

ITA\_User Instruction Manual

Host group Function

*－*Version1.3*－*

Copyright © NEC Corporation 2019. All rights reserved.

Disclaimer

All the contents of this document are protected by copyright owned by NEC Corporation.

Unauthorized reproduction or copying of all or part of the contents of this document is prohibited.

The contents of this document are subject to change without prior notice in the future.

NEC Corporation is not responsible for any technical or editorial or omissions in this document.

NEC Corporation do not guarantee accuracy, usability, certainty of the content in this document

Trademark

* Linux is registered trademark or trademark of Linus Torvalds, registered in the U.S. and other countries
* Red Hat is registered trademark or trademark of Red Hat, Inc., registered in the U.S. and other countries.
* Apache, Apache Tomcat, and Tomcat are registered trademark or trademark Apache Software Foundation.
* Ansible is registered trademark or trademark of Red Hat, Inc.

The names of other systems, company name and products mentioned in this document are registered trademarks or trademarks of their respective companies.

The® mark and TM mark are not specified in this document.

※「Exastro IT Automation」 is written as 「ITA」 in this document

# Table of contents

[Table of contents 2](#_Toc33082449)

[Introduction 3](#_Toc33082450)

[1 Overview and Advantages of host groups 4](#_Toc33082451)

[1.1 Overview of Host groups 4](#_Toc33082452)

[1.2 Advantages of host groups 5](#_Toc33082453)

[1.2.1 Inheritance of parameters between host groups 5](#_Toc33082454)

[2 Main features of the host group function 8](#_Toc33082455)

[3 Menu group of host group function 9](#_Toc33082456)

[3.1 About the「Host Group Management Console」 menu group 9](#_Toc33082457)

[4 Set information for each host 10](#_Toc33082458)

[4.1 Create parameter sheet 13](#_Toc33082459)

[4.2 Register host group 13](#_Toc33082460)

[4.3 Define the parent-child relationship of host group 14](#_Toc33082461)

[4.4 Associating a host group with an operation and target hosts 15](#_Toc33082462)

[4.5 Register parameter sheet menu 20](#_Toc33082463)

[4.6 Hosting 21](#_Toc33082464)

[4.7 Associate operation with setting value of the item in each target host 23](#_Toc33082465)

[4.8 Reflect operation to the associated target host 23](#_Toc33082466)

[4.9 Reflect substitution values 23](#_Toc33082467)

[5 Host group variables 24](#_Toc33082468)

[5.1 The necessity of host group variable 24](#_Toc33082469)

[5.2 Referencing host group variables 26](#_Toc33082470)

[5.3 Register specific values for host group variables 27](#_Toc33082471)

[6 Applicational Operation 29](#_Toc33082472)

[6.1 Maintenance 29](#_Toc33082473)

[6.2 Change log level 30](#_Toc33082474)

# Introduction

This document describes the functions and operation methods of the ITA host group function (hereinafter referred as Host group) system.

# 1 Overview and Advantages of host groups

## Overview of Host groups

A host group is a group of hosts organized in logical units (Functions/Roles)

The conceptual diagram of the host group is as follows.

all

dc1

dc3

zabbix1

zabbix3

Logical unit

（Host group）

・・・

・・・

hostname11

hostname12

hostname31

・・・

hostname32

Physical unit

（Target host）

The above figure consists of the following elements.

■Host group

Described in the square with a single line frame in given figure dc1、dc3、zabbix1、zabbix3 all are applicable and stated above.

■Target host

Host indicating physical device.

Described as a round square corner in the given figure. Hostname11, hostname12, hostname31, hostname32 are applicable and stated above

The upper host group is called the parent host group, and the lower host group is called the child host group, and they have a parent-child relationship. The lower end host group is associated with target host. In above example the child host group is viewed from dc1 is zabbix1 and the parent host group is viewed from zabbix1 is dc1

A hierarchy can be defined as count 1 host group as 1 level, Count a pair of parent and child as 2 levels. Up to 15 hierarchies can be defined from the host group at the top to the host group at the end. In the example from the above figure, if we go from “all” to “zabbix1”, we get 3 hierarchies.

## Advantages of host groups

### Inheritance of parameters between host groups

Parameters set in parent host group are inherited by child host group.

In the given example below, the parameters set for dc1 (for example, user password) are inherited to zabbix1.However, if there is no specific value, it is not inherited.

all

dc1

dc3

zabbix1

zabbix3

Logical unit

（Hostgroup）

・・・

・・・

hostname11

hostname12

hostname31

・・・

hostname32

Physical unit

（Target host）

zabbix11

zabbix31

zabbix32

By localizing the settings in this way, assignment/change of settings can be simplified.

The parameter settings are automatically inherited even when a child host group is added. In the example below, when one zabbix server is added, zabbix12 is linked to zabbix1.As a result, the settings of zabbix1 are inherited by zabbix12 in the same way follows to zabbix11.

all

dc1

dc3

zabbix1

zabbix3

Logical unit

（Host group）

・・・

・・・

hostname11

hostname12

hostname31

・・・

hostname32

hostname13

Physical unit

（Target host）

zabbix11

zabbix12

zabbix31

zabbix32

A child host group can be linked to multiple parent host groups. Here explains the inheritance of parameters when there are multiple parent host groups. In the example below, zabbix11 and zabbix12 belongs to zabbix1 and virtual server.

zabbix32

all

dc1

dc3

zabbix1

zabbix3

Logical unit

（Host group）

・・・

・・・

hostname11

hostname12

hostname31

・・・

hostname32

virtualserver

hostname13

Physical unit

（Target host）

dcx

zabbix11

zabbix12

zabbix31

If parameters are overlap between the host groups, they are applied from the lowest host group. In the above example, if the same parameters are set for dcx and zabbix1, the parameters inherited by zabbix11 and zabbix12 are not dcx but lower zabbix1 parameters.

If the parameters overlap in the same hierarchy, they are inherited from the parent host group with higher priority. In the above example, if the same parameters are set for virtual server and zabbix1, if the priority of virtual server is1 and if the priority of zabbix1 is 2, then the priority of zabbix1 is higher, so the parameter of zabbix1 is inherited by child host group.

How to set the priority is described in 「4.2 Register host group」

# 2. Main features of the host group function

The main features of the host group function are classified into the following categories.

1. Web

Web content。A screen that allows to use the host group function provided by the browser.

1. BackYard

A resident process that runs on a server independently of Web content.

# 3. Menu group of host group function

The menu groups of the host group function and the menus belonging to them are as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| **Menu**  **group** | **Affiliated menu** | **Section** | **Description** |
| Hostgroup management console | Hostgroup Management | 4.2 | Register host group |
| Hostgroup parent-child link list | 4.3 | Define the parent-child relationship of host group |
| Host link list | 4.4 | Associate host group with operation and target hosts |
| Create Hostgroup Variable link list | 5.2 | Register host group variables |
| Hostgroup variable link list | 5.3 | Register specific values for host group variables |

## About the「Host Group Management Console」 menu group

Host group management, host group parent-child link list, host link list are the menu required to register host groups and define which operation is to be performed on which host. Details of the operation are described in 「4 Set information for each host」.

Create Host group variable list and host group variable linking are menus required to convert

Host group variables and register specific values. Details of the operation are explained in

「5Host Group Variables」.

# 4. Set information for each host

The following table shows the workflow for registering host groups and setting information for each host using Web content.

Details of the work are described in each section.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Section** | **Contents of work** | **User operation** | **Menu group used** | **Menu used** | **Remarks** |
| 4.1 | Create parameter sheet | Yes | Create parameter list menu | Menu creation information  Menu item creation information  Execute menu creation  Menu creation management | ― |
| 4.2 | Register host group | Yes | Host group management | Host group management | ― |
| 4.3 | Define a parent-child relationship of host group- | Yes | Host group management | Host group parent child link list | ― |
| 4.4 | Associate host group with operation and target hosts | Yes | Host group management | Host link list | ― |
| 4.5 | Register parameter sheet menu | Yes | ※2 | Menu created in「4.1Create parameter sheet」 | ― |
| 4.6 | Hosting | No  ※1 | ※2 | Menu created in「4.1Create parameter sheet」 | Manual registration/Update unavailable |
| 4.7 | Associate operation with setting value of the item in each target host | Yes | ※2 | Substitute value automatic registration setting | Refer to「Parameter Management Menu Guide」 for substitution value automatic registration setting. |
| 4.8 | Reflect operation to the associated target host | No  ※1 | ※3 | Target host | Refer to「Parameter Management Menu Guide」 for substitution value automatic registration setting. |
| 4.9 | Reflect substitution values | No  ※1 | ※3 | Substitution value list | Refer to「Parameter Management Menu Guide」 for substitution value automatic registration setting. |

※1

No user operation is required because it is executed by internal processing. Processing results can be checked on the menu screen.

※2

Menu group specified in 「4.1 Create parameter sheet」.

※3

Menu group that performs substitution value automatic registration.

After each operation, the image of the record will be as shown in the table below.

|  |  |  |
| --- | --- | --- |
| **Section** | **Contents of work** | **Image of record** |
| 4.2 | Register host group | |  | | --- | | **Host group** | | HG\_1 | | HG\_2 | | hg\_1a | | hg\_1b | | hg\_2a | | hg\_2b | |
| 4.3 | Define parent-child relationship of host group | |  |  | | --- | --- | | **Parent host group** | **Child host group** | | HG\_1 | hg\_1a | | HG\_1 | hg\_1b | | HG\_2 | hg\_2a | | HG\_2 | hg\_2b | |
| 4.4 | Associate host group with operation and target hosts | |  |  |  | | --- | --- | --- | | **host group** | **Operation** | **target host** | | hg\_1a | 2017/10/31\_1001\_OP1 | host\_1a | | hg\_1b | 2017/10/31\_1001\_OP1 | host\_1b | | hg\_2a | 2017/10/31\_1001\_OP1 | host\_2a | | hg\_2b | 2017/10/31\_1001\_OP1 | host\_2b | |
| 4.5 | Register parameter sheet menu | |  |  |  |  | | --- | --- | --- | --- | | **target host**  **or　Host group** | **Operation** | **Item 1** | **Item 2** | | HG\_1 | 2017/10/31\_1001\_OP1 | 111 | AAA | | HG\_2 | 2017/10/31\_1001\_OP1 | ― | BBB | |
| 4.6 | Hosting | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Target host** | **Operation** | | | **Item1** | **Item 2** | | **Scheduled execution date** | **ID** | **Operation** | | host\_1a | 2017/10/31 | 1001 | OP1 | 111 | AAA | | host\_1b | 2017/10/31 | 1001 | OP1 | 111 | AAA | | host\_2a | 2017/10/31 | 1001 | OP1 | ― | BBB | | host\_2b | 2017/10/31 | 1001 | OP1 | ― | BBB | |
| 4.7 | Associate operation with setting value of the item in each target host | |  |  |  |  | | --- | --- | --- | --- | | **Menu　group name：Menu name** | **Item** | **Movement** | **Variable name** | | Parameter management (hosting): Created menu name | Item 1 | Movement1 | VAR\_variable1 | | Parameter management (hosting): Created menu name | Item 2 | Movement2 | VAR\_variable2 | |

|  |  |  |
| --- | --- | --- |
| **Section** | **Contents of work** | **Image of record** |
| 4.8 | Reflect operation to the associated target host. | |  |  |  | | --- | --- | --- | | **Operation** | **Movement** | **Target host** | | 1001\_OP1 | Movement1 | host\_1a | | 1001\_OP1 | Movement1 | host\_1b | | 1001\_OP1 | Movement2 | host\_1a | | 1001\_OP1 | Movement2 | host\_1b | | 1001\_OP1 | Movement2 | host\_2a | | 1001\_OP1 | Movement2 | host\_2b | |
| 4.9 | Reflect substitution values | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Operation** | **Movement** | **Target host** | **Variable name** | **Specific value** | | 1001\_OP1 | Movement1 | host\_1a | VAR\_variable1 | 111 | | 1001\_OP1 | Movement1 | host\_1b | VAR\_variable1 | 111 | | 1001\_OP1 | Movement2 | host\_1a | VAR\_variable2 | AAA | | 1001\_OP1 | Movement2 | host\_1b | VAR\_variable2 | AAA | | 1001\_OP1 | Movement2 | host\_2a | VAR\_variable2 | BBB | | 1001\_OP1 | Movement2 | host\_2b | VAR\_variable2 | BBB | |

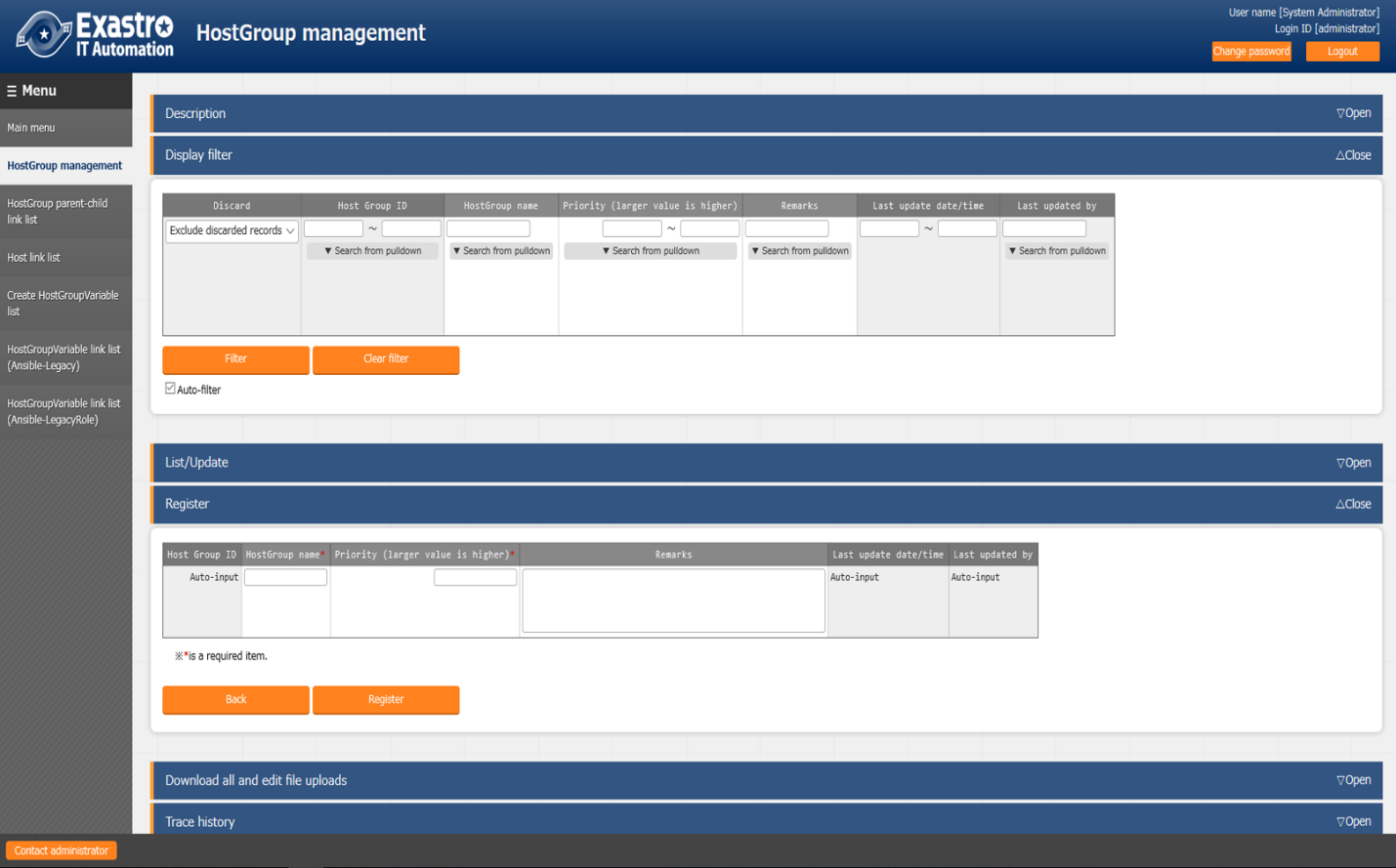
## Create parameter sheet

Create parameter sheet menu using the parameter list creation information.

For the details of the parameter sheet creation function, please refer to 「User instruction manual\_Parameter Sheet Creation Function.」

## Register host group

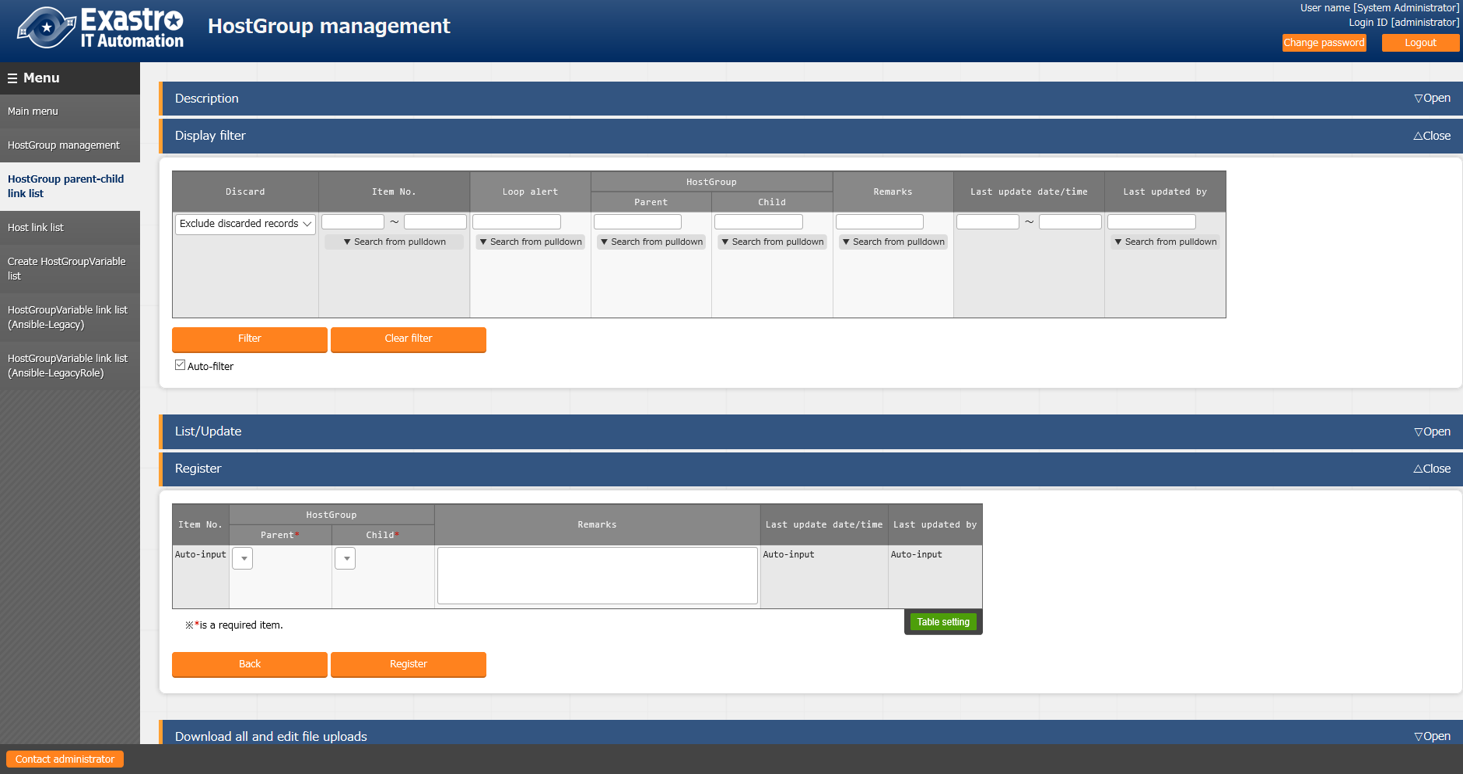
Register host group using the host group management menu.



|  |  |
| --- | --- |
| **Column name** | **Description** |
| Host group name | Enter the name of the host group |
| Priority (larger value is higher) | Enter the priority. The input range is from 1 to 2,147,483,647 |

## 4.3 Define the parent-child relationship of host group

Use the Host group parent-child link list menu to define the parent-child relationship of the host group.

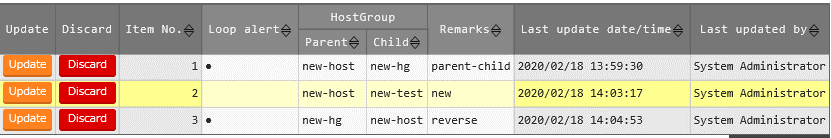


|  |  |  |
| --- | --- | --- |
| **Column name** | | **Description** |
| Host group | Parent | Select the parent host group name |
| Child | Select the child host group name associate with the parent host group |

If there is host group that has parent-child loop, which is 「●」 displayed in the Loop alert column

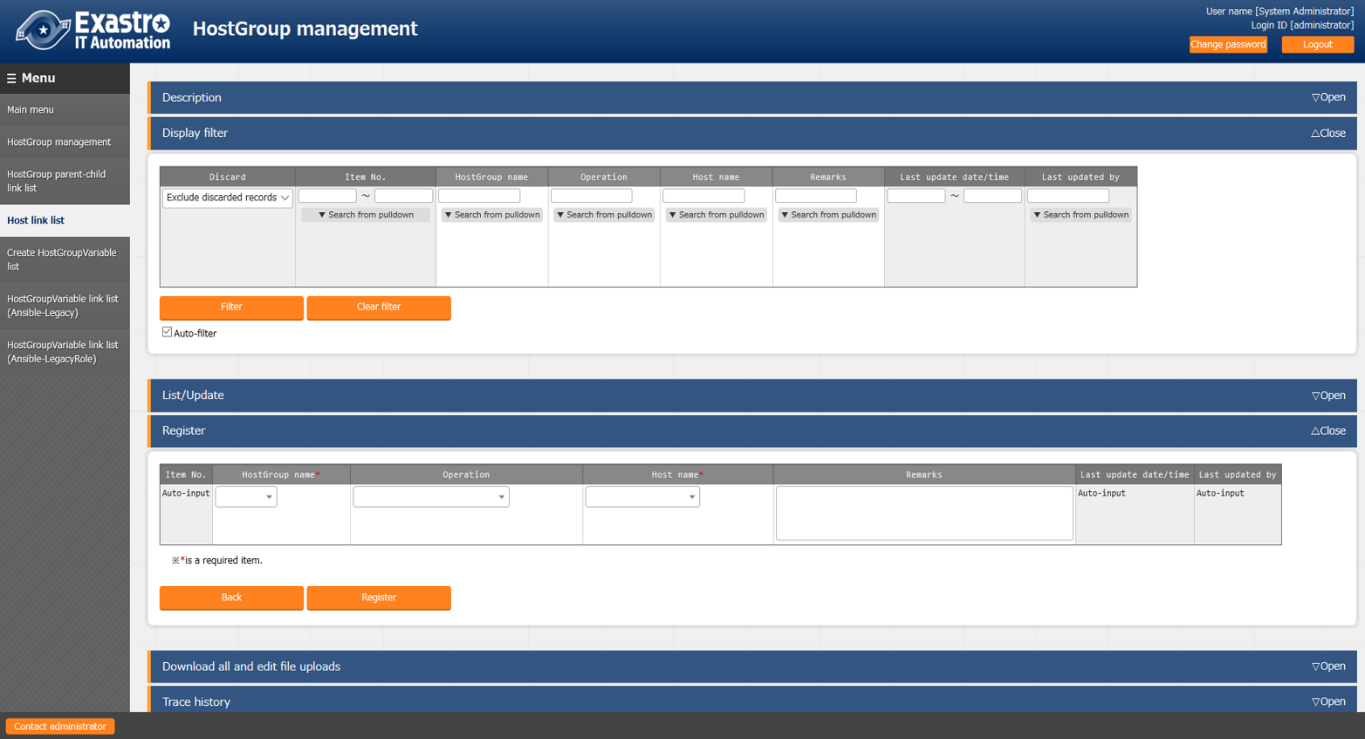
Of the display results of display filter

In the example below. Although the parent-child relationship between zabbix1 (parent) and zabbix11 (child) has already been defined, it also defines the reverse parent-child relationship of zabbix11 parent) and zabbix1 (child), and the parent-child relationship is a loop. Please make sure not to create a loop because if there is a loop already, the internal processing is described below 「Host group decomposition function」, 「Host group variable conversion function」, and 「Host group variable registration function」will not execute.



## Associating a host group with an operation and target hosts

Use the host link list menu to register the target host to be associated with the host group and operation.



|  |  |
| --- | --- |
| **Column name** | **Description** |
| Host group name | Select a host group. |
| Operation | Select a operation.※1 |
| Host name | Select a target host. |

By associating the host group and operation with the target host, you can select the target host in the host group as shown in the figure below.

Host Group A

Host 1

Host 2

Host 3

Operation 1

Operation 4

Host group A

Host 1

Host 2

Host 3

Operation 3

オペレーション３

Operation 2

Host Group A

Host 1

Host 2

Host 3

Example 1

When all hosts belonging to a host group are to be set as operation target host

Example 2

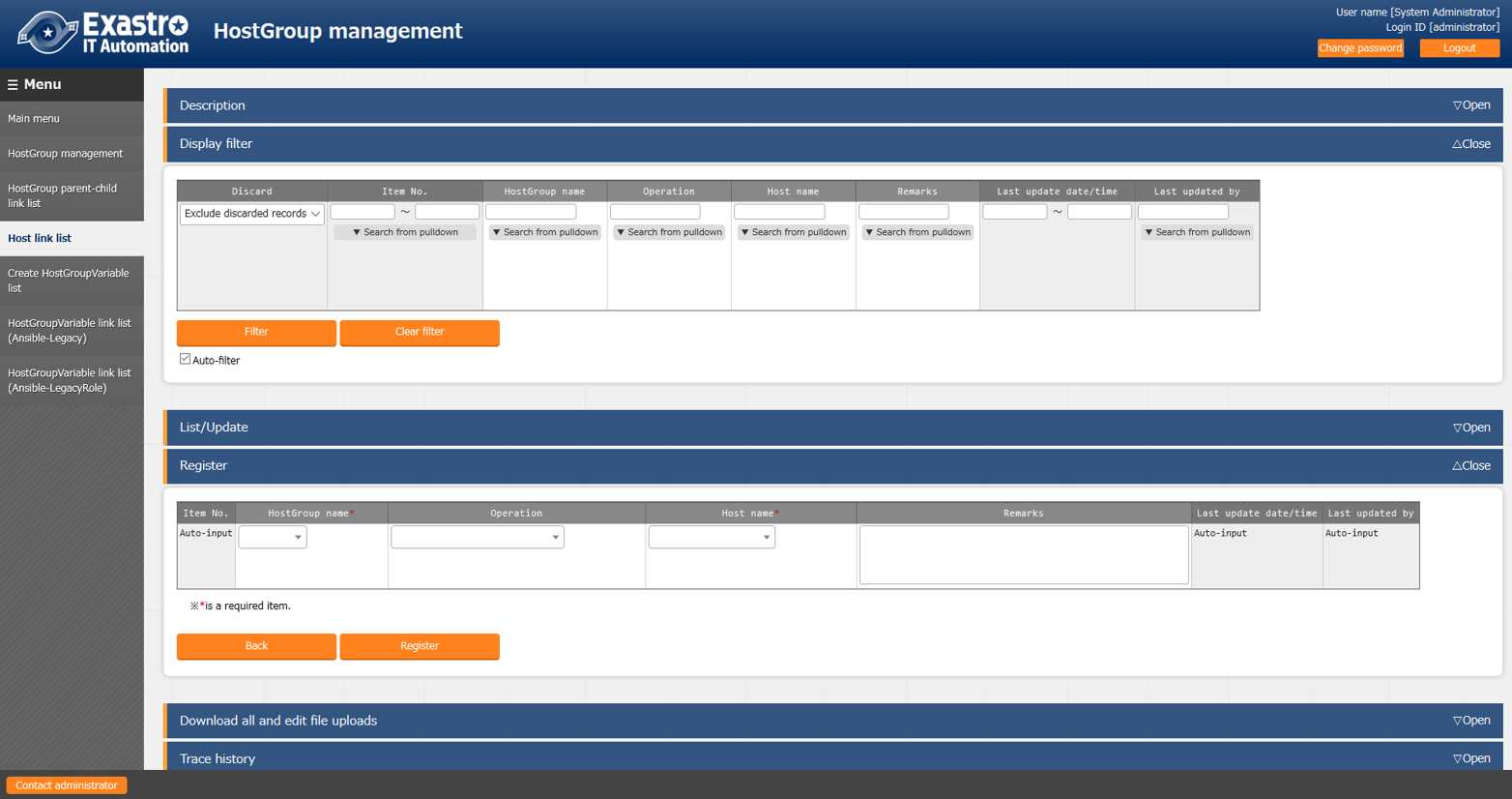
When some hosts are to be set as operation target host

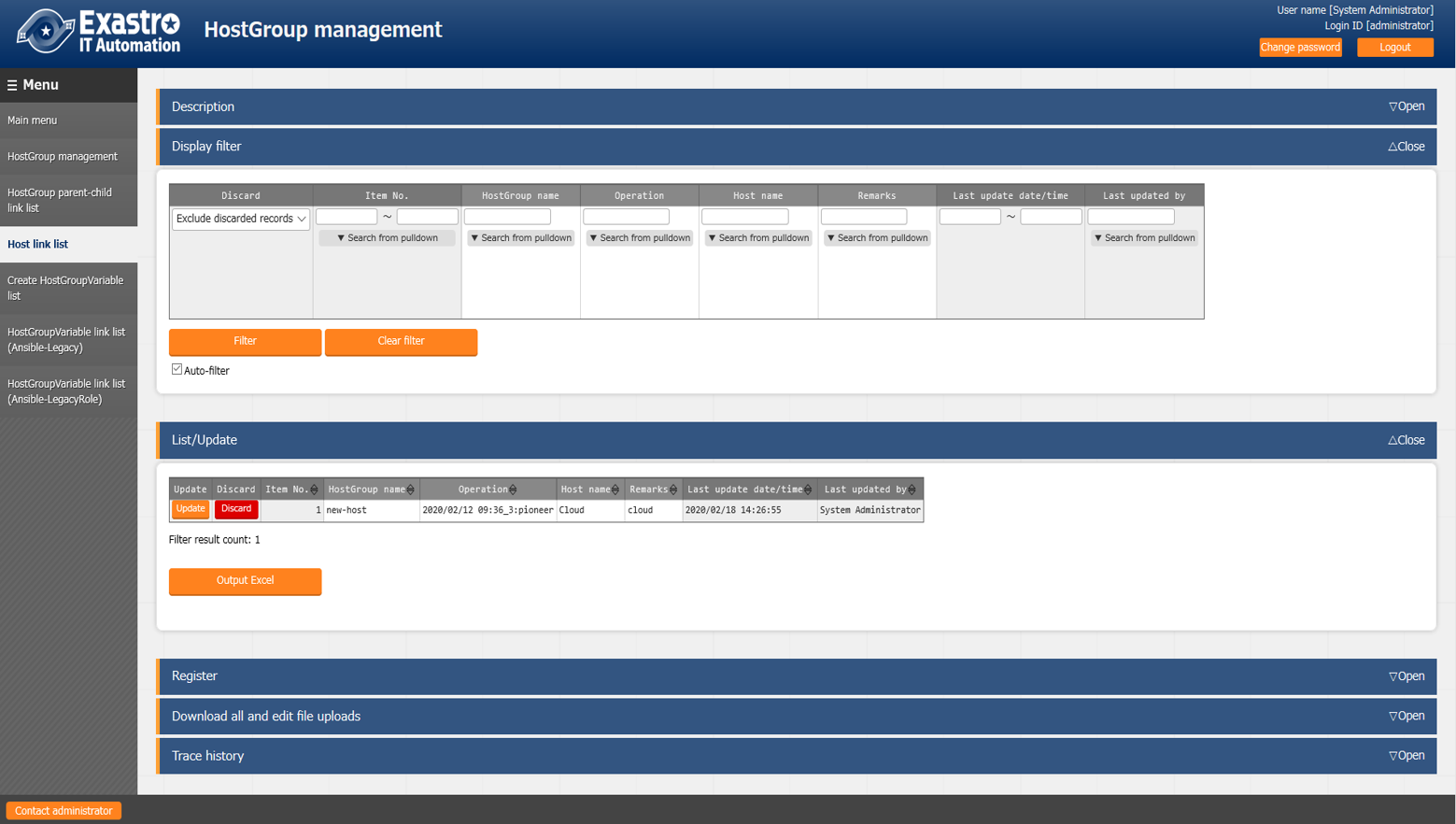
Example 3

Combination of patterns in Example 2

※1

In the host link list menu, the operation can also be registered as NULL. In case if NULL is registered, association will be enabled for all operations.





In the above example, the association of host groups zabbix1 and zabbix2 where the operation is registered as NULL which is valid for all operations. On the other hand, the host group zabbix3 for which the operation has been registered is valid only for the registered operation for 「Operator①」.

Host group association in operation「Operator①」

all

Logical unit

（Host group）

hostname11

hostname12

hostname31

hostname13

Physical unit

（Target host）

zabbix11

zabbix12

zabbix31

Host group association other than operation 「Operator①」

all

Logical unit

（Host group）

hostname11

hostname12

hostname13

Physical unit

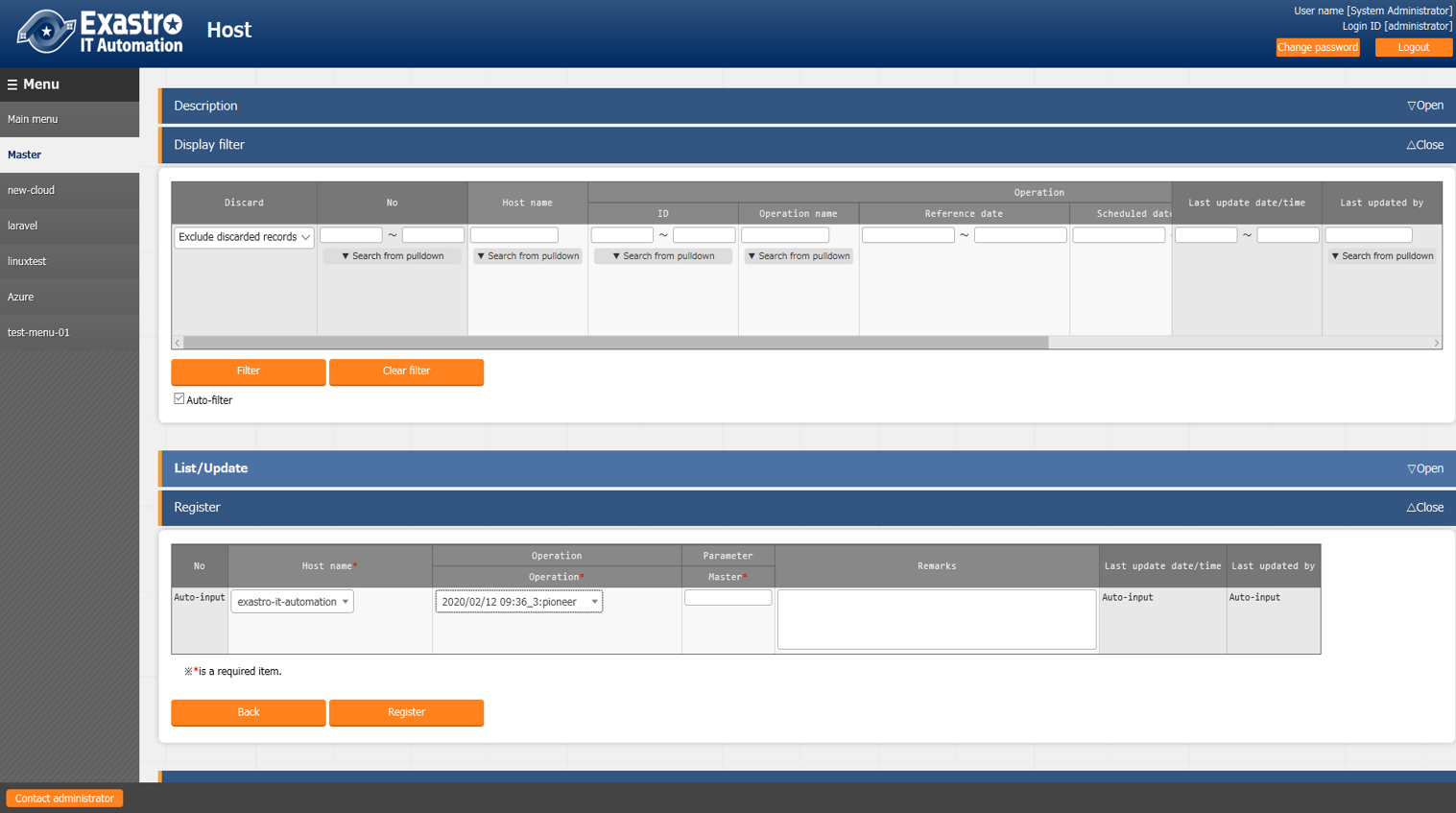
（Target host）

zabbix11

zabbix12

## Register parameter sheet menu

In the parameter sheet menu for the host group created in 「4.1 Create Parameter Sheet」,the specific value for each operation is registered in the item with the target host or host group. After registration, users can View/Update/Discard/Restore the registration.



|  |  |
| --- | --- |
| **Column name** | **Description** |
| Host name/Host group name | Select the target host or host group. The prefix [H] is the target host and [HG] is the host group. |
| Operation | Select an operation. |
| Item name  (Item name defined in 「4.1 Parameter Sheet Creation」) | Enter the specific value of the item.  The specific values entered are reflected as the specific value for the operations and the variables associated with Movement and the target host in the 「4.9 Reflection of the substitution value」 |

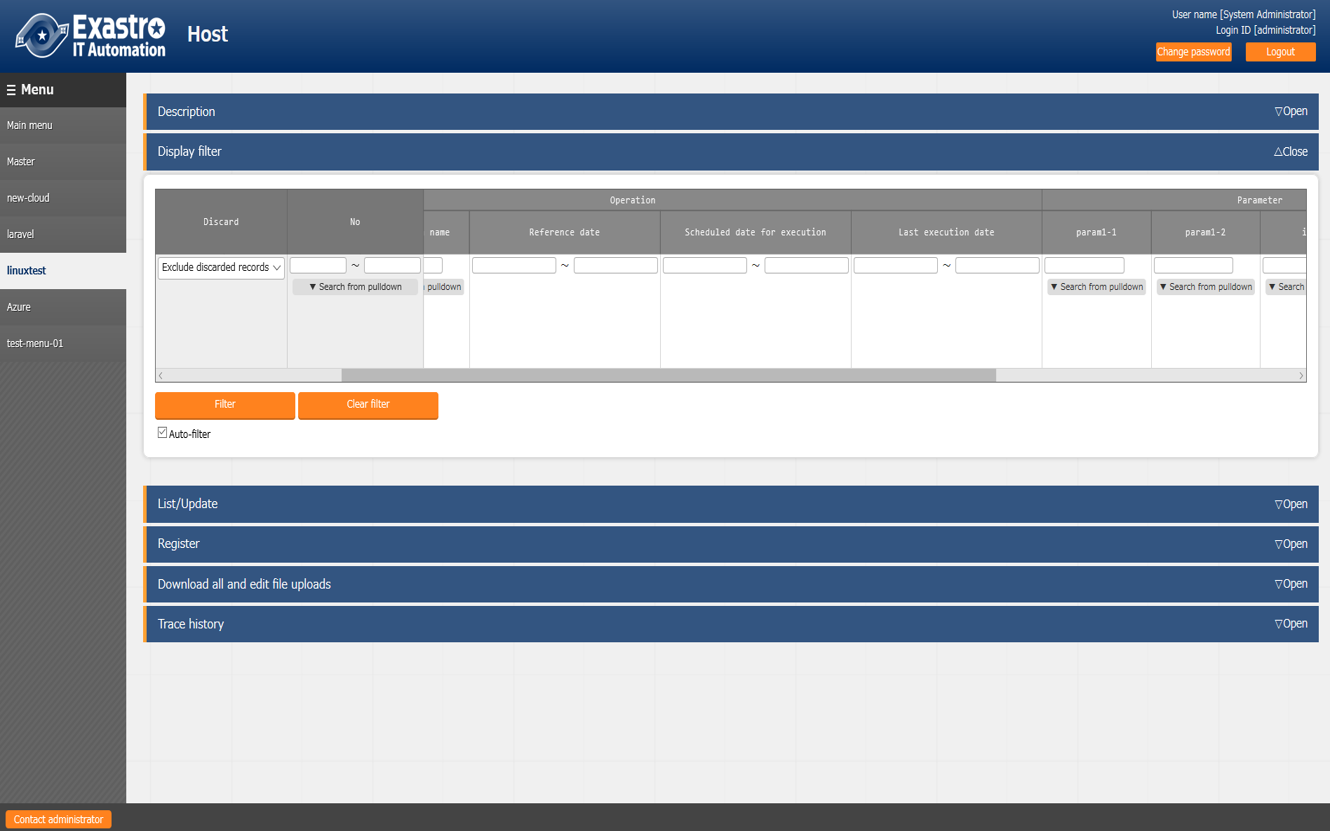
The combination of 「Host name/Host group」 and 「Operation」 is registered uniquely. The same host can be registered if combined with different operations.

## Hosting

The information registered 「4.5 Register the menu in parameter sheet」 menu is compiled for each operation by the internal processing 「Host group decomposition function」and is further inherited to the target host until according to the host group association.

The information inherited by each target host can be viewed in the parameter sheet menu created in 「4.1 Create Parameter Sheet」

Users can only view, but cannot Register/Update/Discard/Restore.



An example of the workflow of hosting is described below.

（1） The items registered in the parameter sheet menu are as follows.

（Information registered in 「4.5 Register the parameter sheet menu」）

|  |  |  |  |
| --- | --- | --- | --- |
| **Target host**  **or　Host group** | **Operation** | **Item 1** | **Item 2** |
| HG\_1 | 2017/10/31\_1001\_OP1 | 111 | AAA |
| HG\_2 | 2017/10/31\_1001\_OP1 | ― | BBB |
| host\_1a | 2017/10/31\_1001\_OP1 | 222 | ― |

（2） The parent-child relationship of the host group is as follows.

（Information registered 「4.3 Defining parent-child relationship of host group」）

|  |  |
| --- | --- |
| **Parent host group** | **Child host group** |
| HG\_1 | hg\_1a |
| HG\_1 | hg\_1b |
| HG\_2 | hg\_2a |
| HG\_2 | hg\_2b |

（3）The association information of the Host group, operation, and target host is as follows.

（Information registered in 「4.4 Associate host group with operation and target hosts」）

|  |  |  |
| --- | --- | --- |
| **Host group** | **Operation** | **Target host** |
| hg\_1a | 2017/10/31\_1001\_OP1 | host\_1a |
| hg\_1b | 2017/10/31\_1001\_OP1 | host\_1b |
| hg\_2a | 2017/10/31\_1001\_OP1 | host\_2a |
| hg\_2b | 2017/10/31\_1001\_OP1 | host\_2b |

（4） If hosting is performed with the information registered in （1）～（3）,the record are as follows, users can know that the information is set to the target host until that belongs to the host group.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Target host** | **Operation** | | | **Item 1** | **Item 2** |
| **Scheduled execution date** | **ID** | **Operation name** |
| host\_1a | 2017/10/31 | 1001 | OP1 | 222　（※1） | AAA　（※2） |
| host\_1b | 2017/10/31 | 1001 | OP1 | 111 | AAA |
| host\_2a | 2017/10/31 | 1001 | OP1 | ―　（※3） | BBB |
| host\_2b | 2017/10/31 | 1001 | OP1 | ―　（※3） | BBB |

（※1）　When you register an item in a host group and target host, the item in the target host takes the priority. Therefore, 「222」 is registered in host\_1a is applied.

（※2）　The item of the target host has priority, but in case if it is empty, it is inherited from the higher level. Since item 2 of host1\_a was empty, So 「AAA」 of the parent host group HG\_1 was inherited.

（※3）　Item1 of host\_2a and host\_2b and is empty because item 1 of HG\_2 is empty

## Associate operation with setting value of the item in each target host

Associate the menus and items that were linked on the 「Substitute value automatic registration setting」menu screen with the variables of Movement. Users can Register/Update/Discard/Restore.

The registered information is reflected to the 「Substitution value list」 menu screen and 「Target host」 menu screen by internal process.

For more details, please refer to 「Parameter Management Menu Guide」.

## Reflect operation to the associated target host

The target host associated with the operation is automatically reflected. user can check the result on 「Target host」 menu screen.

For more details, please refer to 「Parameter Management Menu Guide」.

## Reflect substitution values

For each operation, the specific value is assign to the variable 「VAR\_」 in the Playbook or template file used in the target Movement is automatically reflected.

The specific value here is the specific value entered in 「4.5 Register the menu in parameter sheet」.The reflection result can be confirmed on the 「Substitution value list」 menu screen.

For more details, please refer to 「Parameter Management Menu Guide」.

# 5. Host group variables

If user wants to refer a variable of the host name included in the host group as a specific value in a construction material (for example, the Playbook, etc.), then it is necessary to perform a procedure called Host group variable conversion.

## The necessity of host group variable

As an example, if you want to describe in the playbook that 「Process A is executed when the target host belongs to the host group “my\_group” in Ansible」 ,The following will be performed in ITA.

- Process A

when: inventory\_hostname in "{{ VAR\_hostgroup\_my\_group }}"

The format of the host group variable is 「VAR\_hostgroup \_host group name」.

In the above example, a list of host names belonging to my\_group can be registered as specific values

In the host group variable VAR\_hostgroup\_my\_group.

As another example, when the host group in the conceptual diagram described in 「0 Overview of Host groups」 is converted into host group variables, The specific values to be registered are as shown in table below.

all

dc1

dc3

zabbix1

zabbix3

Logical unit

（Host group）

hostname11

hostname12

hostname31

hostname32

Physical unit

（target host）

zabbix11

zabbix12

zabbix31

zabbix32

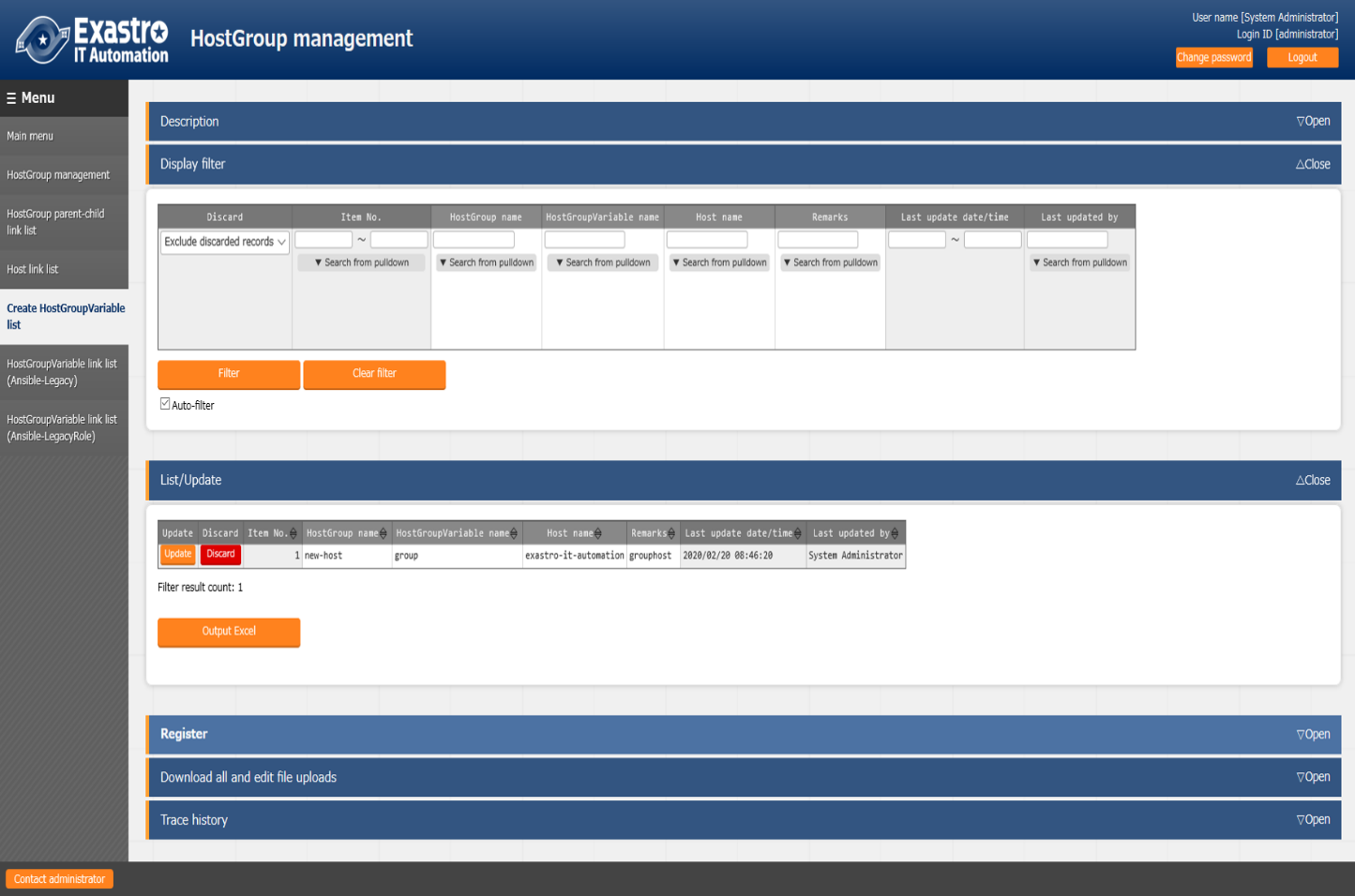
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Host group name** | **Host group variable name** | **Specific value**  **（The target host name that belongs to the host group）** |
| 1 | all | VAR\_hostgroup\_all | hostname11 |
| 2 | hostname12 |
| 3 | hostname31 |
| 4 | hostname32 |
| 5 | dc1 | VAR\_hostgroup\_dc1 | hostname11 |
| 6 | hostname12 |
| 7 | zabbix1 | VAR\_hostgroup\_zabbix1 | hostname11 |
| 8 | hostname12 |
| 9 | zabbix11 | VAR\_hostgroup\_zabbix11 | hostname11 |
| 10 | zabbix12 | VAR\_hostgroup\_zabbix12 | hostname12 |
| 11 | dc3 | VAR\_hostgroup\_dc3 | hostname31 |
| 12 | hostname32 |
| 13 | zabbix3 | VAR\_hostgroup\_zabbix3 | hostname31 |
| 14 | hostname32 |
| 15 | zabbix31 | VAR\_hostgroup\_zabbix31 | hostname31 |
| 16 | zabbix32 | VAR\_hostgroup\_zabbix32 | hostname32 |

The advantage of using host group variable conversion is that if the host group configuration changes due to change in parent-child linkage etc., it is automatically reflected in the specific values of the host group variables, so there is no need to manually change the contents of the specific values.

## Referencing host group variables

Host group variable is automatically performed by the internal process 「Host group variable conversion function」. Normally, manual updates are not possible.

When referring to the list of host group variables in the host group variable menu, if the last updater is the 「Host group variable function」, the host group variables are converted by internal process.

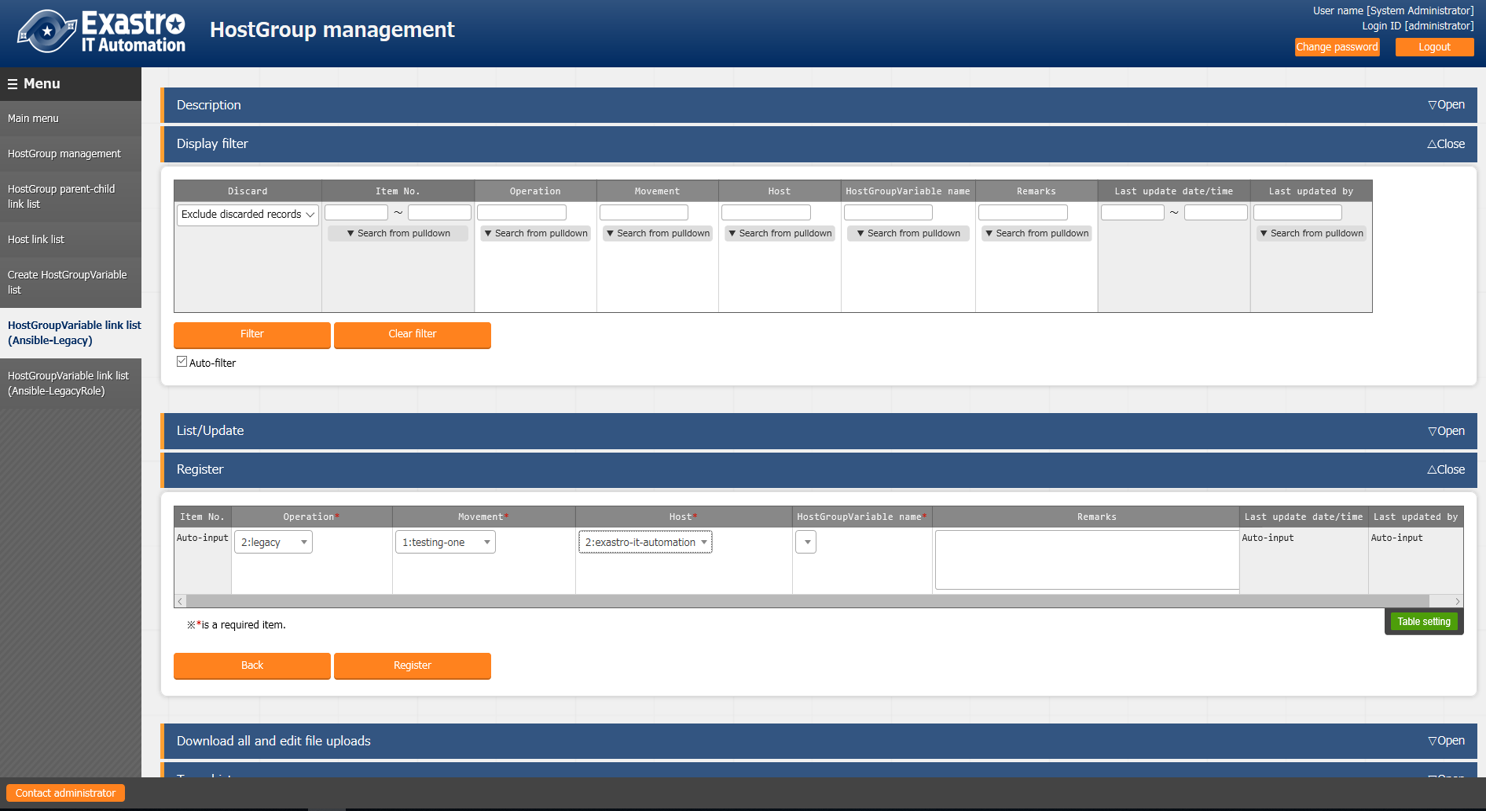


## Register specific values for host group variables

Use the host group variable link list menu to link the operation, Movement, target host and host group variable.

After linking, the specific value (host name belonging to corresponding host group) is registered in the host group variable by the internal process 「Host group variable registration function」.

There are multiple menus for linking host group variables. Here, the 「Host Group variable link list (Ansible-Legacy)」 menu will be described as an example.



|  |  |
| --- | --- |
| **Column name** | **Description** |
| Operation | Select an operation |
| Movement | Select a Movement. |
| Host | Select the target host |
| Host group variable name | Select the host group variable name |

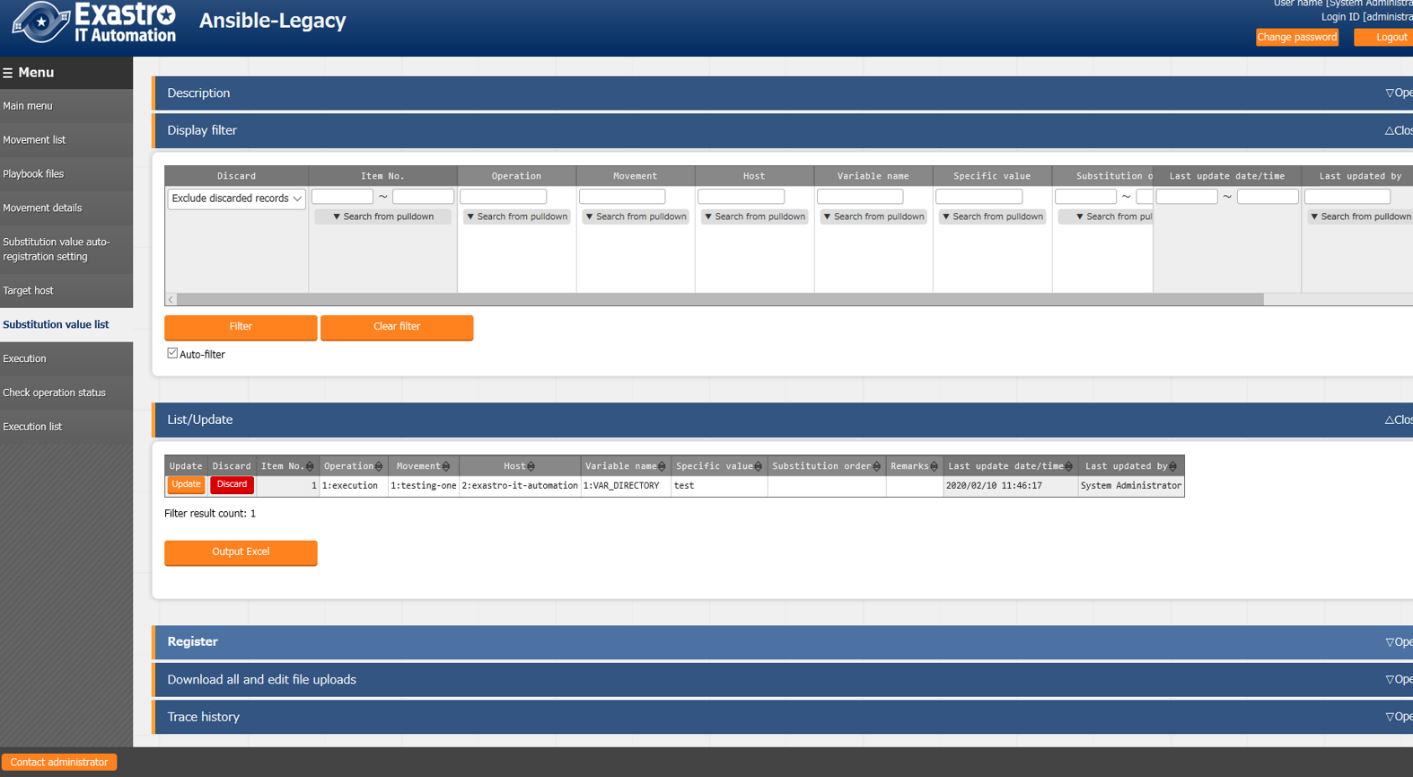
An example of specific value registration is explained under the following assumptions.

|  |  |
| --- | --- |
| **No** | **Assumption** |
| 1 | HostA,hostB,hostC belong to the host group my\_group |
| 2 | The target hosts for Movement1 are hostA and B |
| 3 | Host group i.e my\_group is converted to host group variable  (VAR\_hostgroup\_my\_group) |
| 4 | The name of the operation to be executed is Operation1 |

Assumption 1 is defined in the 「Host group Parent-Child link list menu」

Assumption 2 to 4 are linked using the 「Host Group Parent-Child link list menu」

The record of the linked information is entered as follows.



After that, the host belonging to the VAR\_hostgroup\_my\_group is registered as the specific value by executing the host group variable registration function.

The result of registering the specific value can be seen in the 「Substitution value list」 menu.

Please do not Update/Delete registration results.

# 6. Applicational Operation

Operations that utilize the host group function include not only input by the user from the browser screen of the client PC, but also operations by system operation and maintenance. The operation and maintenance are as follows.

■　Maintenance

■　Change log level

## Maintenance

The following files are required to Start/Stop/Restart the host group function process.

|  |  |
| --- | --- |
| **Description** | **Target file name** |
| Host group variable conversion function  Convert a host group and its associated host into variables. | ky\_hostgroup\_make\_var.service |
| Host group variable registration function.  In the Ansible-Legacy console, data is set in ITA target host and assigned the value management based on the host group variable association menu. | ky\_hostgroup\_regist\_var\_legacy.service |
| Host group variable registration function  In the Ansible-Legacy console, data is set in ITA target host and assigned the value management based on the host group variable association menu. | ky\_hostgroup\_regist\_var\_legacy\_role.service |
| Host group decomposition function  Decompose design information which is entered in host group and host group units. | ky\_hostgroup\_split.service |
| Loop check function  This function checks whether the parent-child relationship of the host group is in a loop. | ky\_hostgroup\_check\_loop.service |

The target file is stored in 「/usr/lib/systemd/system」.The method of Start/Stop/Restart a process is as follows. Please execute the command with root privileges.

1. Process start

# systemctl start ky\_hostgroup\_split.service 

1. Process stop

# systemctl stop ky\_hostgroup\_split.service 

1. Process restart

# systemctl restart ky\_hostgroup\_split.service 

Similarly, please replace each target file name to start/stop/restart the process.

## Change log level

1. Change to NORMAL level

Rewrite 「DEBUG」 on the 8 th line of the target file to 「NORMAL」

Log level setting file： <Install directory>/ita-root/confs/backyardconfs/ita\_env

1. Change to DEBUG level  
   Rewrite 「NORMAL」 on the 8th line of the target file to 「DEBUG」  
   Log level setting file： <Install directory>/ita-root/confs/backyardconfs/ita\_env

After rewriting, the log level change is effective after restarting (restart) the process.

Please refer to 「6.1 Maintenance」 to restart.

Log file output destination： <Install directory>/ita-root/logs/backyardlogs