software and documentation, including all warranty rights, limitations of liability, and disclaimers of warranty.

Material contained in this document may describe Hitachi products not available or features not available in your country.

No part of this material may be reproduced in any form or by any means without permission in writing from the publisher.



■ Issued

Sep. 2023: 3021-3-L55(E)

■ Copyright

Copyright (C) 2019, 2023, Hitachi, Ltd.

Copyright (C) 2019, 2023, Hitachi Solutions, Ltd.

Summary of amendments

The following table lists changes in this manual (3021-3-L55(E)) and product changes related to this manual.

Changes	Location
Windows 11 was added as an applicable operating system.	--

Legend:

--: Not applicable

In addition to the above changes, minor editorial corrections were made.

Preface

■ What you can do with Client Process Automation

Among the many business operations carried out by computers, some are performed regularly such as daily backups or month-end closing, while some are performed irregularly by events such as receiving orders.

Moreover, computer systems can be even more complicated with cloud services or data centers, where services and data are stored in virtual locations, as well as when different computers relate to one another to perform a set of operations. This, as a result, can impose a burden on services and system users who access the data.

Client Process Automation (simply called *CPA* hereafter) is designed to automate periodic or routine processing to help system users do their system operations in a more automated and simplified way.

■ What is explained in this manual

This manual explains the basic methods of configuring and operating CPA. The purpose of this manual is to help readers briefly understand the basic usage of CPA.

This manual is intended for the following users:

- Users who will be evaluating the installation of CPA
- Users who want to understand the concepts of CPA and to obtain in a short time an overview from configuration to operation

■ Organization of this manual

This manual is organized as follows:

1. Installation

This chapter describes how to install CPA.

2. Automating Applications by Using CPA

This chapter describes the basic methods of operating CPA.

A. Version Revisions

This appendix explains the version revisions.

B Reference Manual for This Manual

This appendix provides reference material for this manual, including a list of related manuals and a description of the conventions used by the writer.

■ Conventions: Fonts and symbols

The following table explains the text formatting conventions used in this manual:

Text formatting	Convention
Bold	Bold characters indicate text in a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example:

Text formatting	Convention
	<ul style="list-style-type: none"> From the File menu, choose Open. Click the Cancel button. In the Enter name entry box, type your name.
<i>Italic</i>	<p>Italic characters indicate a placeholder for some actual text to be provided by the user or system. For example:</p> <ul style="list-style-type: none"> Write the command as follows: <code>copy source-file target-file</code> The following message appears: <code>A file was not found. (file = file-name)</code> <p>Italic characters are also used for emphasis. For example:</p> <ul style="list-style-type: none"> Do <i>not</i> delete the configuration file.
Monospace	<p>Monospace characters indicate text that the user enters without change, or text (such as messages) output by the system. For example:</p> <ul style="list-style-type: none"> At the prompt, enter <code>dir</code>. Use the <code>send</code> command to send mail. The following message is displayed: <code>The password is incorrect.</code>

The following table explains the symbols used in this manual:

Symbol	Convention
	In syntax explanations, a vertical bar separates multiple items, and has the meaning of OR. For example: <code>A B C</code> means A, or B, or C.
{ }	In syntax explanations, curly brackets indicate that only one of the enclosed items is to be selected. For example: <code>{ A B C }</code> means only one of A, or B, or C.
[]	In syntax explanations, square brackets indicate that the enclosed item or items are optional. For example: <code>[A]</code> means that you can specify A or nothing. <code>[B C]</code> means that you can specify B, or C, or nothing.
. . .	<p>In coding, an ellipsis (. . .) indicates that one or more lines of coding have been omitted.</p> <p>In syntax explanations, an ellipsis indicates that the immediately preceding item can be repeated as many times as necessary. For example: <code>A, B, B, . . .</code> means that, after you specify A, B, you can specify B as many times as necessary.</p>

■ Conventions: Version numbers

The version numbers of Hitachi program products are usually written as two sets of two digits each, separated by a hyphen. For example:

- Version 1.00 (or 1.0) is written as 01-00.
- Version 2.05 is written as 02-05.
- Version 2.50 (or 2.5) is written as 02-50.
- Version 12.25 is written as 12-25.

The version number might be shown on the spine of a manual as *Ver. 2.00*, but the same version number would be written in the program as *02-00*.

Contents

Notices 2

Summary of amendments 4

Preface 5

1 Installation 8

1.1 Preparation before installation 9

1.1.1 Memory and disk space required for installation 9

1.1.2 Requirements 9

1.2 Installing a new instance 10

2 Automating Applications by Using CPA 11

2.1 Basic functions for automation 12

2.2 Creating a job 13

2.3 Checking job status 17

2.3.1 Rerunning a job 20

2.3.2 Forcibly terminating a job 23

2.4 Creating an action flow 26

2.4.1 Checking action status 34

2.5 Setting a schedule 44

2.6 Registering a job executed on event 46

Appendixes 49

A Version Changes 50

A.1 Changes in version 13-00 50

A.2 Changes in version 12-50 50

A.3 Changes in version 12-10 50

A.4 Changes in version 12-01 50

B Reference Material for This Manual 51

B.1 Related publications 51

B.2 Conventions: Abbreviations for product names 51

B.3 Conventions: Acronyms 51

B.4 Conventions: KB, MB, GB, and TB 51

Index 52

1

Installation

This chapter describes how to install CPA.

1.1 Preparation before installation

This section describes the preparations required before installing CPA.

1.1.1 Memory and disk space required for installation

For details about the amount of memory and disk space that is required for installation, see the *Release Notes*.

1.1.2 Requirements

- CPA must be installed on an NTFS file system.
- A host on which you want to install CPA must have a host name with a length of 128 bytes or less.

1.2 Installing a new instance

To install a new instance of CPA:

1. As a member of the Administrators group, log in to the host on which you want to install

2. Insert the CPA media.

The Hitachi integrated installer window appears.

3. Follow the instructions from the Hitachi integrated installer and specify information as necessary to install the software.

Specify the following information during installation:

- User name^{#1}

Enter the name of the CPA user. The length must not exceed 50 bytes.

- Organization^{#1}

Enter the name of your company or department. The length must not exceed 80 bytes.

- Installation folder^{#2}

Specify the absolute path to the folder in which you want to install CPA. The length, including a backslash (\) that is automatically added to the end of the path, must not exceed 80 bytes. The absolute path must point to a local directory.

If you do not specify any value, the following value is assumed:

system-drive\Program Files (x86)\HITACHI\JP1CPA^{#3}

- Data Folder^{#2}

Specify the absolute path to the folder in which you want to store CPA data. The length, including a backslash (\) that is automatically added to the end of the path, must not exceed 80 bytes. The absolute path must point to a local directory.

If you do not specify any value, the following value is assumed:

system-drive\ProgramData\HITACHI\jpl\jpl_default\JP1CPA

#1

Do not include the following characters in the user name or organization name:

- Symbols except for the following:

Space, left parenthesis ((), right parenthesis ()), plus sign (+), hyphen (-), and underscore (_)

#2

Ensure that the installation folder and data folder do not include the following characters or paths:

- Symbols except for the following:

Space, left parenthesis ((), right parenthesis ()), plus sign (+), hyphen (-), and underscore (_)

Note that you cannot include a space at the beginning or end of the path, before or after a backslash (\), or in succession with another space.

- Multibyte characters

- Paths to non-fixed drives or UNC paths

#3

If you are using a 32-bit operating system, replace *Program Files (x86)* with *Program Files*.

4. When the installation is completed, click the **Finish** button.

2

Automating Applications by Using CPA

This chapter describes basic procedures to use CPA to automate applications.

2.1 Basic functions for automation

CPA provides the following functions for you to automate applications.

Subsequent sections describe how to use the functions.

Creating a job

Create a job to run an action.

Checking a job

Check whether a started job finished successfully.

Creating an action flow

Create a sequence of multiple actions that you want to run as a single job.

Setting a schedule

Specify a date and time to run a job you created.

Registering a job executed on event

Specify a condition to trigger a job you created.

For details on detailed functions and settings in these procedures, see the manual for Client Process Automation, which is listed in [B.1 Related publications](#).

2.2 Creating a job

This section describes how to create a job with the actual GUI.

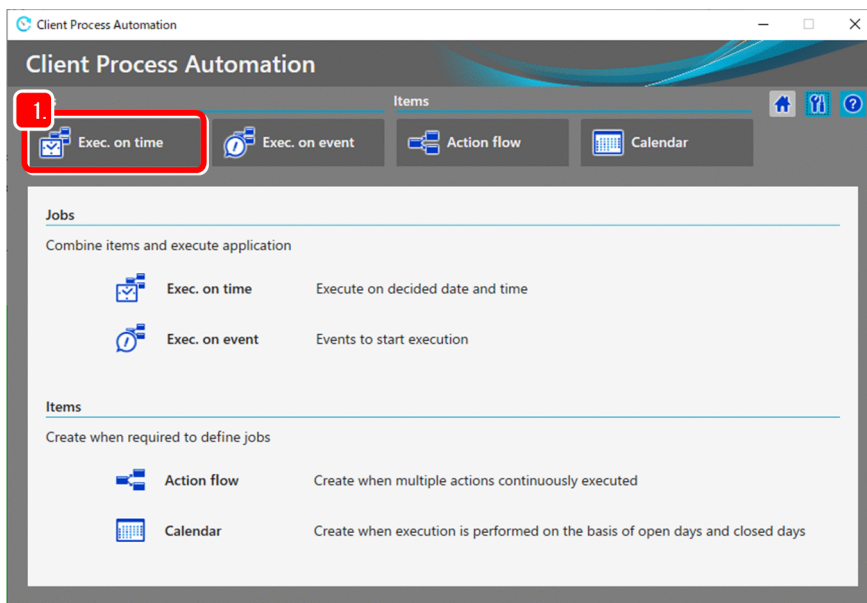
As an example, let's create a job that opens a message dialog box on the screen to display the string `hello!!`.

This example uses the `msg` command (`msg.exe console /w hello!!`) to open the message dialog box.

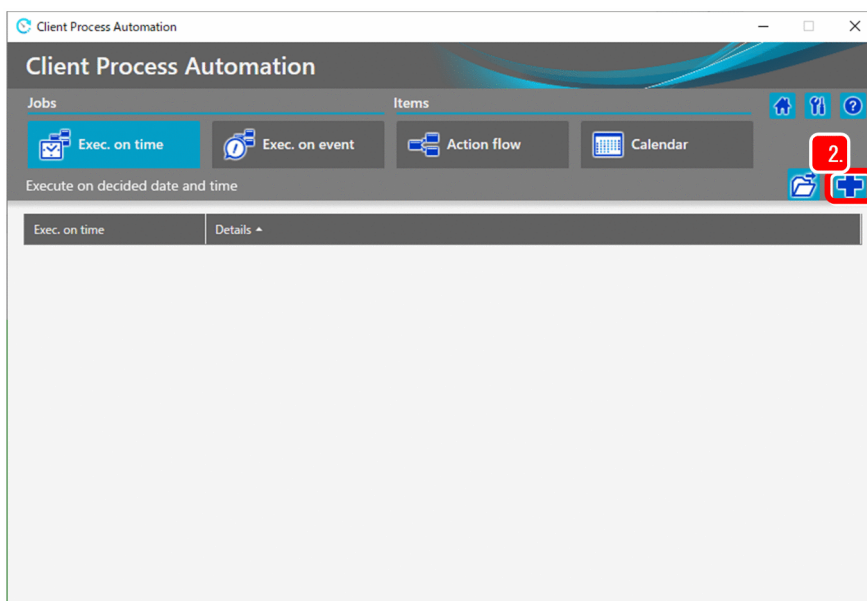
On a computer with CPA installed, from the program menu, select **JP1 Client Process Automation** and **RegistJob**. The Job Design View appears.

Procedure

1. Select the **Exec. on time** tab.



2. Click the **+** (New) button. The **Job exec. on time** dialog box appears.



3. Enter the job name and description.

Example

Job name: TEST01

Details: Hello Output

Job exec. on time

Select item

Action Exec. date Exec. time Calendar

Command execution

Command execution by list

Command execution with recovery option

Command execution in window

Send mail

View the explanation of selected item

Job exec. on time

*are mandatory fields

Job name* TEST01

Details Hello Output

Execute action

Action* Select from item list Clear

Schedule (☐ Execute immediately after finishing register)

Exec. date Select a date 15 - Select a date 15

One day only

Exec. time :

Calendar Select from item list Clear

Save job file Register Cancel

4. Under **Select item**, select the **Action** tab.

Job exec. on time

Select item

Action Exec. date Exec. time Calendar

Command execution

Command execution by list

Command execution with recovery option

Command execution in window

Send mail

Execute command.

Job exec. on time

*are mandatory fields

Job name* TEST01

Details Hello Output

Execute action

Action* Command execution Clear

Execution command* Browse

Parameter

Virtual user name

Schedule (☐ Execute immediately after finishing register)

Exec. date Select a date 15 - Select a date 15

One day only

Exec. time :

Calendar Select from item list Clear

Save job file Register Cancel

5. Select the **Command execution** item, and then click the **Apply** button.

This sets the **Action** field in the **Execute action** area to **Command execution**.

6. Enter the execution command, the parameter and the virtual user name.

Example

Execution command: msg.exe[#]

Parameter: console /w hello!!

Virtual user name: Virtual user name^{#2}

#1

If you want to execute the command on a 64-bit operating system, specify the command as follows.

When the installation path of the Windows system is C:\Windows:

C:\Windows\sysnative\msg.exe

#2

If you want to execute the command as a specific Windows user, specify the name of a virtual user associated with that Windows user.

If this is omitted, the command is executed with the account for the CPA job execution service. For details on the virtual user name, see the manual *JPI/Client Process Automation Configuration and Administration Guide*.

Execute action

Action* Command execution Clear

6. Execution command* msg.exe Browse

Parameter console /w hello!!

Virtual user name

7. Set a schedule.

In this example, select the **Execute immediately after finishing register** check box to execute the job immediately after the job is registered. (Leaving **Exec. date**, **Exec. time**, and **Calendar** in the **Schedule** section blank has the same effect as selecting the **Execute immediately after finishing register** check box.)

If you want to set a schedule to run the job at the specified date and time, see [2.5 Setting a schedule](#).

Job exec. on time

Select item

Action Exec. date Exec. time Calendar

Command execution Apply

Command execution by list

Command execution with recovery option

Command execution in window

Send mail

Job name* TEST01

Details Hello Output

Execute action

Action* Command execution Clear

Execution command* msg.exe Browse

Parameter console /w hello!!

Virtual user name

7. Schedule ☒ Execute immediately after finishing register

Exec. date Select a date 15 - Select a date 15

One day only

Exec. time :

Calendar Select from item list Clear

8. Register

9. OK

KNAO4018-I

Job [TEST01] was registered.

Save job file

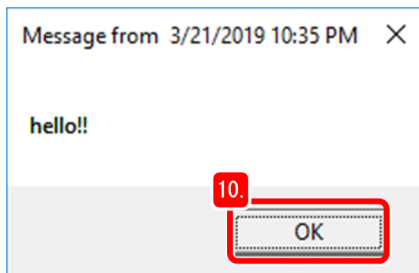
8. Click the **Register** button.

A message appears indicating the completion of the registration.

9. Click the **OK** button to run the job.

Check that the job TEST01 runs and that a message dialog box appears to show the string hello!!.

10. Click the **OK** button to close the message dialog box.

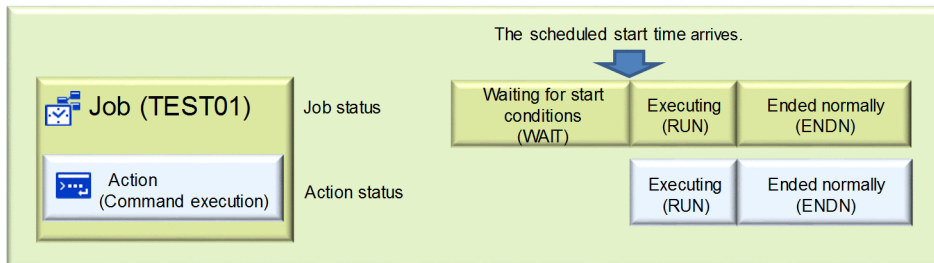


2.3 Checking job status

After you register a job and the job and its actions run, you can view the status indicating the results. If you set a schedule, the action flow and actions run in order when the specified date and time arrives. The job enters an ended state when the last action finishes.

The following figure shows the status transition of a job and its actions.

Figure 2–1: Overview of the status transition of a job

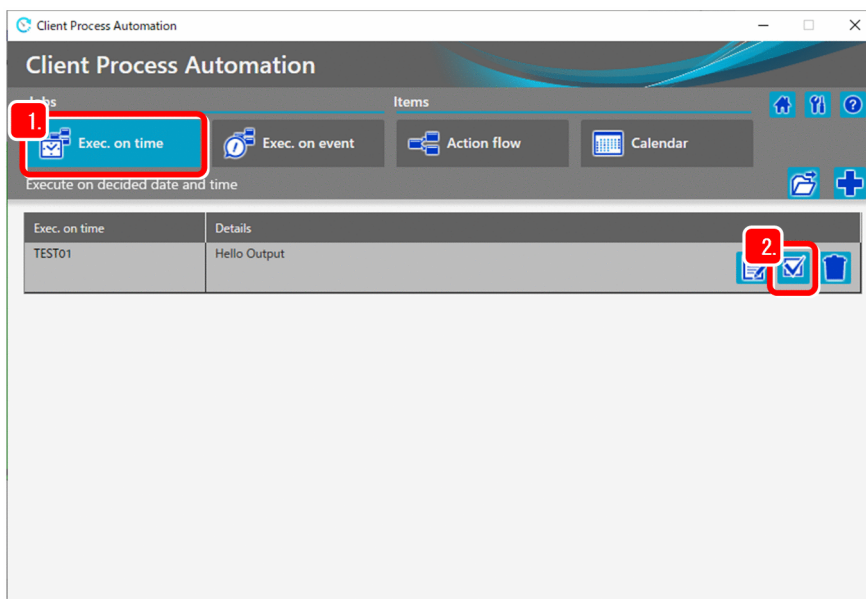


To see the execution results of a job, open the **Checking the status** dialog box from the Job Design View.

Procedure

1. Start the Job Design View and then select the **Exec. on time** tab.

When you select a job, the  (**Checking the status**) button appears.

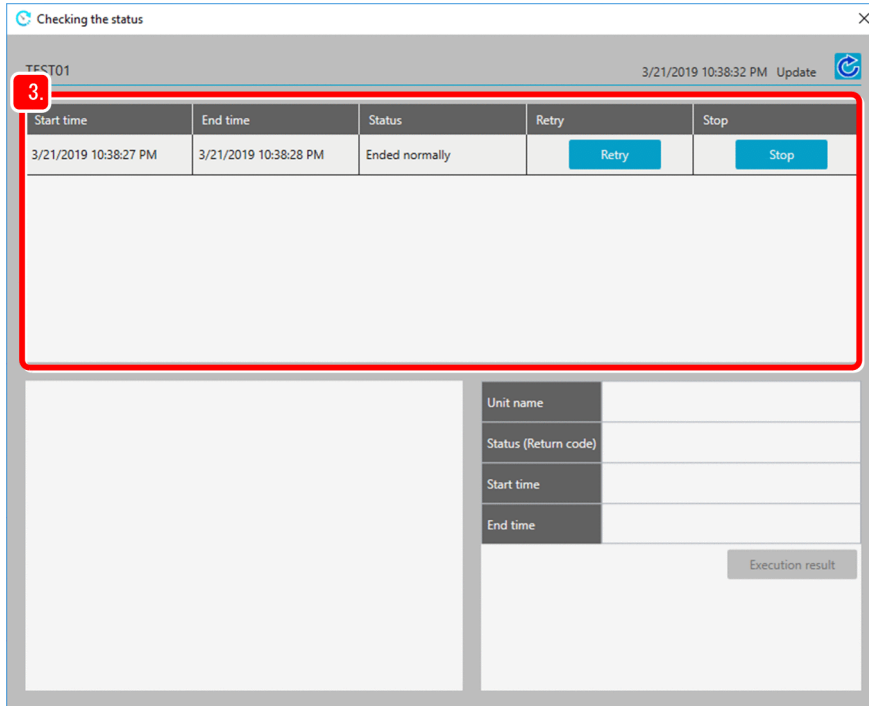


2. Click the  (**Checking the status**) button.

The **Checking the status** dialog box appears.

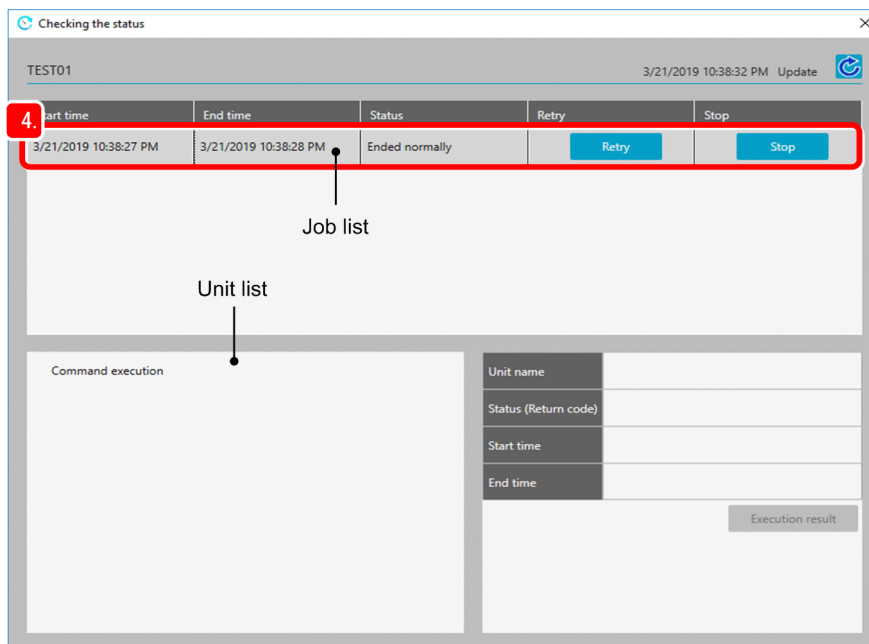
3. Check the status of the job.

The upper area of the **Checking the status** dialog box displays a job list, including the job's start time, end time, and status.



4. In the job list, select a job.

A unit list appears in the lower left area of the dialog box.



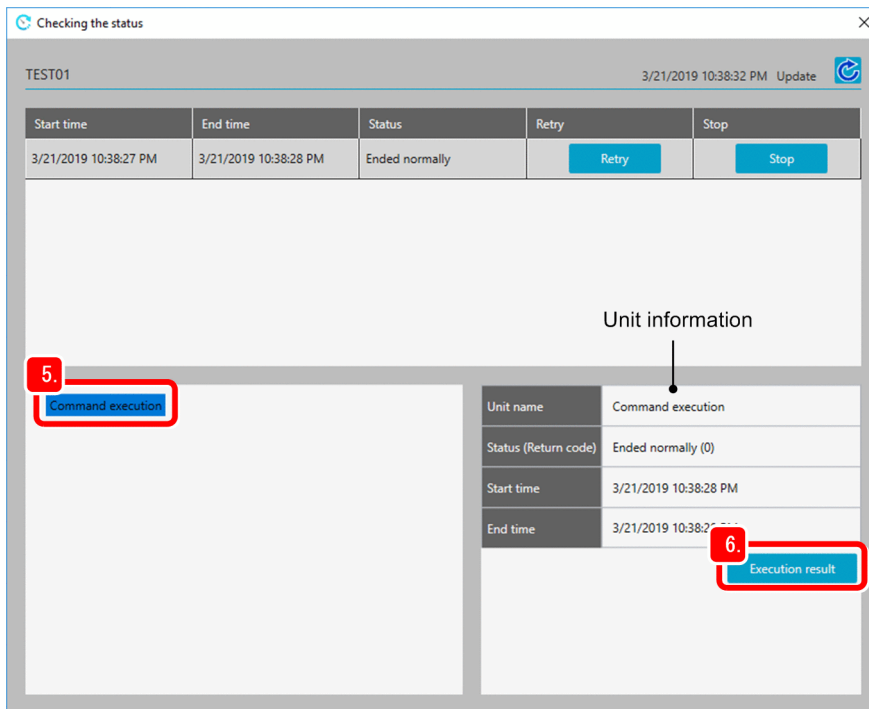
Note

In the job list and unit list shown in steps 3 and 4, a job or unit is displayed in red font if the job or unit ended abnormally. For details on how abnormally-ended jobs and units are displayed, see [2.4.1\(3\) Checking for abnormal end](#).

5. In the unit list, select a unit.

The lower right area of the dialog box displays the information of the unit, including the unit's status, the start time, and the end time.

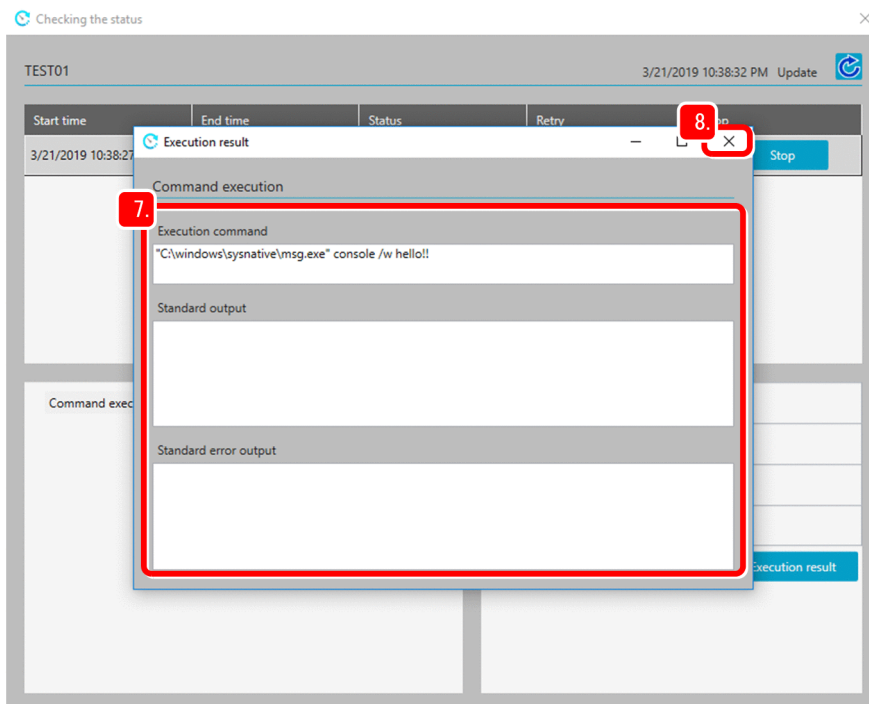
The **Execution result** button becomes available if the selected unit is an action and the command execution finished.



6. Click the **Execution result** button.

The **Execution result** dialog box appears.

7. Check the execution command, standard output, and standard error output.

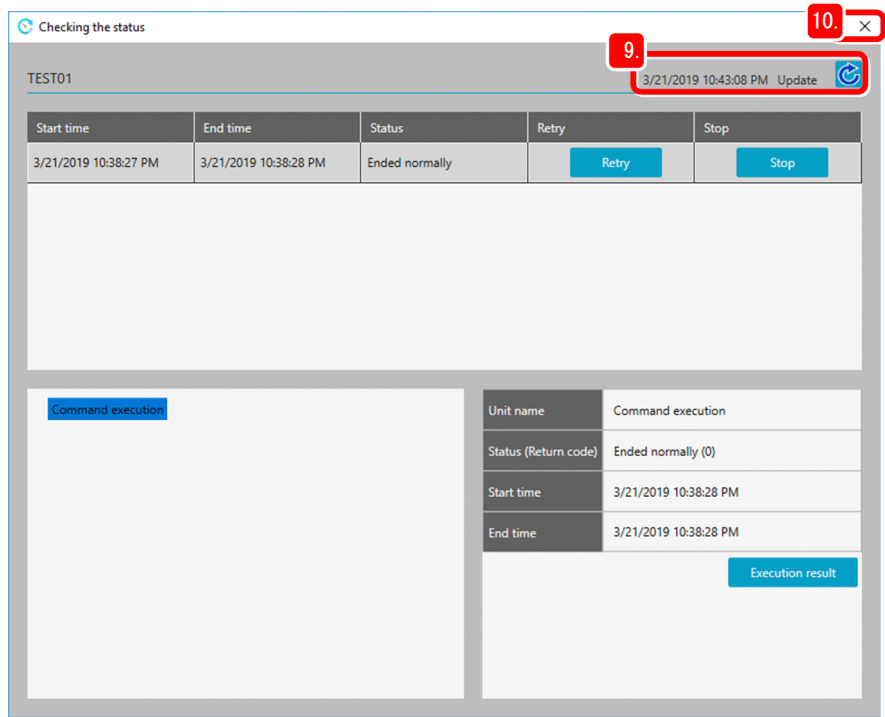


8. Click the **× (Close)** button.

After you check the displayed information, close the **Execution result** dialog box.

9. Click the **↻ (Update)** button to update the information displayed in the **Checking the status** dialog box.

This updates the information displayed in the **Checking the status** dialog box, including the update date and time.



10. Click the **×** (**Close**) button.

After you check the updated information, close the **Checking the status** dialog box.

For details on job status and action status, see the manual for Client Process Automation, which is listed in [B.1 Related publications](#).

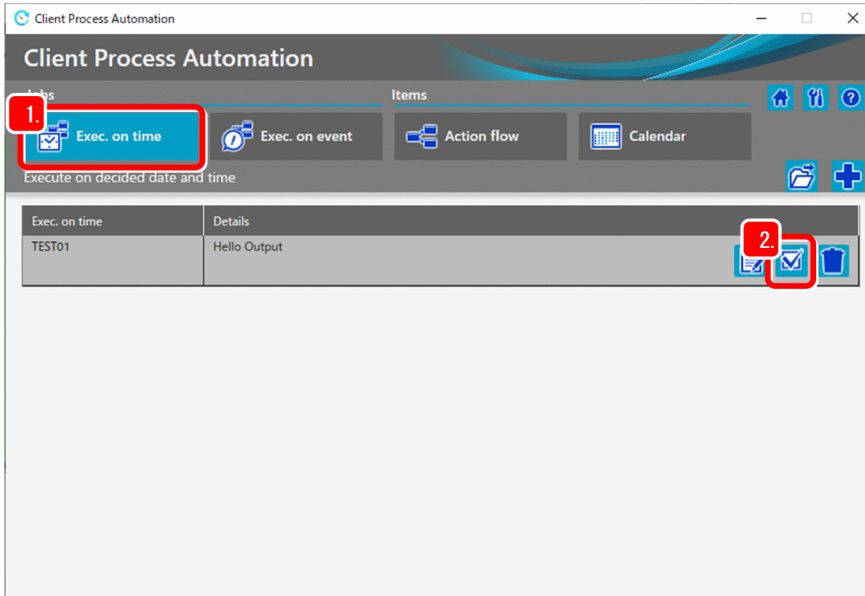
2.3.1 Rerunning a job

CPA enables you to rerun a finished job. To do this, open the **Checking the status** dialog box from the Job Design View.

Procedure

1. Start the Job Design View, and then select the **Exec. on time** tab.

When you select a job, the  (**Checking the status**) button appears.

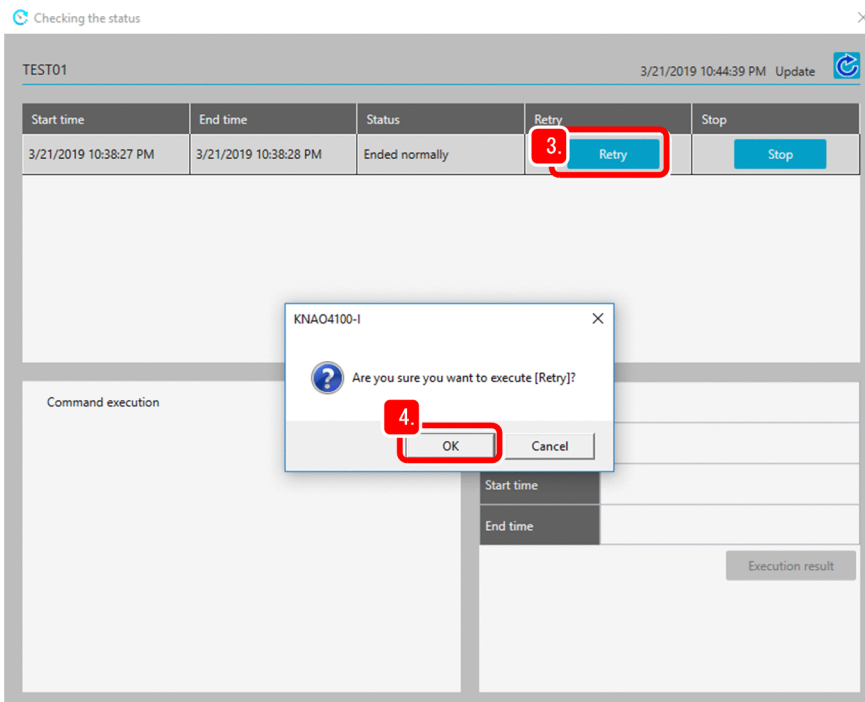


2. Click the  (**Checking the status**) button.

The **Checking the status** dialog box appears.

3. In the job list in the **Checking the status** dialog box, click the **Retry** button.

A message dialog box appears for confirmation.

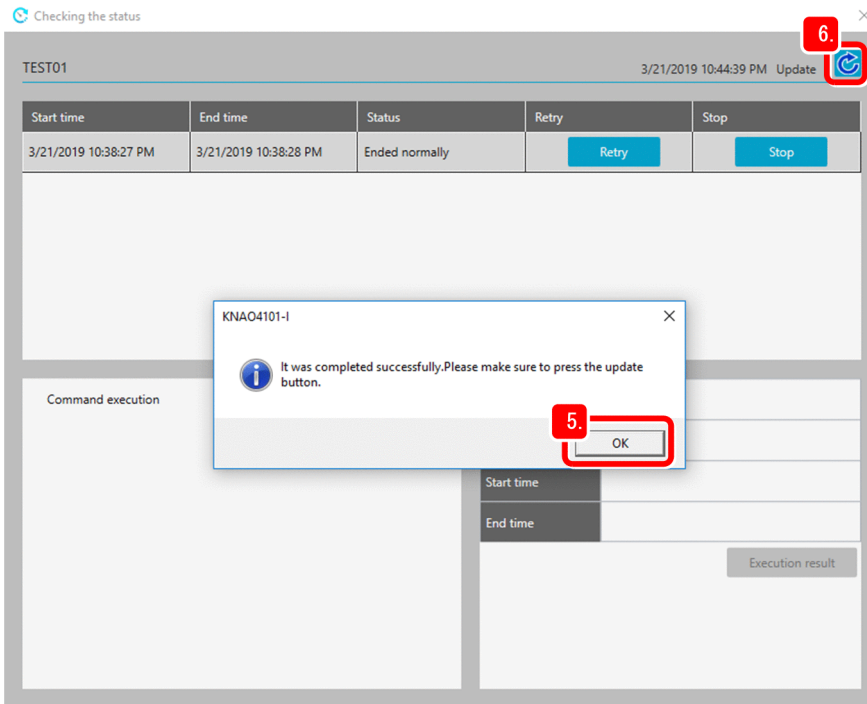


4. Click the **OK** button.

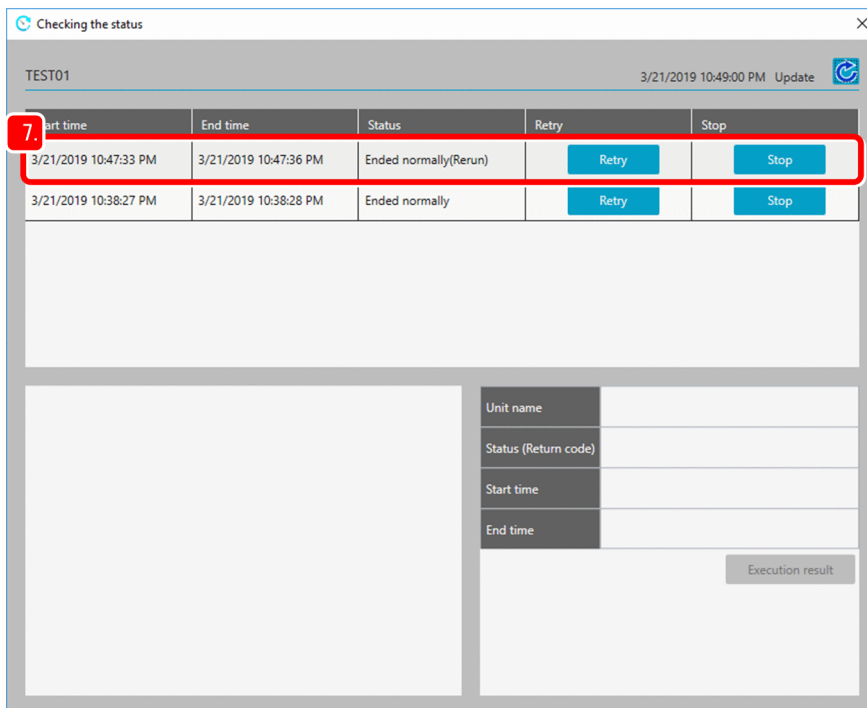
A message appears.

5. Check the information in the message dialog box, and then click the **OK** button.

Now the rerun starts.



6. Click the (**Update**) button to update the information displayed in the **Checking the status** dialog box.
7. Check the information displayed in the **Checking the status** dialog box.
A new line is added to the job list. The results of rerun jobs are indicated by the suffix (Rerun) on the state name.



For details on how to check the execution results of jobs, see [2.3 Checking job status](#).

2.3.2 Forcibly terminating a job

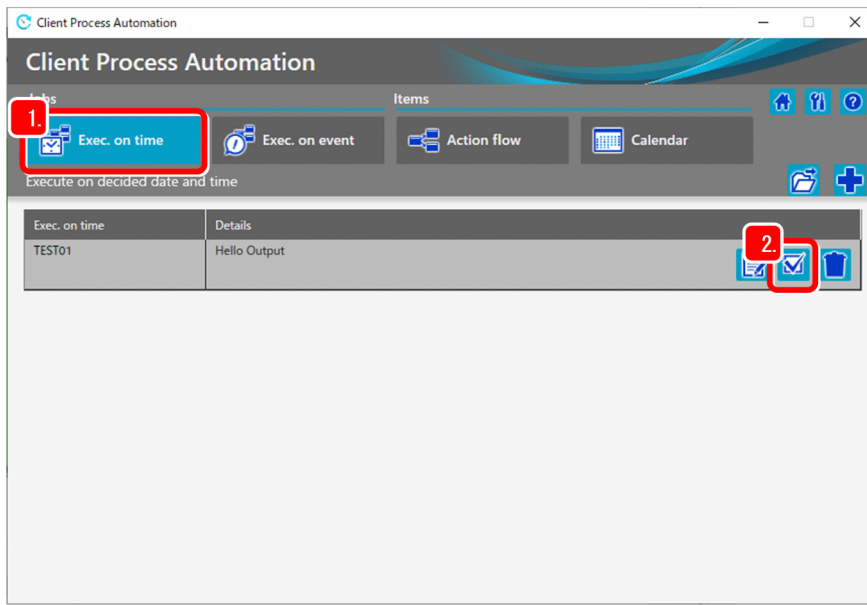
To terminate a running job forcibly, open the **Checking the status** dialog box from the Job Design View. You can terminate a job forcibly only when it is in the `Executing` state.


This procedure assumes that a job is in the `Executing` state.

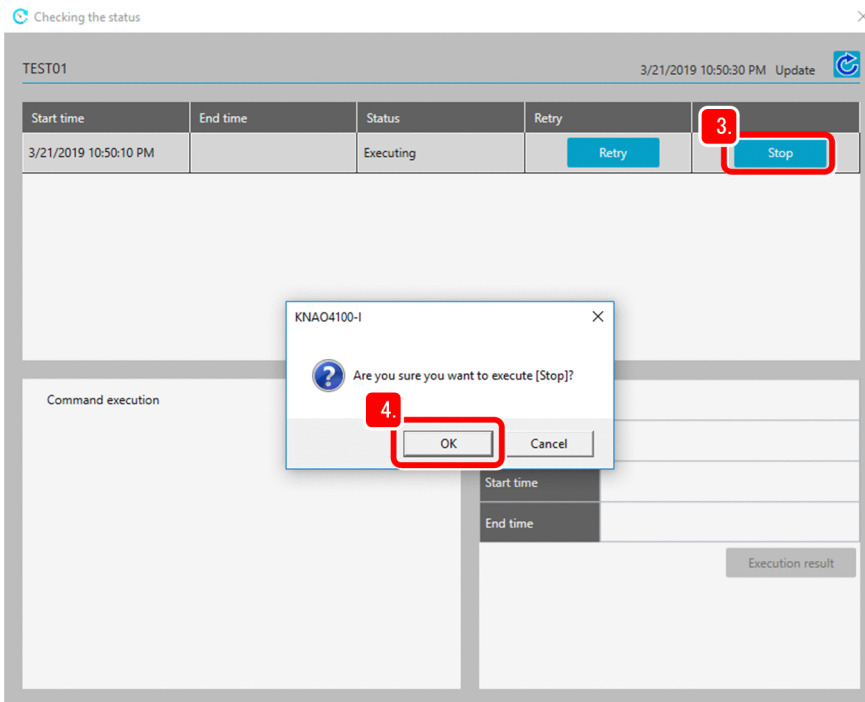
Procedure

1. Start the Job Design View, and then select the **Exec. on time** tab.

When you select a job, the  (**Checking the status**) button appears.



2. Click the  (**Checking the status**) button to open the **Checking the status** dialog box.
3. In the job list in the **Checking the status** dialog box, click the **Stop** button.
A message dialog box appears for confirmation.

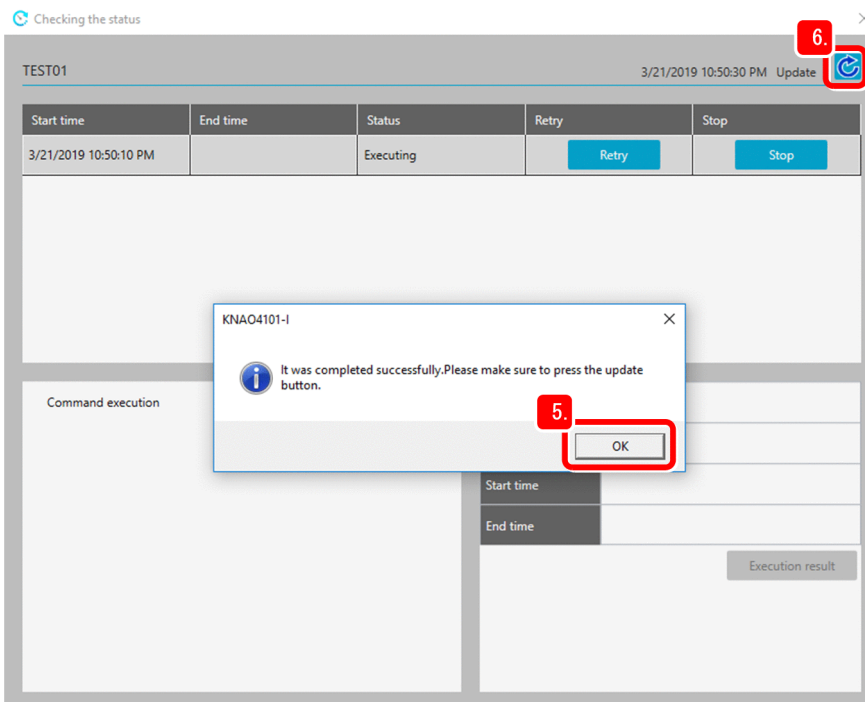


4. Click the **OK** button.

A message dialog box appears.

5. Check the information in the message dialog box, and then click the **OK** button.

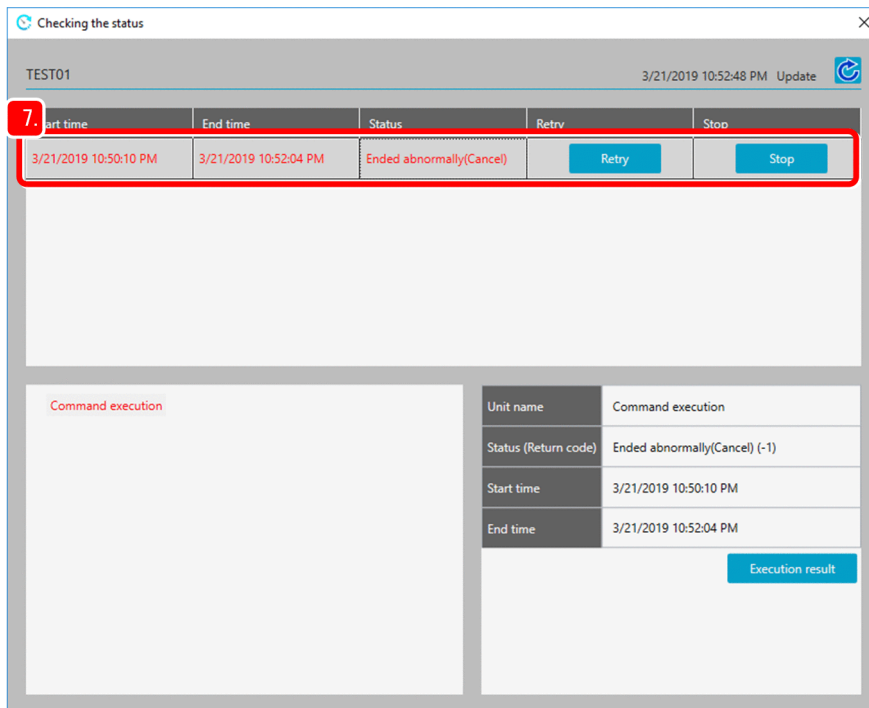
Now the job is terminated forcibly.



6. Click the **Update** button to update the information displayed in the **Checking the status** dialog box.

7. Check the information displayed in the **Checking the status** dialog box.

You can confirm that the status is set to **Ended abnormally (Cancel)**, indicating that the job was forcibly terminated.



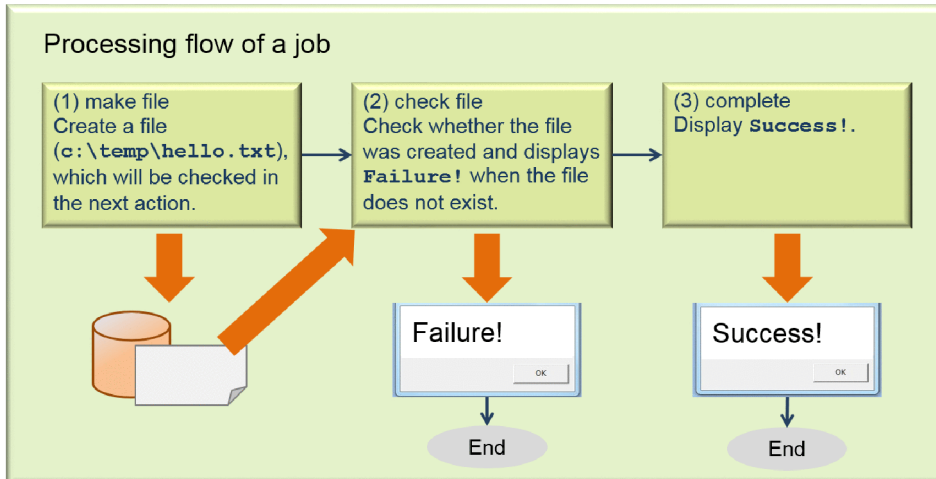
If you want to suspend a job or to stop monitoring whether an execution condition of a job executed on event is satisfied, use the `cpajobstop` command. For details on the `cpajobstop` command, see the manual for Client Process Automation, which is listed in [B.1 Related publications](#).

2.4 Creating an action flow

You can create a job that contains action flows. This section describes how to combine standard provided action items to create an action flow that executes three actions: (1) creates an executable file, (2) checks whether the file exists, and (3) displays **Success!** when the file exists or **Failure!** when the file does not exist.

The following figure shows the processing flow of a job that contains an action flow.

Figure 2–2: Processing flow of a job



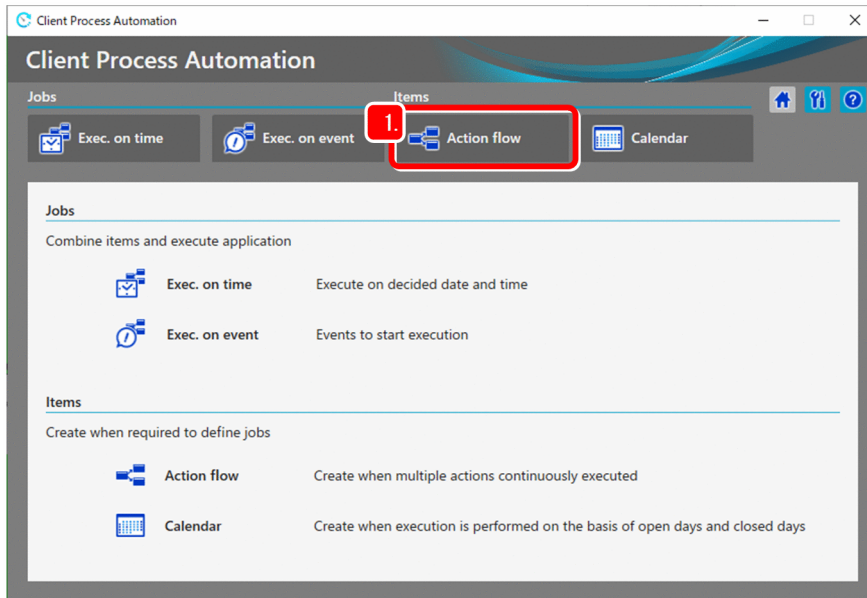
The first action creates a file named `hello.txt` by using the `echo` command (`cmd.exe /c echo "hello" >c:\temp\hello.txt`).

The second action checks the file by using the `if` command (`cmd.exe /c if not exist c:\temp\hello.txt exit 1`). This command will end with the return value of 1 when the file does not exist. Subsequently, when the file does not exist, the action displays a message dialog box by using the `msg` command (`msg.exe console /w Failure!`). Then the action ends with the return value of 1.

The third action displays a message dialog box by using the `msg` command (`msg.exe console /w Success!`).

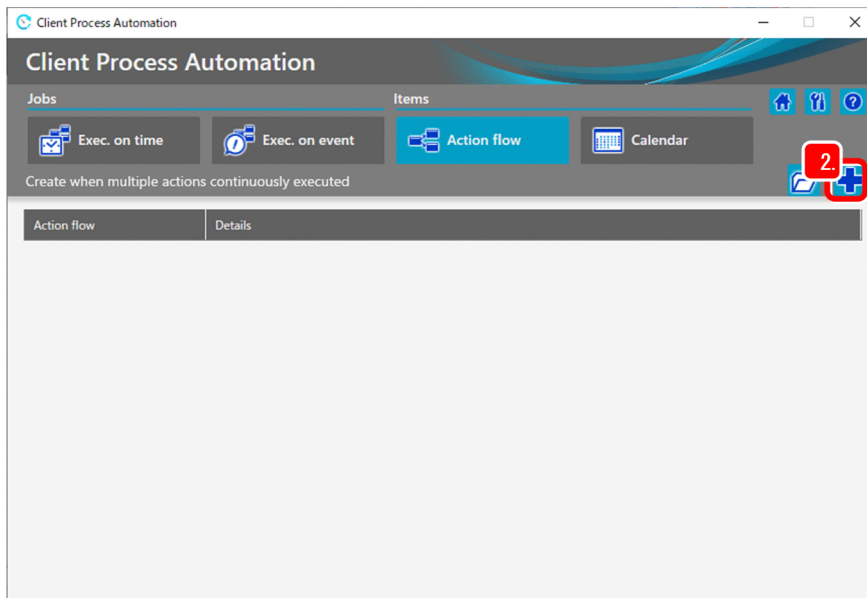
Procedure

1. Start the Job Design View, and then select the **Action flow** tab.



2. Click the **+** (New) button.

The **Action flow** dialog box appears.

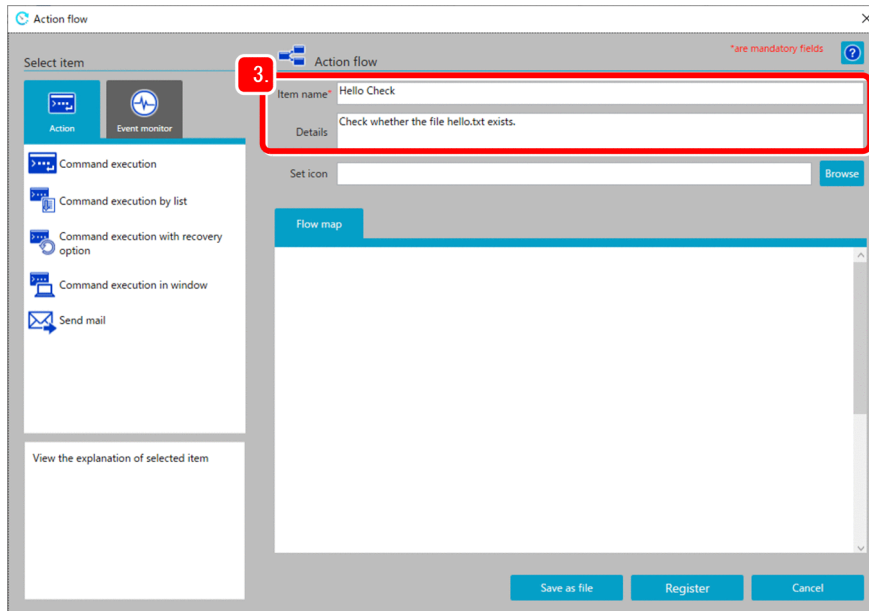


3. Enter the item name and description.

Example

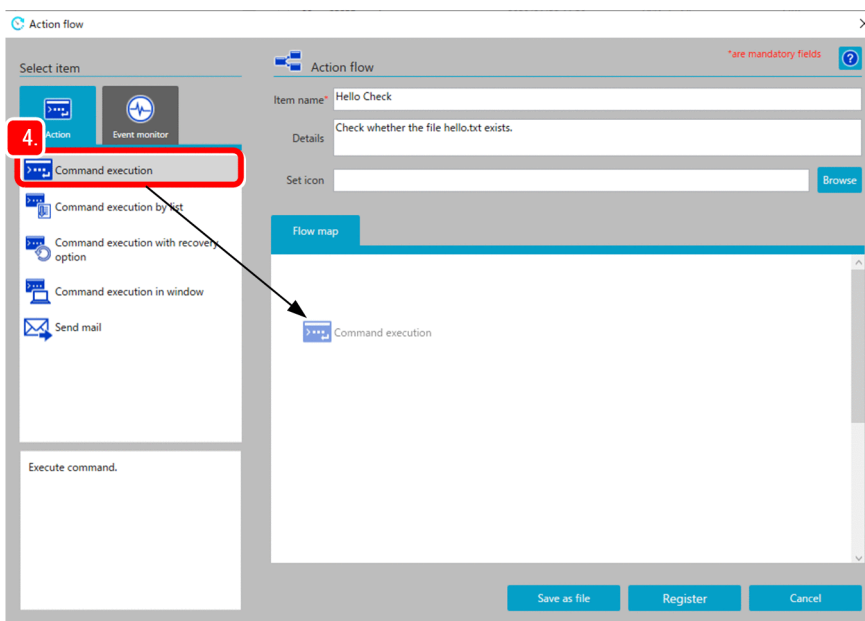
Item name: Hello Check

Details: Check whether the file hello.txt exists.



4. Create the first action to create a file.

In the **Action** tab, select the **Command execution** item, and then drag the item to the **Flow map** area. When the item is placed in the **Flow map** area, the **Unit** dialog box appears.



5. Enter the unit name, execution command, parameter and virtual user name of the action, and then click the **OK** button. Note that unit names must be unique in an action flow.

Example

Unit name: make file

Execution command: cmd.exe

Parameter: /c echo hello! >c:\temp\hello.txt

Virtual user name: *Virtual user name*[#]

#

If you want to execute the command as a specific Windows user, specify the name of a virtual user associated with that Windows user. If this is omitted, the command is executed with the account for the CPA job execution

service. For details on the virtual user name, see the manual *JPI/Client Process Automation Configuration and Administration Guide*.

6. Create the second action to check the file.

In this step, use an item that can run a recovery execution command when the execution command fails.

Drag the **Command execution with recovery option** item to the **Flow map** area in the same way as in step 4. When the item is placed in the **Flow map** area, another **Unit** dialog box appears.

7. Enter the unit name, execution command, parameter and virtual user name of the action, and then click the **OK** button.

Example

Unit name: check file

Execution command: cmd.exe

Parameter: /c if not exist c:\temp\hello.txt exit 1

Recovery execution command: cmd.exe

Parameter for recovery execution command: /c msg.exe console /w Failure! & exit 1^{#1}

Virtual user name: *Virtual user name*^{#2}

#1

If you want to run the recovery execution command in a 64-bit operating system, specify the parameter for the command as follows.

When the installation path of the Windows system is C:\Windows:

```
/c c:\Windows\sysnative\msg.exe console /w Failure! & exit 1
```

#2

If you want to execute the command as a specific Windows user, specify the name of a virtual user associated with that Windows user. If this is omitted, the command is executed with the account for the CPA job execution service. For details on the virtual user name, see the manual *JPI/Client Process Automation Configuration and Administration Guide*.

8. Create the third action to complete the file.

Drag the **Command execution** item to the **Flow map** area in the same way as in step 6. When the item is placed in the **Flow map** area, another **Unit** dialog box appears. Enter the unit name, execution command, parameter and virtual user name of the action, and then click the **OK** button.

Example

Unit name: complete

Execution command: msg.exe^{#1}

Parameter: console /w Success!

Virtual user name: Virtual user name^{#2}

#1

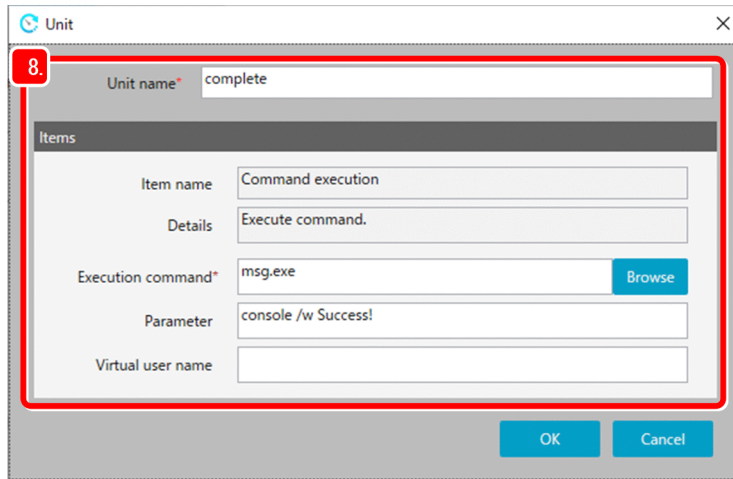
If you want to execute the command on a 64-bit operating system, specify the command as follows.

When the installation path of the Windows system is C:\Windows:

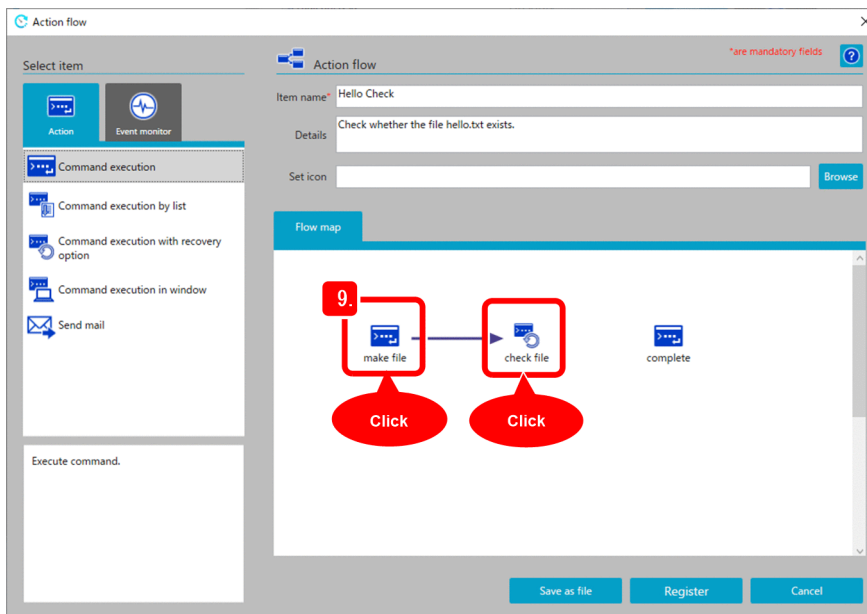
```
C:\Windows\sysnative\msg.exe
```

#2

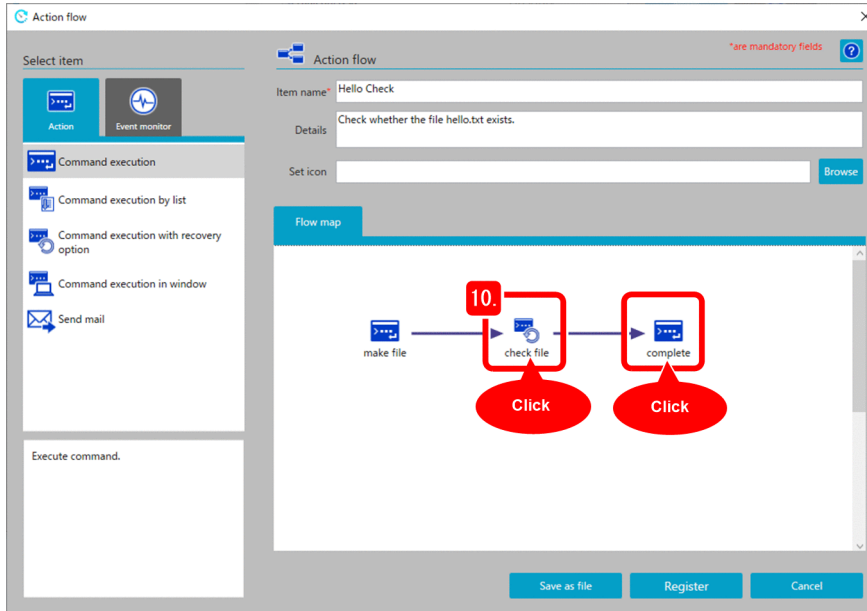
If you want to execute the command as a specific Windows user, specify the name of a virtual user associated with that Windows user. If this is omitted, the command is executed with the account for the CPA job execution service. For details on the virtual user name, see the manual *JPI/Client Process Automation Configuration and Administration Guide*.



9. Draw a relation line between two units to define the order in which the actions will run.
Click the first unit `make file`, and then click the second unit `check file`.



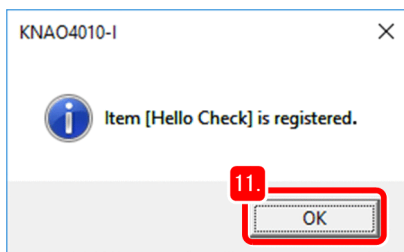
10. Draw a relation line between the second unit `check file` and the third unit `complete`.
Draw a relation line in the same way as in step 9, and then click the **Register** button. A message appears, indicating that registration is completed.



11. Check the information in the message dialog box, and then click the **OK** button.

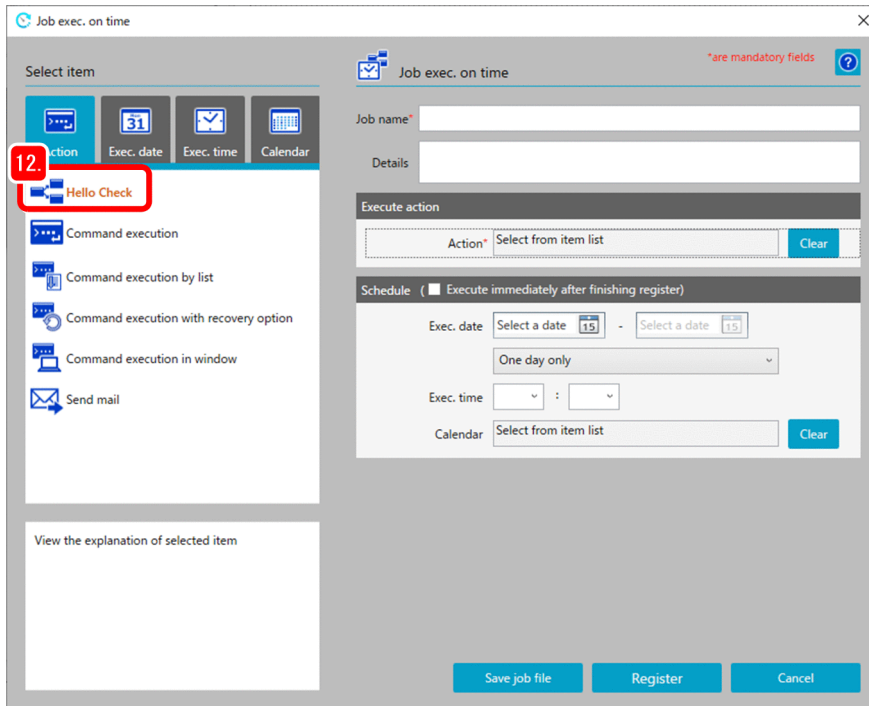
Now the action flow is created. Next, let's register a job.

For details on how to register a job, see [2.2 Creating a job](#).



12. Return to the Job Design View, and open the **Job exe. on time** dialog box.

Under **Select item**, you find that the **Action** tab contains the **Hello Check** item you created.

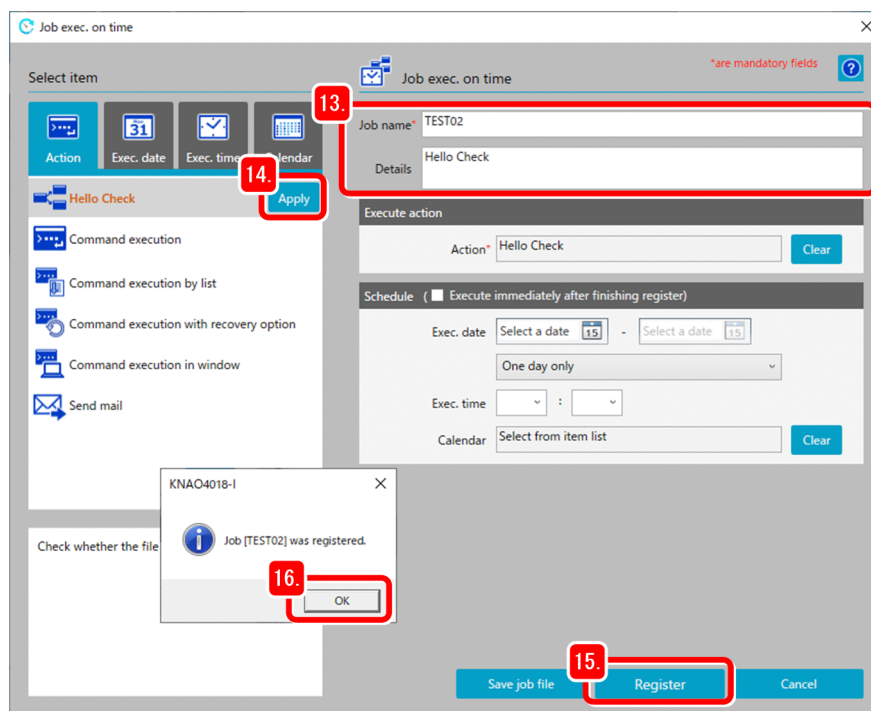


13. Enter the job name and description.

Example

Job name: TEST02

Details: Hello Check



14. Select the item, and then click the **Apply** button. This sets **Execute action** to the item.

15. Click the **Register** button.

A message appears indicating the completion of the registration.

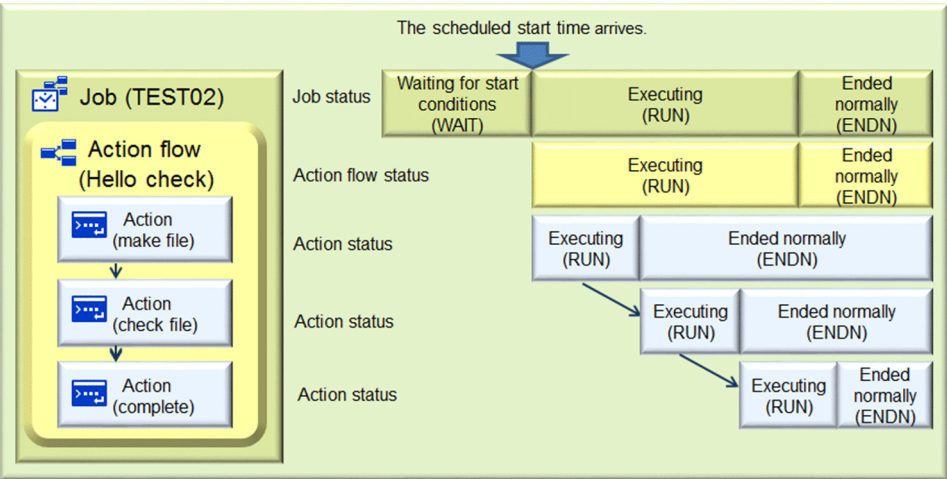
16. Click the **OK** button.
- Now the job is registered.

2.4.1 Checking action status

To see the execution results of actions, open the **Checking the status** dialog box from the Job Design View.

A status is assigned to each of the action flow (Hello Check) contained in the job and the actions (make file, check file, and complete) defined in the action flow.

Figure 2–3: Overview of the status transition of a job

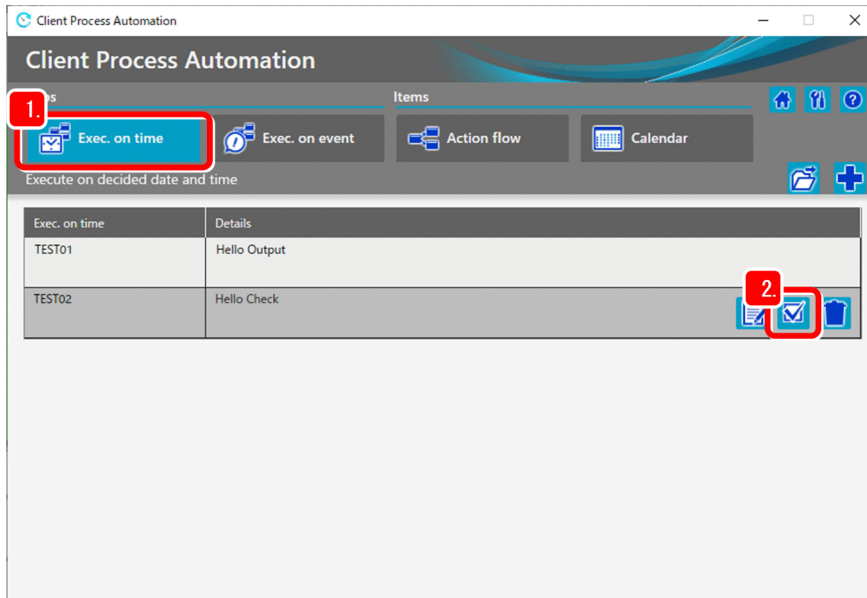


(1) Checking actions

In this example, the action `make file` created the `c:\temp\hello.txt` file and thus the action `check file` ended normally. The normal end caused the third action to display a message dialog box containing the message `Success!`.

Procedure

1. In the Job Design View, click the **Exec. on time** tab.
- When you select a job, the ☒ (**Checking the status**) button appears.

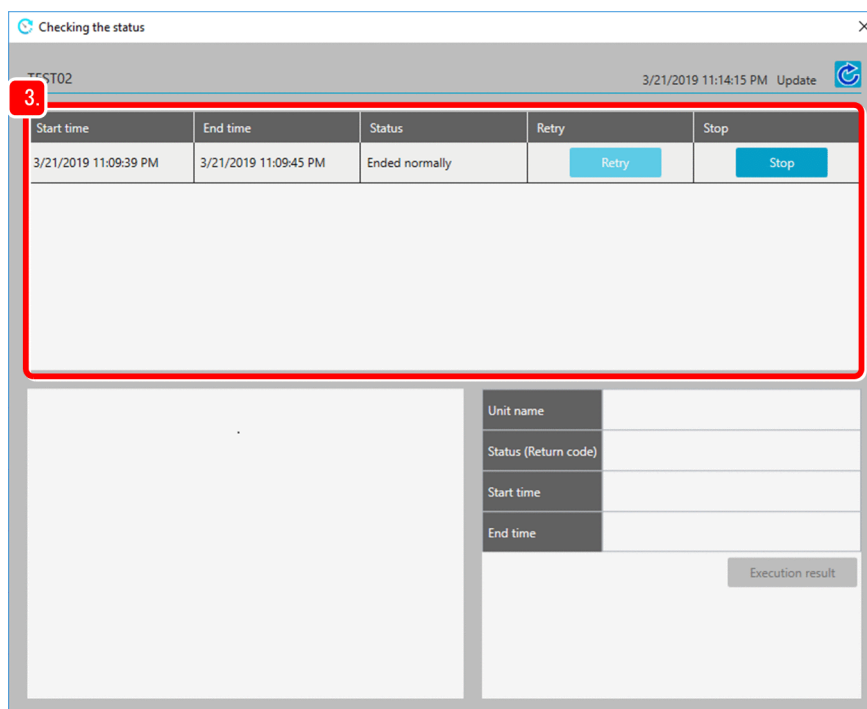


2. Click the  (**Checking the status**) button.

The **Checking the status** dialog box appears.

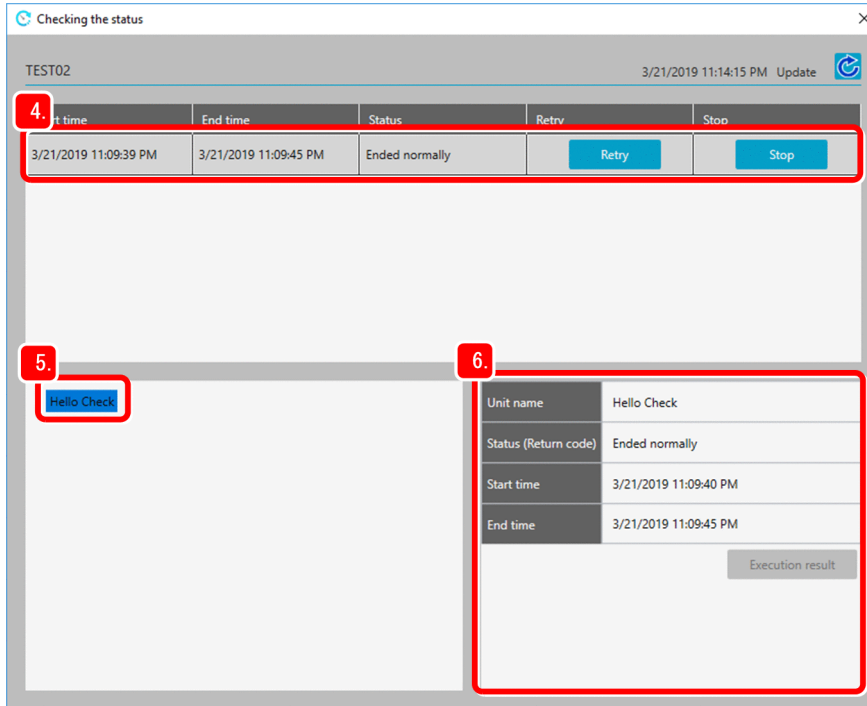
3. Check the status of the job.

The upper area of the **Checking the status** dialog box displays a job list, including the job's start time, end time, and status.



4. In the job list, select a job.

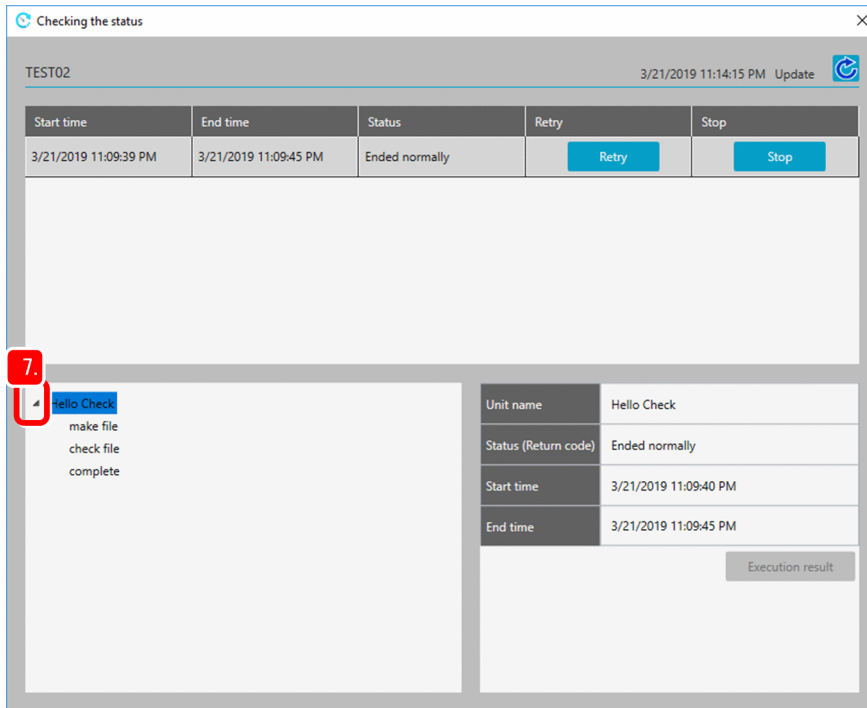
A unit list appears in the lower left area of the dialog box.



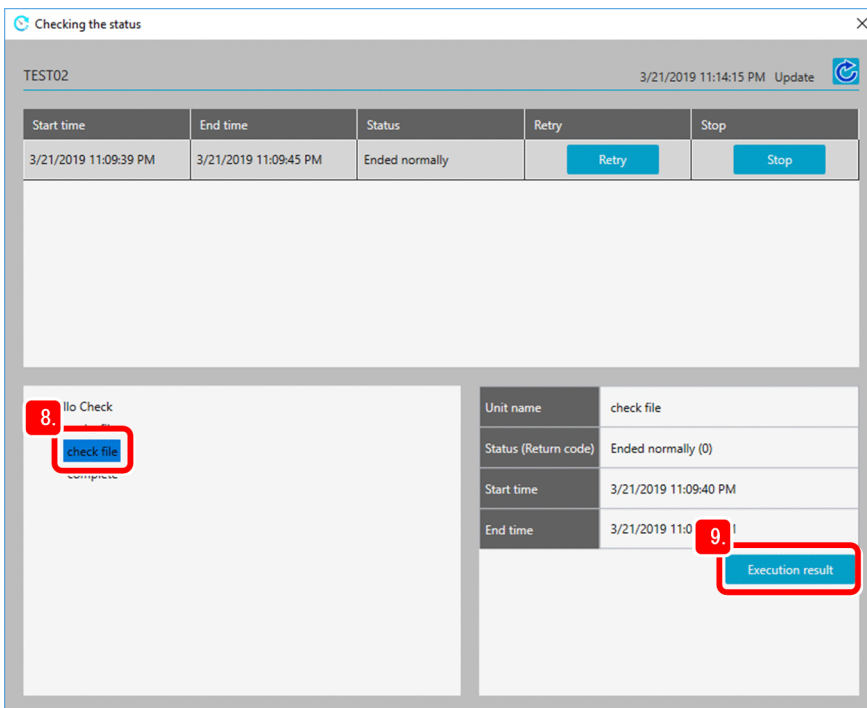
Note

In the job list and unit list shown in steps 3 and 4, a job or unit is displayed in red font if the job or unit ended abnormally. For details on how abnormally-ended jobs and units are displayed, see [2.4.1\(3\) Checking for abnormal end](#).

5. In the unit list, select a unit.
6. The lower right area of the dialog box displays the information of the unit, including the unit's status, the start time, and the end time.
7. In the unit list, click the triangle mark to the left of the unit name.
Units in the action flow appear in a tree view.



8. Select the `check file` unit.

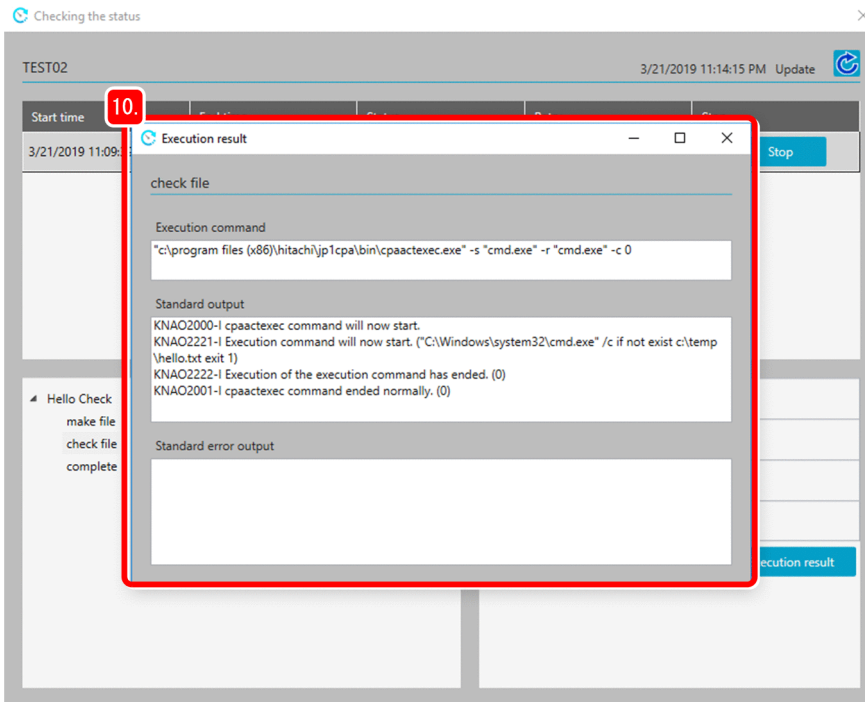


9. Click the **Execution result** button in the lower right of the dialog box.

The **Execution result** dialog box appears.

10. Check the execution command, standard output, and standard error output.

You find that the `check file` action ended normally.



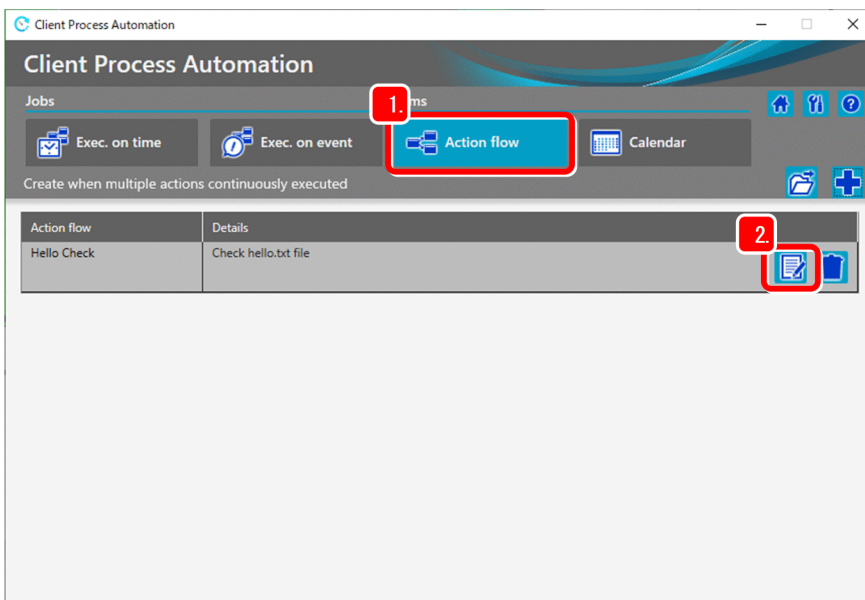
(2) Testing the recovery execution command


Let's test the recovery execution command in the `check file` action.

Firstly, delete the `hello.txt` file that you initially created.

Procedure

1. In the Job Design View, click the **Action flow** tab.

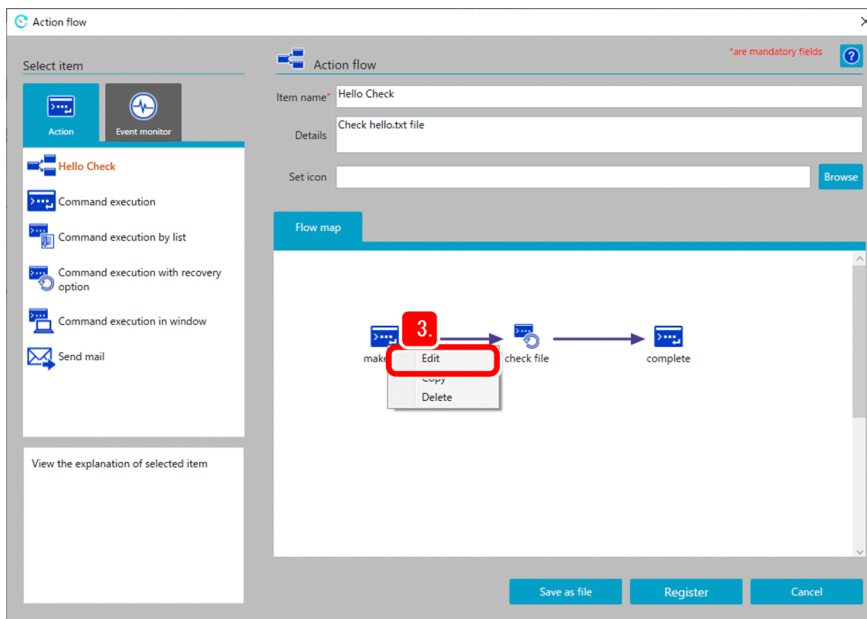


2. Select the action flow `Hello Check`, and then click the  (**Edit**) button.

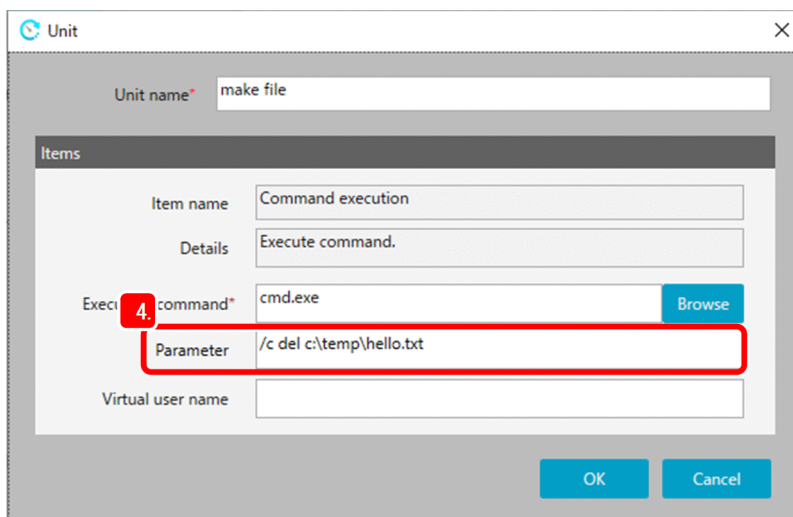
A dialog box appears for the action flow `Hello Check`.

3. Right-click the icon of the unit `make file` and select **Edit** from the menu. Alternatively, double-click the icon of the unit `make file`.

A dialog box appears for the unit `make file`.

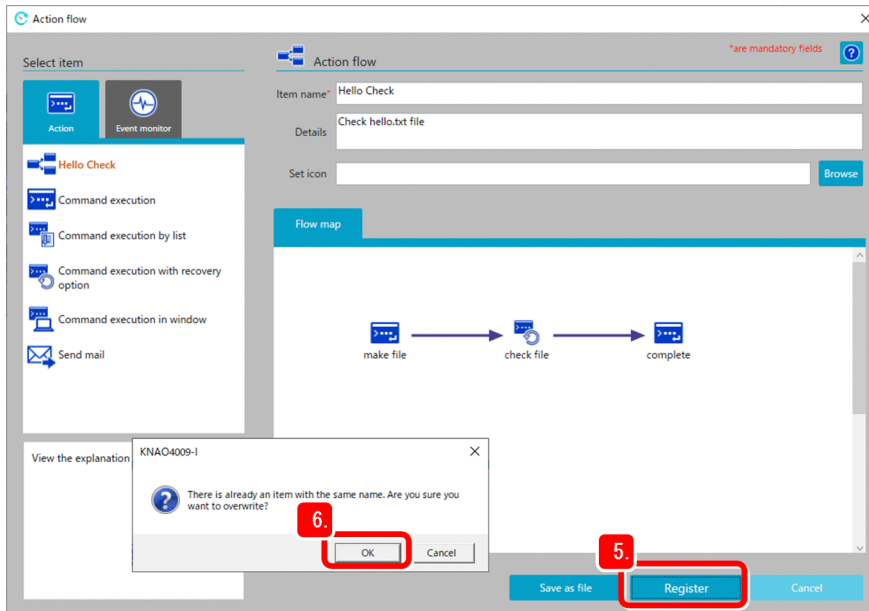


4. Change the parameter from `/c echo hello!! >c:\temp\hello.txt` to `/c del c:\temp\hello.txt`, and then click the **OK** button.



5. Click the **Register** button to register the action flow `Hello Check`.

A message appears, asking whether you really want to overwrite the action flow.

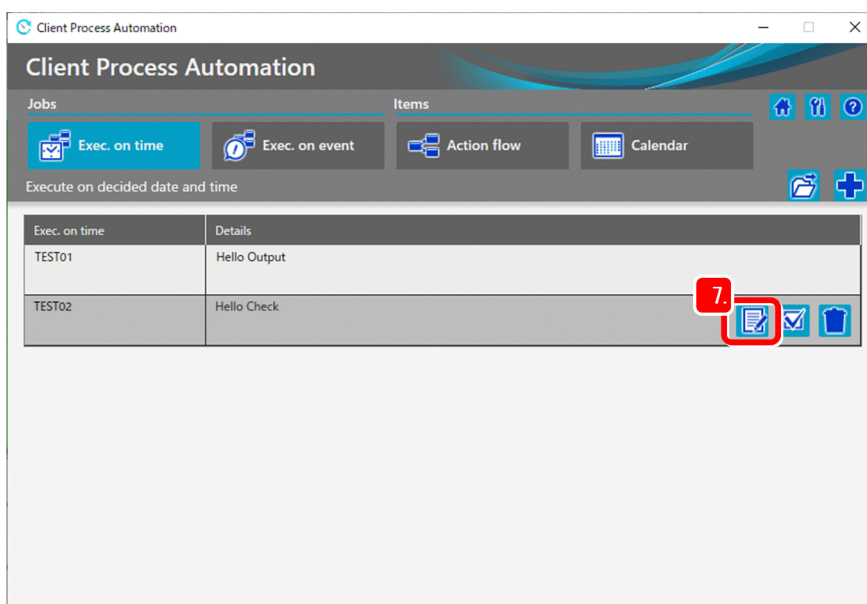


6. Click the **OK** button.

The Job Design View appears again. In the **Exec. on time** tab, select the job TEST02.

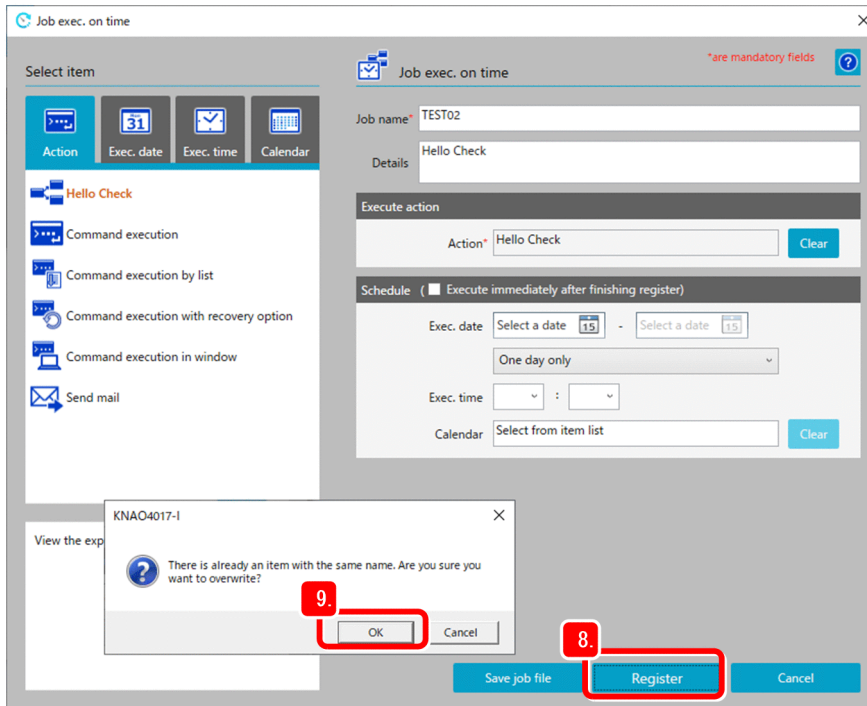
7. Click the  (**Edit**) button.

A dialog box appears for the job TEST02.



8. Click the **Register** button to register the job.

A message appears, asking whether you really want to overwrite the job.

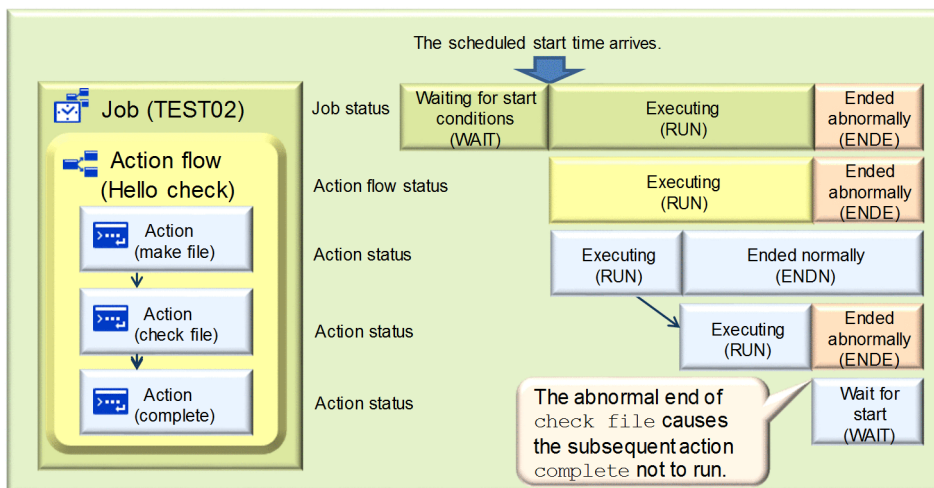


9. Click the **OK** button.

(3) Checking for abnormal end

The following figure shows the status transition of the action flow (Hello Check) contained in the job and the actions (make file, check file, and complete) contained in the action flow. The second action (check file) ended abnormally (ENDE). This caused the subsequent action (complete) not to run and the status of both the action flow and the job, which are containers of the second action, to be set to Ended abnormally (ENDE).

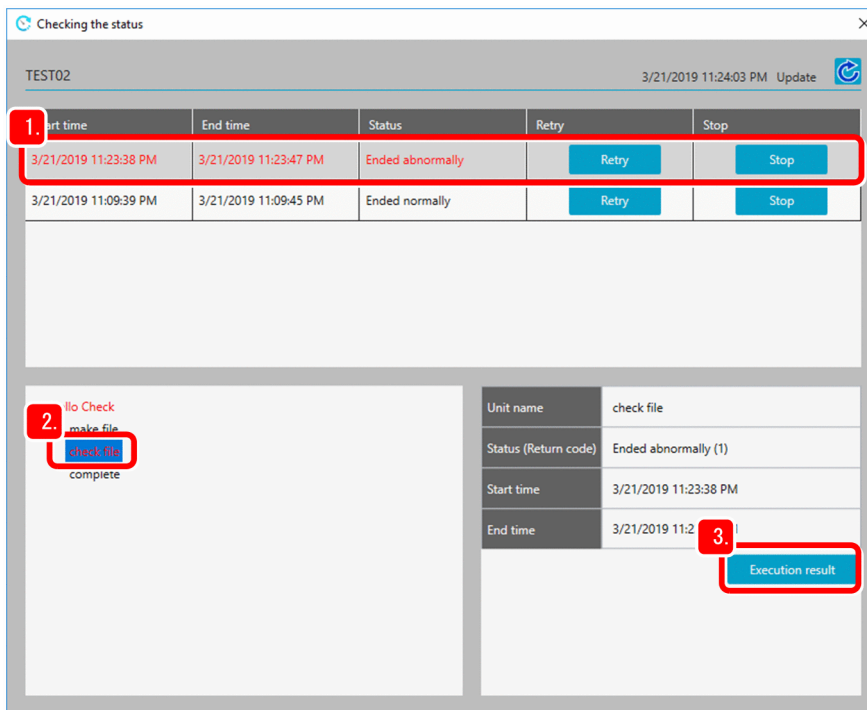
Figure 2–4: Status transition in a job with an abnormally-ended action



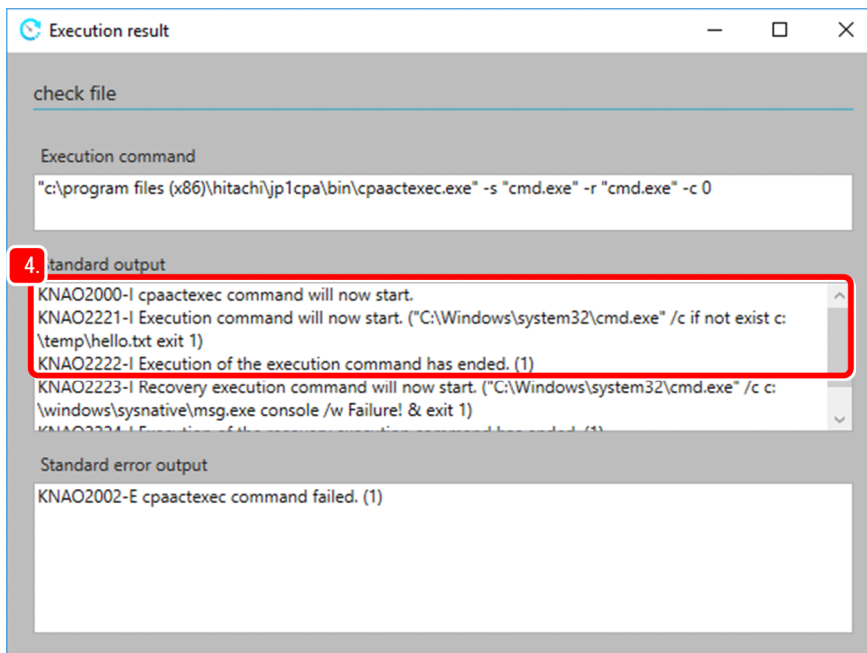
In this example, the action `make file` deleted the `c:\temp\hello.txt` file and thus the `if` statement in the action `check file` returned true. This caused the execution command (`cmd.exe`) ended abnormally. The abnormal end, in turn, caused the recovery execution command to run to display a dialog box containing the message `Failure!`.

Procedure

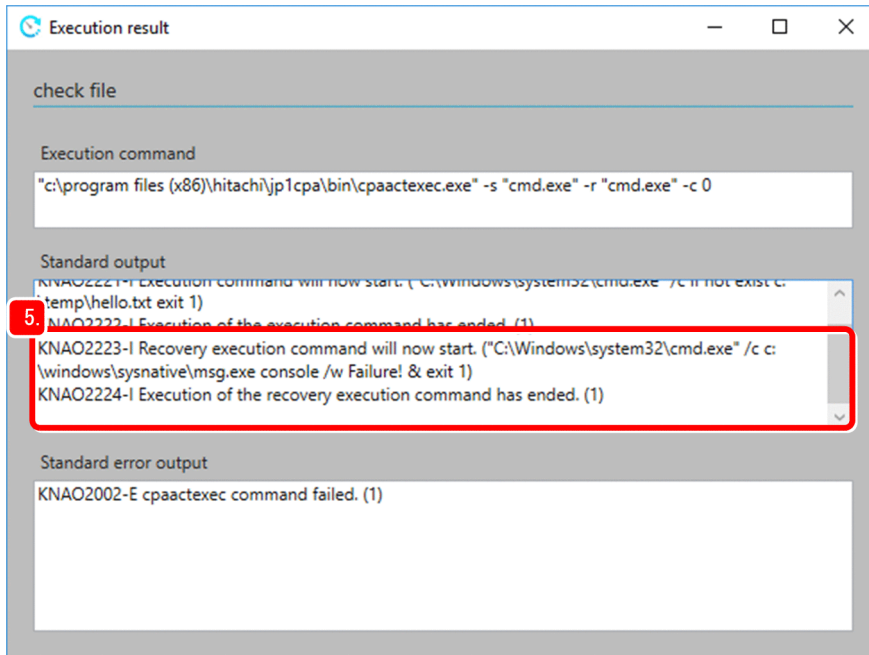
1. Open the **Checking the status** dialog box. From the job list, select an abnormally-ended job, which is displayed in red font.



2. Select the unit `check file`, which is displayed in red font.
3. Click the **Execution result** button.
The **Execution result** dialog box appears.
4. Check the execution command, standard output, and standard error output.
You find that the execution command in `check file` ended abnormally.



5. Additionally, confirm that the recovery execution command run has ended.



2.5 Setting a schedule

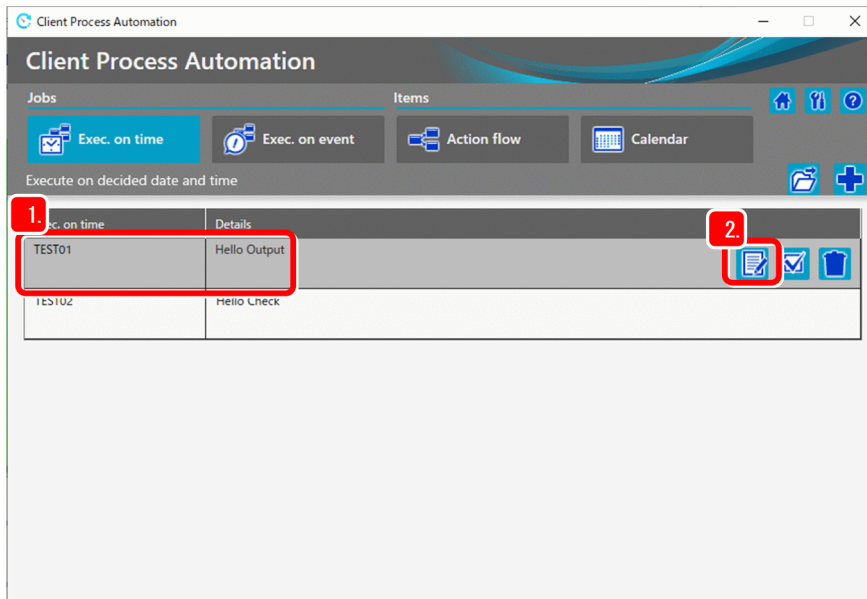
The previous example in [2.2 Creating a job](#) creates a job and runs it immediately. In addition to it, you can set a schedule to run a job at a desired time.

This example configures a job to run three minutes after the current time. You close the Job Design View after registering a job and wait until the job runs.

Procedure

1. In the Job Design View, click the **Exec. on time** tab.

Select the job TEST01.

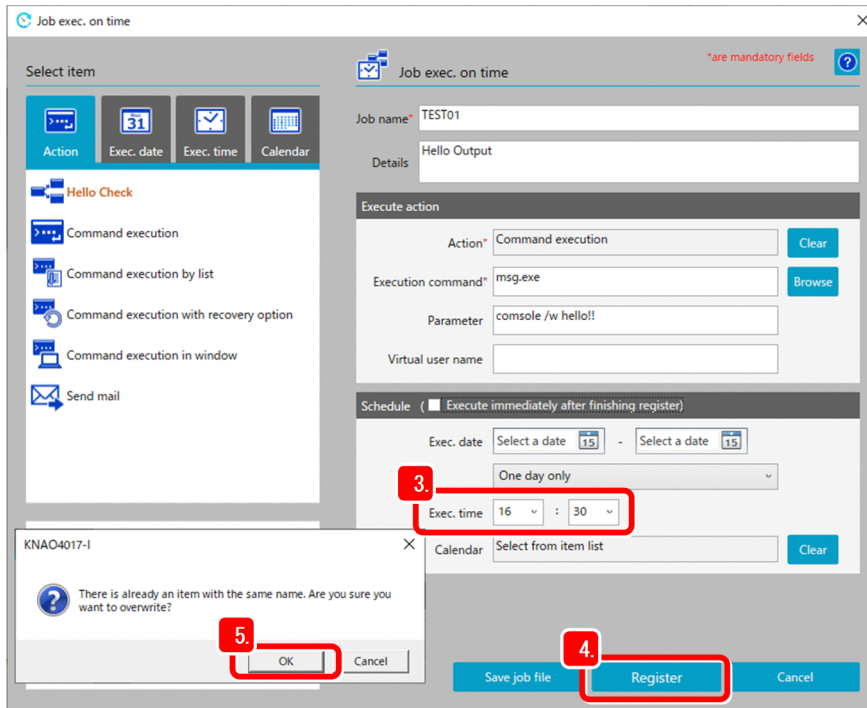


2. Click the  (**Edit**) button.

The **Job exec. on time** dialog box appears.

3. Specify the job execution time.

Specify the execution time by adding three minutes to the current time. For example, specify 16:30 if the current time is 16:27.



4. Click the **Register** button.

A message appears, asking whether you really want to overwrite the job.

5. Click the **OK** button.

The Job Design View appears again. Wait until the time you specified in step 3.

Check that the job displays a message dialog box containing the message `hello!` at the specified time (16:30 in this example).

2.6 Registering a job executed on event

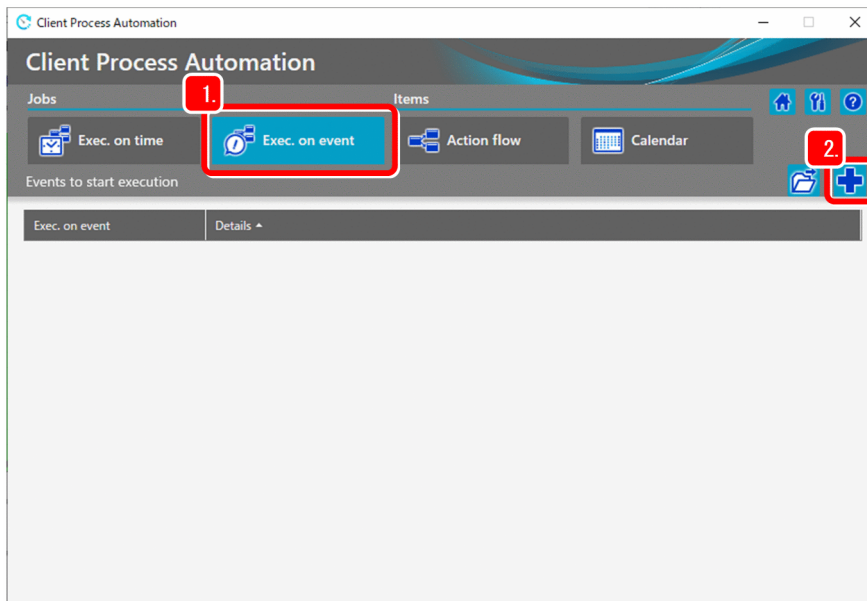
The example in [2.2 Creating a job](#) runs a job immediately and the example in [2.5 Setting a schedule](#) runs a job at a specified time. In addition to them, you can configure a job to run when a specified event occurs.

This example configures a job to monitor a file event and then copy the file when the file is created.

Specifically, when the text file `c:\temp\test.txt` is created, the job copies it to `c:\temp\backup.txt`.

Procedure

1. In the Job Design View, click the **Exec. on event** tab.

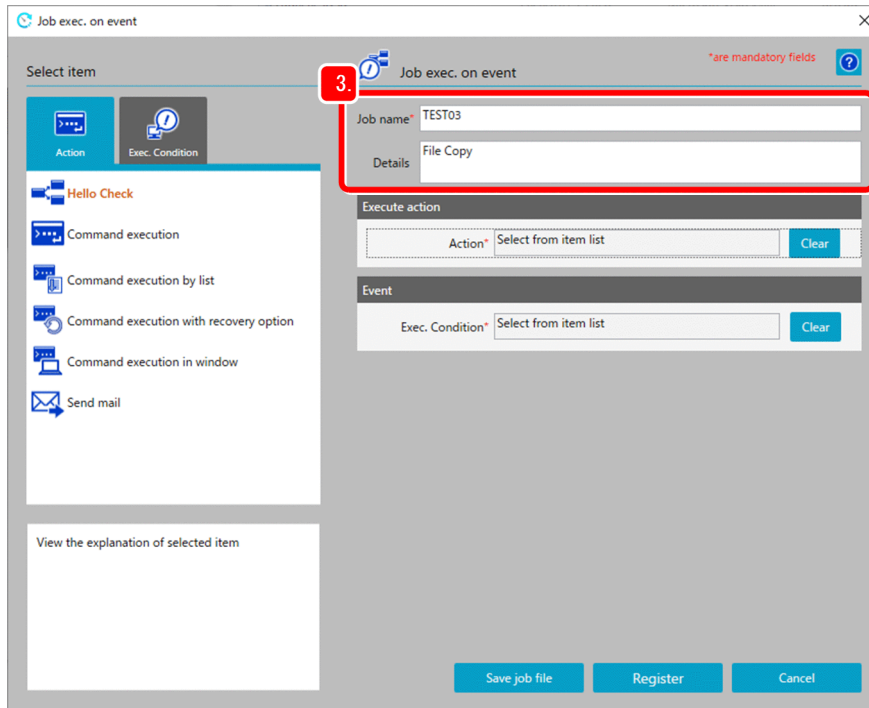


2. Click the **+** (New) button.
The **Job exec. on event** dialog box appears.
3. Enter the job name and description.

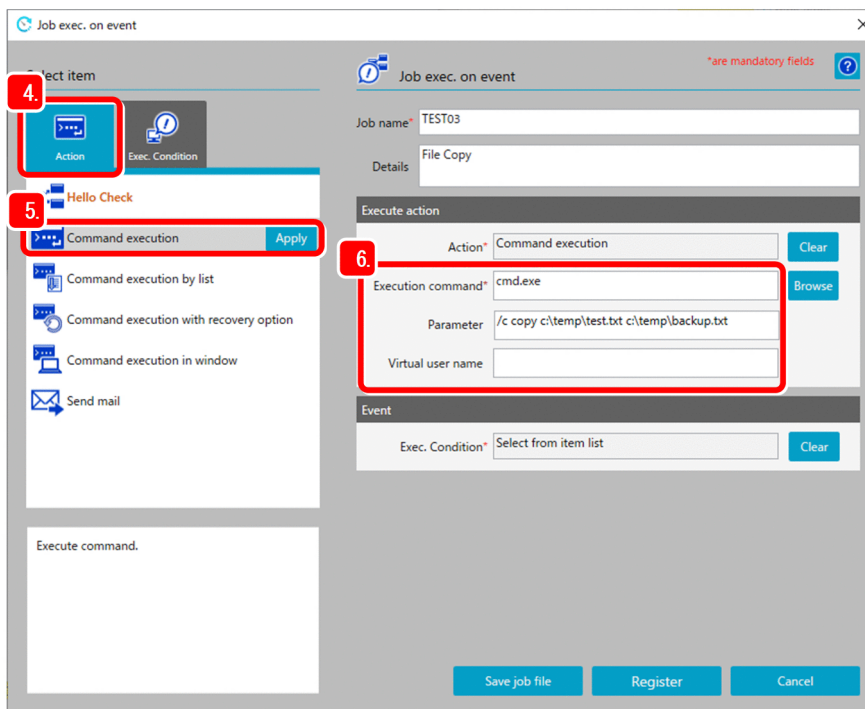
Example

Job name: TEST01

Details: File Copy



4. Under **Select item**, select the **Action** tab.



5. Select the **Command execution** item, and then click the **Apply** button.

6. Enter the execution command, the parameter and the virtual user name.

Example

Execution command: `cmd.exe`

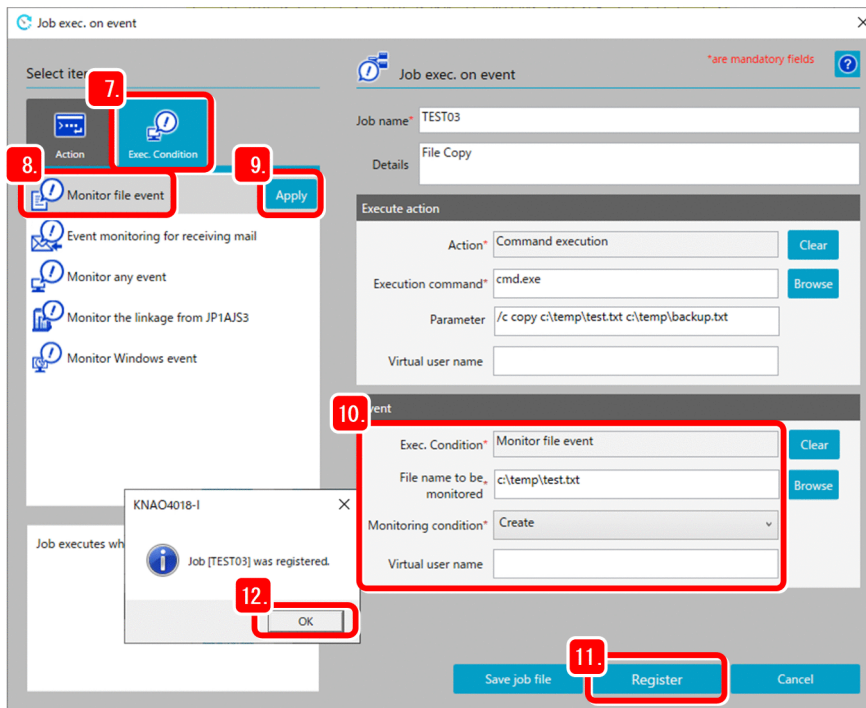
Parameter: `/c copy c:\temp\test.txt c:\temp\backup.txt`

Virtual user name: *Virtual user name*[#]

#

If you want to execute the command as a specific Windows user, specify the name of a virtual user associated with that Windows user. If this is omitted, the command is executed with the account for the CPA job execution service. For details on the virtual user name, see the manual *JPI/Client Process Automation Configuration and Administration Guide*.

7. Under **Select item**, select the **Exec. Condition** tab.



8. Select the **Monitor file event** item.

9. Click the **Apply** button.

10. Enter the name of the file you want to monitor, the monitoring condition and virtual user name.

Example

File name to be monitored: c : \temp\test.txt

Monitoring condition: Create

Virtual user name: *Virtual user name*[#]

#

If you want to monitor the file as a specific Windows user, specify the name of a virtual user associated with that Windows user. If this is omitted, the command is executed with the account for the CPA job execution service. For details on the virtual user name, see the manual *JPI/Client Process Automation Configuration and Administration Guide*.

11. Click the **Register** button.

A message appears indicating the completion of the registration.

12. Click the **OK** button.

Now the job executed on event is registered.

Create the test file c : \temp\test.txt, and then check whether the file is copied.

Appendixes

A. Version Changes

This appendix describes the changes made in the manuals for each version.

A.1 Changes in version 13-00

- Windows 11 was added as an applicable operating system.
- The following operating systems are no longer supported:
 - Windows 7
 - Windows 8
 - Windows 8.1

A.2 Changes in version 12-50

- Added an item for specifying a virtual user name as an execution user name in the item definition.
- Changed how the product name is expressed in this manual.

A.3 Changes in version 12-10

- Design modifications were made to each icon in the windows.

A.4 Changes in version 12-01

There are no modifications.

B. Reference Material for This Manual

This appendix provides reference information, including various conventions, for this manual.

B.1 Related publications

This manual is part of a related set of manuals. The manuals in the set are listed below (with the manual numbers):

- *JP1 Version 13 JP1/Client Process Automation Configuration and Administration Guide* (3021-3-L56(E))
- *JP1 Version 13 JP1/Client Process Automation Messages* (3021-3-L57(E))

B.2 Conventions: Abbreviations for product names

This manual uses the following abbreviations for product names:

Abbreviation	Full name or meaning
CPA	JP1/Client Process Automation

B.3 Conventions: Acronyms

This manual also uses the following acronyms:

Acronym	Full name or meaning
GUI	Graphical User Interface
NTFS	New Technology File System
OS	Operating System

B.4 Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is 1,024² bytes.
- 1 GB (gigabyte) is 1,024³ bytes.
- 1 TB (terabyte) is 1,024⁴ bytes.

Index

A

abbreviations for products 51
acronyms 51

B

basic functions for automation 12

C

checking action status 34
checking job status 17
conventions
 abbreviations for products 51
 acronyms 51
 fonts and symbols 5
 KB, MB, GB, and TB 51
 version numbers 6
CPA
 installing a new instance 10
 requirements 9
creating a job 13
creating an action flow 26

F

font conventions 5
forcibly terminating a job 23

G

GB meaning 51

I

installing a new instance
 CPA 10

K

KB meaning 51

M

MB meaning 51
memory and disk space required for installation 9

P

preparation before installation 9

R

Reference Material for This Manual 51
registering a job executed on event 46
Related publications 51
requirements
 CPA 9
rerunning a job 20

S

setting a schedule 44
symbol conventions 5

T

TB meaning 51

V

version number conventions 6



6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 100-8280 Japan
