

ITA\_User Instruction manual

Construction file management function

*－*Version 1.5*－*

Copyright © NEC Corporation 2020. 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 errors 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、Tomcat are registered trademarks or trademarks of Apache Software Foundation.
* Ansible is a registered trademark or trademark of the 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 is not specified in this document.

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

**Table of Contents**

[Introduction 3](#_Toc31620717)

[1. Overview of construction file management function 4](#_Toc31620718)

[1.1 What is construction file management 4](#_Toc31620719)

[1.2 System configuration 4](#_Toc31620720)

[1.3 The feature of construction file management function 4](#_Toc31620721)

[2. Construction file management Menu configuration 5](#_Toc31620722)

[2.1 Menu/Screen list 5](#_Toc31620727)

[2.2 Construction file management　Status list 6](#_Toc31620735)

[3. Construction file management Operation procedure 7](#_Toc31620738)

[3.1 Workflow 7](#_Toc31620740)

[**3.1.1** **Initial configuration Git association** 7](#_Toc31620745)

[**3.1.2** **Register management target file** 8](#_Toc31620746)

[**3.1.3** **Check-in / Check-out files** 9](#_Toc31620747)

[**3.1.4** **Cancellation workflow** 11](#_Toc31620748)

[4. Construction file management Function・operation method 13](#_Toc31620749)

[4.1. File control master 13](#_Toc31620752)

[**4.1.1.** **Interface information** 13](#_Toc31620753)

[**4.1.2.** **Directory master** 15](#_Toc31620754)

[**4.1.3** **File master** 17](#_Toc31620759)

[**4.1.4** **File list** 19](#_Toc31620763)

[**4.1.5** **File link list (Ansible)** 21](#_Toc31620764)

[**4.1.7** **File link list（OpenStack）** 23](#_Toc31620775)

[4.2 File control check-in/check-out console 25](#_Toc31620776)

[**4.2.1.** **Browse** 25](#_Toc31620783)

[**4.2.2.** **Check-out request** 26](#_Toc31620792)

[**4.2.3.** **Check-out** 27](#_Toc31620801)

[**4.2.4** **Check-in request** 28](#_Toc31620818)

[**4.2.5** **Check-in** 29](#_Toc31620819)

[**4.2.6** **Cancel** 30](#_Toc31620820)

[**4.2.7** **Automatic difference extraction function** 31](#_Toc31620821)

[5. Material Linkage function 32](#_Toc31620822)

[5.1 What is Material Linkage function 32](#_Toc31620825)

[5.2 Image of Material Linkage function 33](#_Toc31620826)

[6. Application operation 34](#_Toc31620827)

[6.1 Maintenance 34](#_Toc31620830)

[6.2 Change log level 35](#_Toc31620831)

# Introduction

This document explains the function and the operation method of ITA construction file management function (referred to as construction file management hereafter) system.

# Overview of construction file management function

## What is construction file management

Construction file management is for managing the check-in/check-out of files such as document and contents, and version controlling by using version control tool Git.

## System configuration

Construction file management is divided into construction file management function part and Git.

Git can run on the same server with ITA or on a separate server.

## The feature of construction file management function

The main function of construction file management are classified into the following categories.

1. Web

Web content. The construction file management function screen provided on the browser.

1. BackYard

Resident processes that runs on a server independent from the Web content.

# Construction file management Menu configuration



This chapter explains the menu configuration of construction file management function.  
For the method to login the web console or the element / basic operation of menu screen, please refer to “User instruction manual - Basic console”.



## Menu/Screen list



The list of console menu used in construction file management is as below.

Table 2.1-1 Construction file management menu / screen list

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Menu group** | **Menu・Screen** | **Overview** |
| 1 | File control management console | Interface information | Initial synchronization with Git |
| 2 | Directory master | Register the file storage directory |
| 3 | File Master | Register management target file |
| 4 | File list | The data of all managed file  Download all management file at once |
| 5 | File link list(Ansible) | Link file to Ansible driver menu |
| 6 | File link list(OpenStack) | Link file to Openstack driver menu |
| 7 | File control  check-in/check-out console | Browse | Browse all check-in/check-out request data |
| 8 | Check-out request | Request for file check-out |
| 9 | Check out | Approve/Reject file check-out request |
| 10 | Check-in request | Request for file check-in |
| 11 | Check in | Approve/Reject file check-in request |
| 12 | Cancel | Request for cancellation |



## Construction file management　Status list



The file status used in construction file management is listed below.

**Table 2.2-1 File status list**

|  |  |  |
| --- | --- | --- |
| **No** | **Status** | **Overview** |
| 1 | Check out requested | The data registered for check out request.  Update the status of data to approve/reject the check out request. |
| 2 | Check out requested (duplicate) | The data registered for check out request.  The request will become this status if the request target file has already been registered in status No3~5, 7~8.  ※The target file will be able to check out if it's status become "check in complete". |
| 3 | Checking out | The data whose check out request has been approved.  Update the status of this data to request for check in. |
| 4 | Check in requested | The data registered for check in request.  Update the status of the data to approve/reject the check in request. |
| 5 | Checking in | The data whose check in request has been approved.  Update the status of this data to link with Git.  ※In the case of auto check-in, update the "check-in requested" data to link with Git. |
| 6 | Check in complete | The data whose file update has been completed.  Please make new check out request to update the file again. |
| 7 | Return(check-out request) | The data whose check out request is rejected.  Please check the request content and register for request again. |
| 8 | Return(check-in request) | The data whose check in request is rejected.  Please check the request content and register for request again. |
| 9 | Cancel | The data which is cancelled during process.  Please refer to 4.2.6 for status that has been cancelled. |

# Construction file management Operation procedure

Explain the operation procedure of construction file management function.



## Workflow

The standard workflow of each console in construction file management is as below.

The details of each operation is described in next section.

**Legend**

Menu Name

File status

Content

Reference chapter

Start

Execute

Comment

Process

branch

End



### **Initial configuration Git association**

　　　When using construction file management function, it's required to associate version control tool Git with ITA as initial configuration.

|  |  |  |
| --- | --- | --- |
| **File operator** | **System administrator** | **BackYard** |
|  | Interface information  Initial synchronization with Git.  4.1.1 Interface information |  |

Figure 3.1-1 Workflow of Initial configuration　Git association

### **Register management target file**

　　 In order to manage target file safely, the system administrator registers storage directory and management target file, and if necessary, configure the association with Ansible, etc.

|  |  |  |
| --- | --- | --- |
| **File Operator** | **System administrator** | **BackYard** |
|  | Directory master  Register the file storage directory of management target file  4.1.2 Directory master  Manage file linkage  It's able to link with the processing mode (Legacy, etc.) of Ansible if the registered file is related to Ansible.  44.1.5 File link list (Ansible)  File Master  Register the management target file  44.1.3 File master |  |

Figure 3.1-2 Workflow ②

### **Check-in / Check-out files**

　　　The user check-in/ check-out the files that are registered by the system administrator. For file check-in/check-out, there are "Auto check-in" that doesn't need administrator's approval procedure and "Manual check-in" that requires approval procedure. It's possible to set "Auto check-in" and "Manual check-out" for each file accordingly.

1. Flow of auto check-in

|  |  |  |
| --- | --- | --- |
| **File operator** | **System administrator** | **BackYard** |
| For the following cases, the request will be returned automatically.  ・The check-in file name is different from the check-out file name  ・The content of check-in file is same as the check-out file  Check-in request  Request for file check in.  4.2.4 Check-in request  Check-out request  Request for file check out.  4.2.2 Check-out request | Automatic difference extraction function ※For the details of the function, please refer to 4.2.7 Automatic difference extraction function  Check-out requested  Checking out  Check in requested  Return (check-in request) | Check-in complete  Request  approval  Approve  Reject  -  Approve the check-in request automatically and reflect the file to Git  -  Approve the check-out request automatically. |

1. The flow of manual check-in

|  |  |  |
| --- | --- | --- |
| **File operator** | **System administrator** | **BackYard** |
| Automatic difference extraction function ※For the details of the function, please refer to 4.2.7 Automatic difference extraction function  For the following cases, the request will be returned automatically.  ・The check-in file name is different from the check-out file name  ・The content of check-in file is same as the check-out file  Return (check-in request)  Check-out request  Request for file check out  4.2.2 Check-out request | Checking in  Approve  Request  approval  Return (check-in request)  Reject  Check-in  Approve check-in request.  4.2.5 Check-in  Check out requested  Checking out | Approve  Reject  Consistency check  Check in requested  -  Reflect the file to git.  Check in complete  Check in requested  Check-in request  Request for file check in  4.2.4 Check-in request  -  Check the consistency of the check-in file.  -  Approve check-out request automatically. |

### **Cancellation workflow**

　　　Before check-in is done, it's possible to cancel the operation if the process status is in "Requested", "Checking out", "Checking in", etc. The cancelled target file will return to the status before "Check-out request"

|  |  |  |
| --- | --- | --- |
| **File operator** | **System administrator** | **BackYard** |
| Check out requested  Check in requested  Checking out  Check out requested (duplicate)  Checking in  Return (check-in request)  Return (check-out request)  Return (check-in request)  Checking in  Cancel  Cancel the operation of target file.  4.2.6 Cancel  Cancel | Return (check-out request)  Check out requested  Check out requested (duplicate)  Checking out  Check in requested  Cancel  Cancel  Cancel the operation of target file.  4.2.6 Cancel |  |

* **Legend of register screen item list**

The content of the register screen item list are described in the next section.

**②②**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **②** | **③** | **④** | **⑤** |
| **Item** | **Description** | **Input Required** | **Input type** | **Restrictions** |
|  |  |  |  |  |

1. **Item**

・The name of item in the submenu

1. **Description**

・The description of the item

1. **Input Required**

・○：Items that entering contents are required for them.

・‐ ：Items that entering contents are optional for them.

1. **Input type**

・Manual： Items that require manual input.

・Auto： Items whose content are entered automatically.

・Checkbox: Check box format item.

・Button: Radio button format item.

・List: List box format item.

1. **Restriction**

　・The restrictions for the item(Limitation on number of characters, etc.)

# Construction file management Function・operation method

This chapter explains each console function used in construction file management.

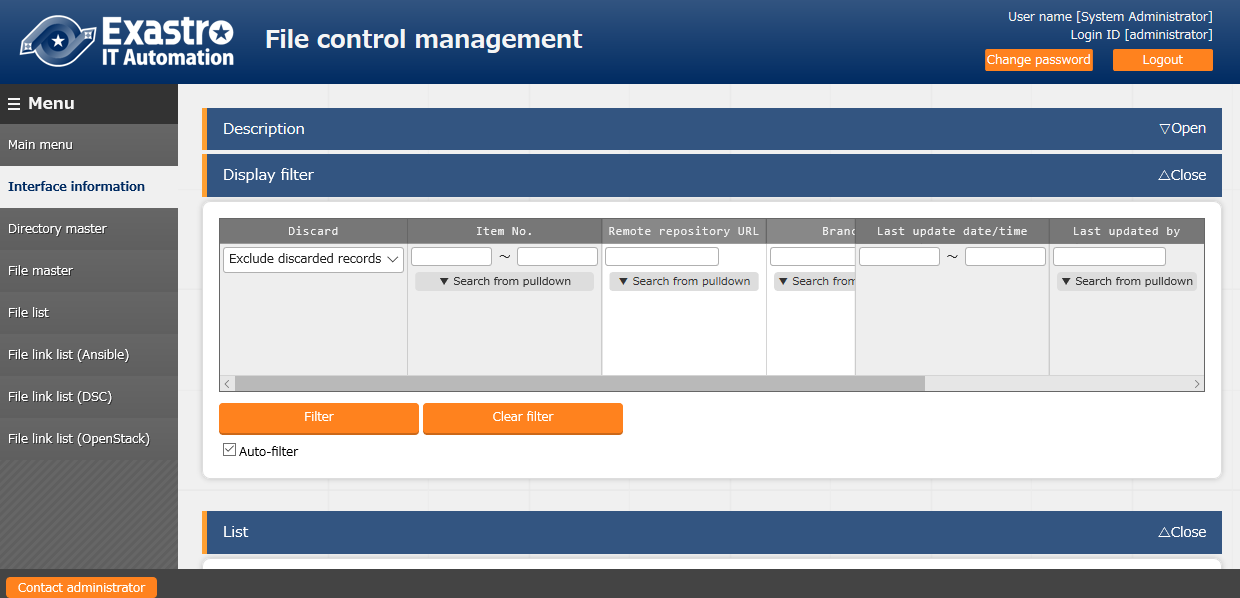


## File control master

This section describes the operation in file control master console

### **Interface information**

1. Initial synchronization of construction file management function and Git is performed in "interface information" submenu.



**Figure 4.1-1 submenu screen (Interface information)**

1. The item list of “interface information” screen is as below.  
   Please register each information and perform initial synchronization with Git.

**Please do not perform any other operations during the initial synchronization.**

※1 Please update the already registered empty data while initial registration.

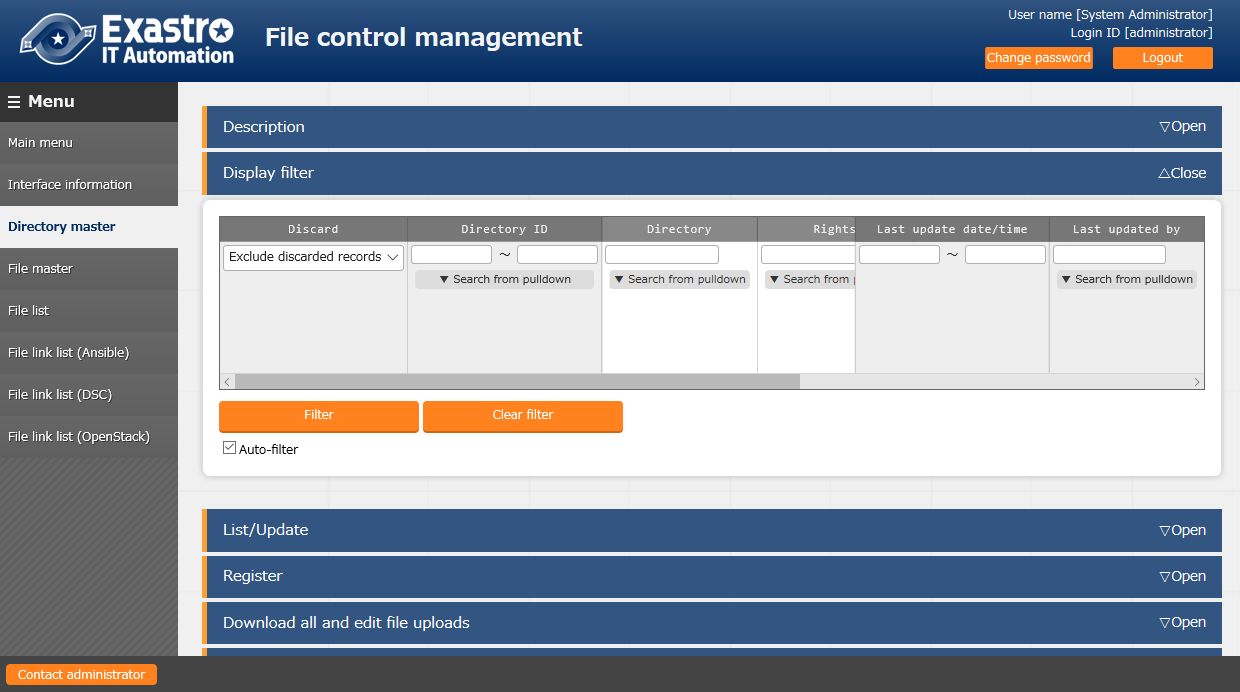
※2 Please set "remote repository URL" and "Clone repository" to the value of the argument which is passed to the clone command of Git as the following example command.

**Table 4.1-1 Register screen item list (Interface information)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Description | Input Required | Input  Type | Restrictions |
| Remote repository | Please enter the URL of an already created Git remote repository.  ※The following setting is required for the specified repository.  　・push.default=simple  　・core.bare=true  　・core.sharedrepository=1  (Example) ssh://root@192.168.1.1/dir1/dir2/main.git | ○ | Manual |  |
| Clone repository | Please enter the path of local directory that the user wants to create the clone repository.  If the entered directory does not exist, the directory will be created.  ※Since differential synchronization is not allowed, the repository in use can't be specified. In the case of specifying a exist directory, the directory is required to be empty. | ○ | Manual |  |
| Password | Please enter the password that is required when prompted according to the protocol specified in the "Remote repository URL".  If ssh password is needed on the remote repository side, please enter the password. | ○ | Manual |  |
| Initial synchronization | Press the button and start initial synchronization with Git. | Optional | Manual | Required for the first time only |
| Remarks | Free description field | Optional | Manual | Maximum length 4000bytes |

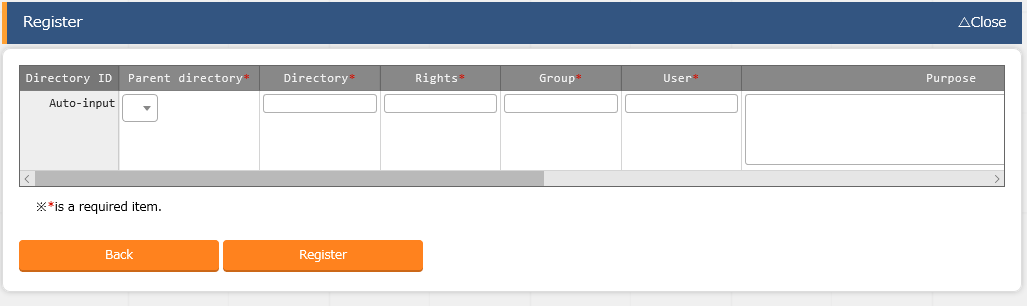
### **Directory master**

1. Created the directory for construction file management in "Directory master".



**Figure 4.1-2 Submenu screen (directory master)**

1. Click the "Register" - "Start Registration" button to register the operation information.



**Figure 4.1-3 Registration screen (directory master)**

1. The item list of "Directory master" screen is as below.

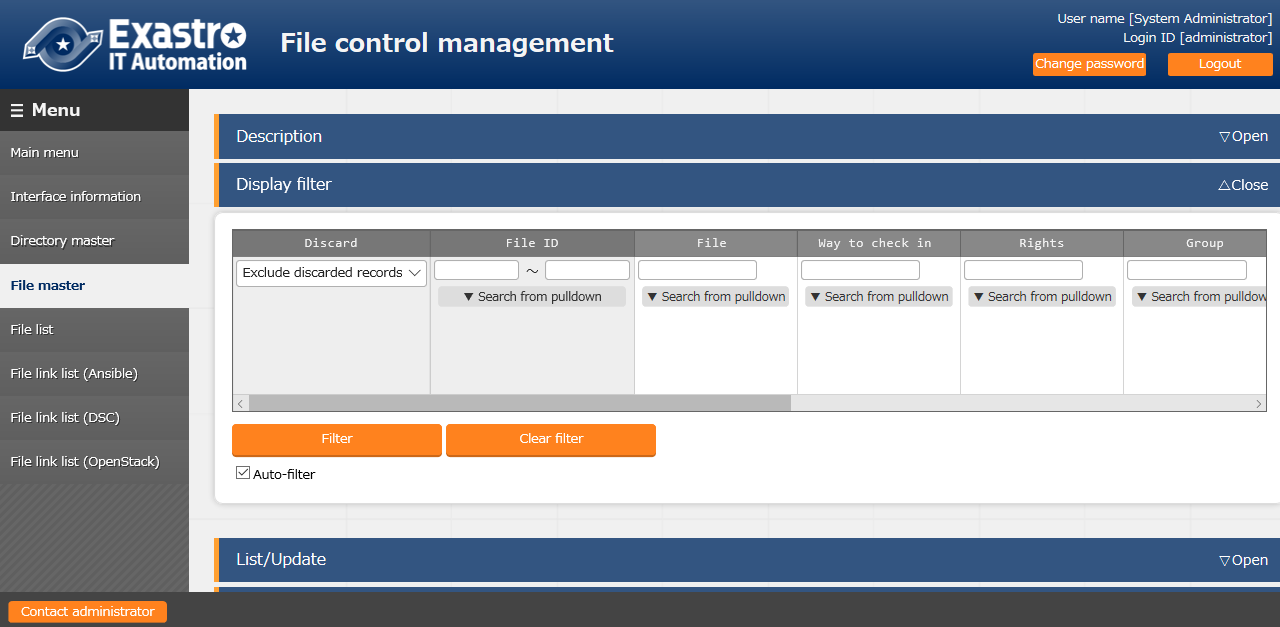
**Table 4.1-2 Item list of Registration screen (directory master)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Description | Input  Required | Input  Type | Restrictions |
| Parent  directory | Registered directory will be displayed.  To use the registered directory as parent directory, please select the target directory. | ○ | List |  |
| Directory | Please enter desired directory name.  ※1 Characters as「!”#$%&’|`;:\*<>?¥」,「.」,「..」can't be used in directory names  ※2 "/" can't be used at the beginning or end. | ○ | Manual |  |
| Rights | Please enter the permission of the creating directory with three digits from 0 to 7 | ○ | Manual |  |
| Group | Enter the group of the creating directory in half-width alphanumeric characters.  ※ If the entered group does not exist in the server, it will be managed as the "root" group. | ○ | Manual |  |
| User | Enter the owner of the creating directory in half-width alphanumeric characters.  ※ If the entered owner does not exist in the server, it will be managed as "root". | ○ | Manual |  |
| Purpose | Please enter the column if there is description of usage. | Optional | Manual |  |
| Remarks | Free description field | Optional | Manual | Maximum length 4000 bytes |

1. The specifications of update/discard are as below.
2. In the case of update/discard, if there is any file under the directory in check-in/check-out request, the update/discard can't be performed (validation error).
3. If the initial synchronization with Git has already done, updating/discarding the data in directory master menu that is also on Git, the data on Git will also be updated/deleted. In the case of register/restore, no operation will be performed on Git.
4. In the case of discard, the subdirectories in directory, the file in "File master", the file data in "Browse" and "File list" menu will all be discarded also.

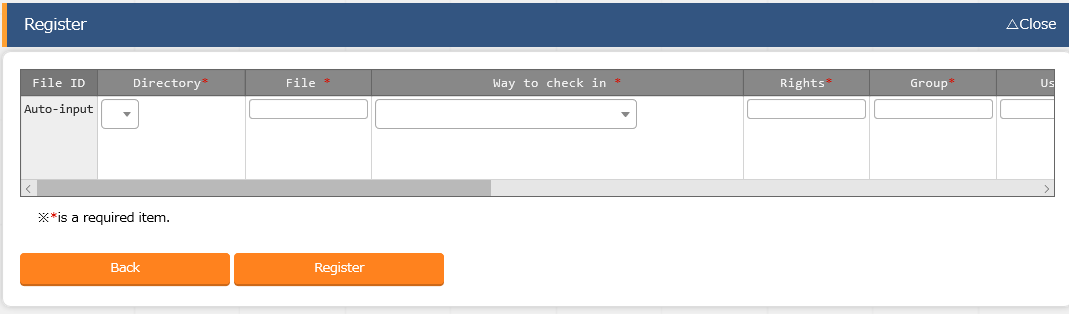
### **File master**

1. Register the manage target file in "File master".



**Figure 4.1-4 Submenu screen (File master)**

1. Click the "Register" - "Start Registration" button to register the operation information.



**Figure 4.1-5 Registration screen (File master)**

1. The item list of the file master screen is as below.

**Table 4.1-3 Register screen item list (File master)**

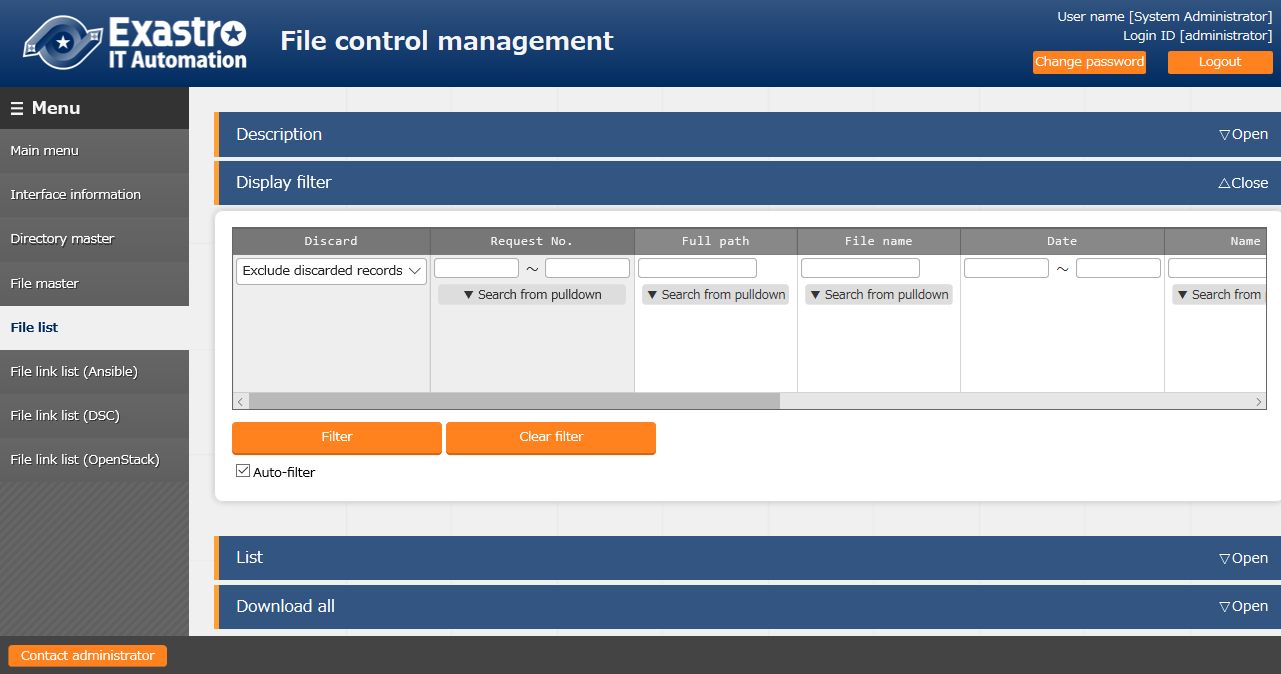
| Item | Description | Input required | Input type | Restrictions |
| --- | --- | --- | --- | --- |
| Directory | Please select the directory to manage. | ○ | List |  |
| File | Please enter the file name to manage.  ※1 Characters as 「!”#$%&’|`;:\*<>?¥」、 「.」、 「..」 can't be used in directory names. | ○ | Manual |  |
| Way to check in | Please select according to the operation needs.  ・Check in after administrator approval  Process the check-in request manually.  ・Auto-check in  Process the check-in request automatically. | ○ | List |  |
| Rights | Please enter the permission of the managing file with three digits from 0 to 7. | ○ | Manual |  |
| Group | Enter the group of the managing file in half-width alphanumeric characters.  ※ If the entered group does not exist in the server, the item will be managed as in the "root" group. | ○ | Manual |  |
| User | Enter the owner of the creating directory in half-width alphanumeric characters.  ※ If the entered owner does not exist in the server, the item will be managed as by "root" user. | ○ | Manual |  |
| Purpose | Please enter the column if there is description of usage. | Optional | Manual |  |
| Remarks | Free comment section. | Optional | Manual | Maximum length 4000 bytes |

1. The specification of update/discard is as below
2. In the case of update/discard, if there is any file in check-in/check-out request, the update/discard can't be performed (validation error).
3. If the initial synchronization with Git has done, updating/discarding the data on directory master that is also on Git, the data on Git will also be update/deleted. In the case of register/restore, no operation will be performed on Git.
4. In the case of discard, the file data in "Browse" and "File list" menu will all be discarded also.

### **File list**

1. The whole file managed in construction file management function is displayed in "file list".

Also, the files displayed in filter results can all be downloaded at once.



**Figure 4.1-6 Registration screen (file list)**

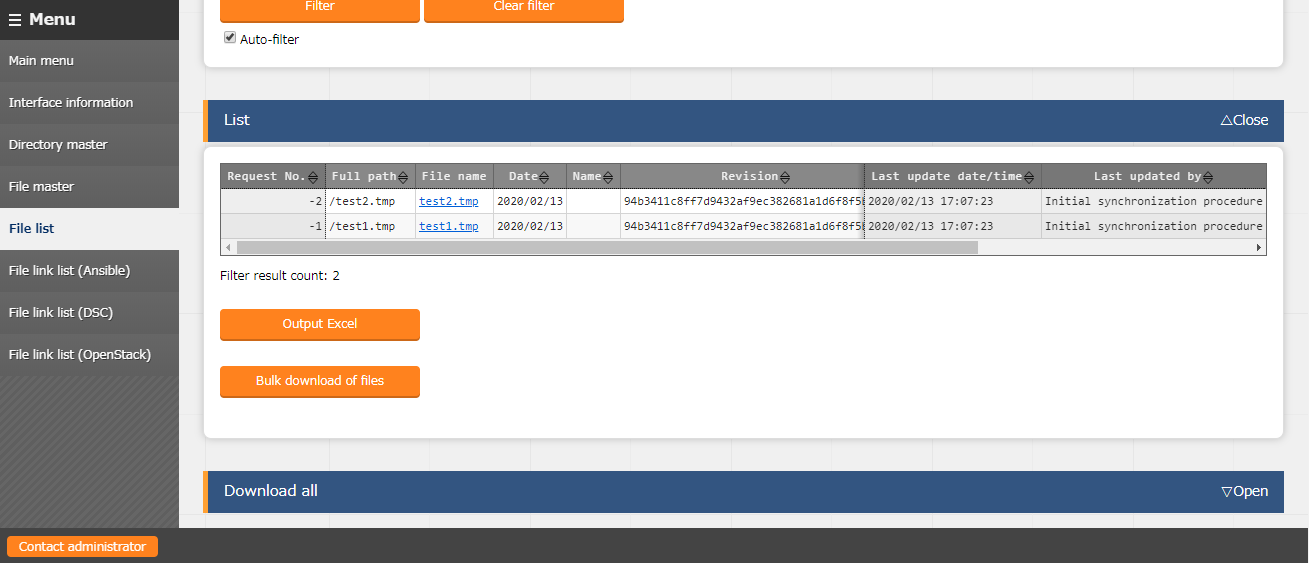
* 1. On opening the menu, the "latest flag" will be checked automatically and all newest file will be displayed with filter

Please press the "Download all" button on the bottom of the screen to download all files.

The file is downloaded in zip format.

※1 The “newest” flag is attached to the newest file among the files that have completed check-in

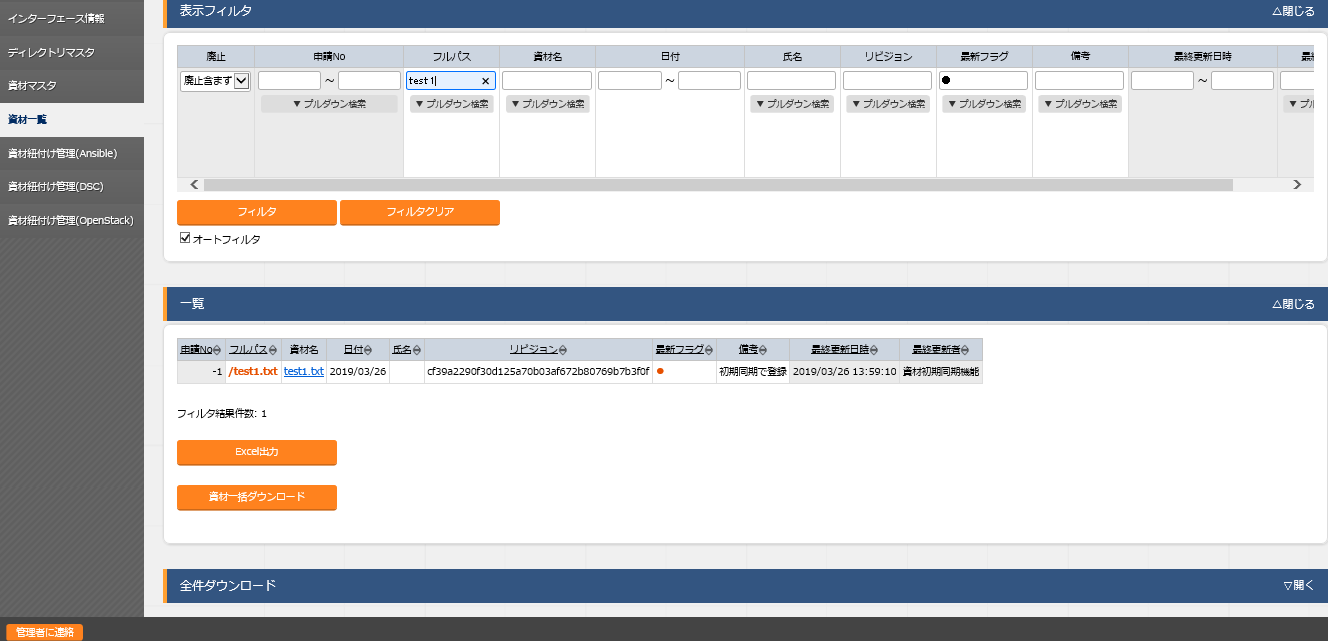
※2 The name of the zip file is "material\_YYYYMMDDHHMMSS.zip"





**Figure 4.1-7 "Bulk download" screen ① (File list)**

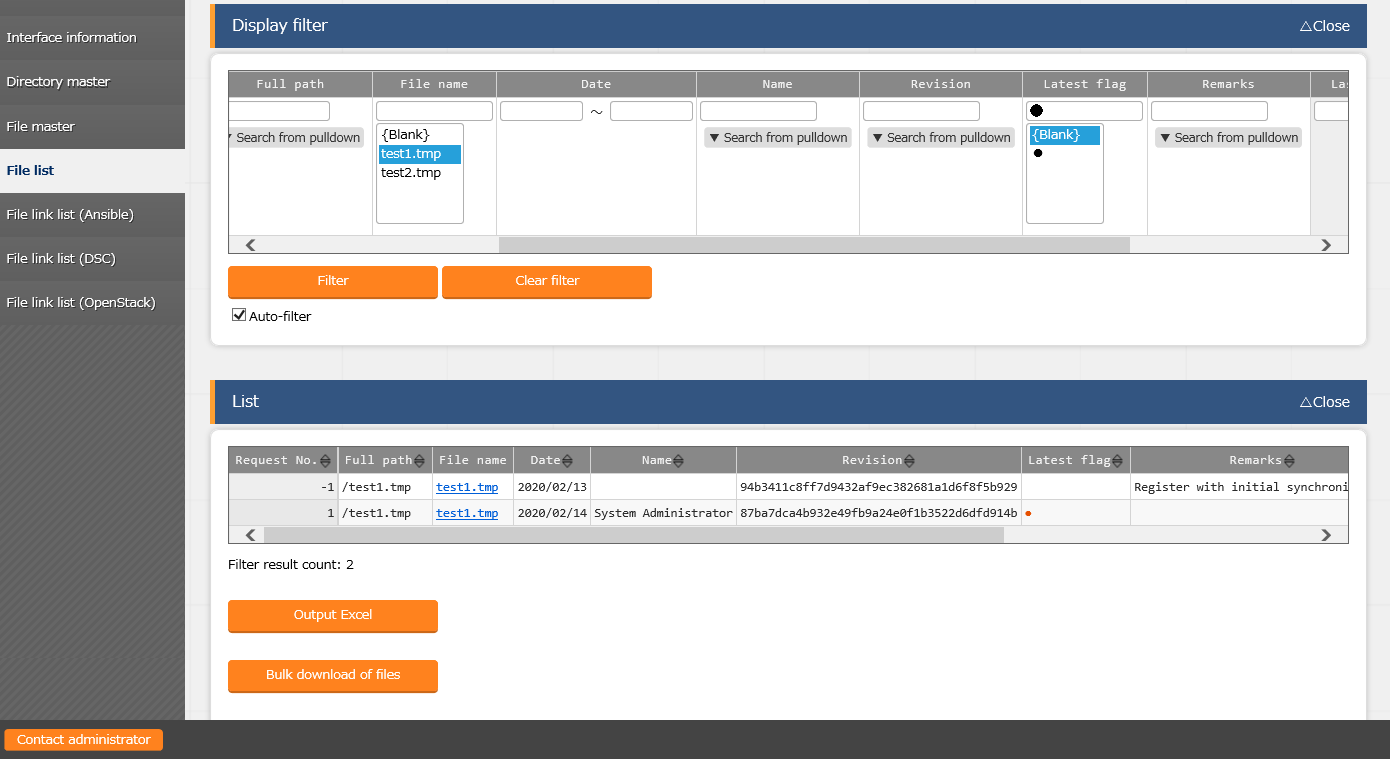
* 1. Filter by any information and download only necessary files at once is possible.





**Figure 4.1-8 "Bulk download" screen ② (File list)**

1. If the same material is displayed multiple times in the result of the filtered display, only the latest data of the file is downloaded.





**Figure 4.1-9 "Bulk download" screen ③ (File list)**

### **File link list (Ansible)**

1. Register the link between the file registered in construction file management and the file used in Ansible-driver console in "File link list (Ansible)" menu.

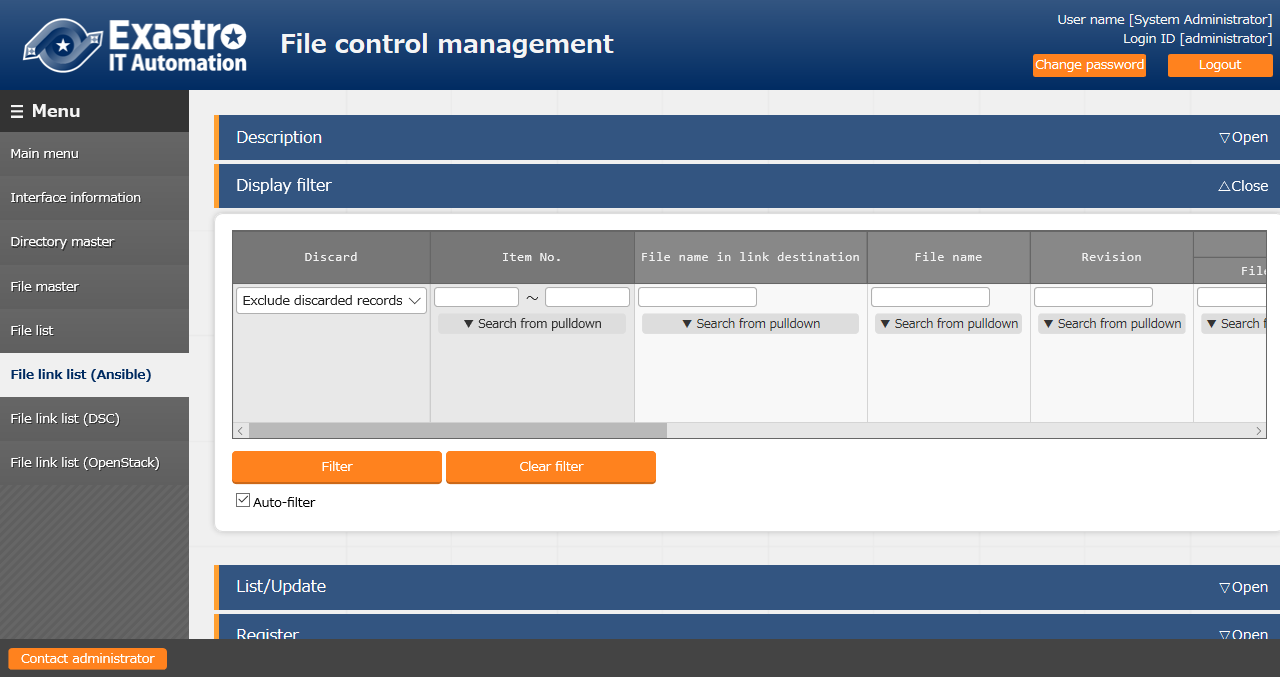
The link destination menus are as below.

* Ansible Common contents list
* Ansible Common template list
* Ansible-Legacy Playbook files
* Ansible-Pioneer Dialog files
* Ansible-LegacyRole Role package list

The registered file will automatically reflected each time when construction file management is updated.

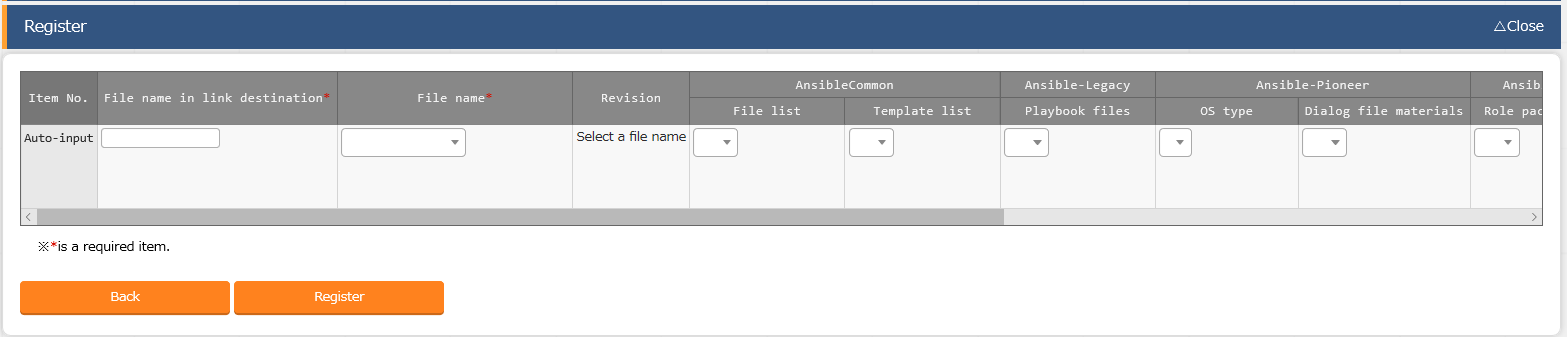
To use this function, Ansible-driver is required to be installed in ITA.

For details of this function, please refer to chapter 5.



**Figure 4.1-10 Submenu screen (File link list (Ansible))**

1. Click the "Register" - "Start Registration" button to register the operation information.



**Figure 4.1-11 Registration screen (File link list (Ansible))**

1. The item list of "File link list (Ansible)" screen is as below.

**Table 4.1-4 Register screen item list(File link list(Ansible))**

| Item | | Description | Input  Required | Input  Type | Restriction |
| --- | --- | --- | --- | --- | --- |
| File name in link destination | | Enter the file name registered in the link destination.  If the file name entered does not exist in the link destination, it will be newly registered. | ○ | Manual |  |
| File name | | Please select the target file to link with. | ○ | List |  |
| Revision | | Select the revision number you want to specify. If no selection is made, the latest revision is automatically specified.  ※ If a revision is specified, automatic update will not be performed. | Optional | List |  |
| Ansible Common | Template list | Select “●” when linking to the "Template list" menu.  ※ Please Enter "TPF\_" at the beginning and enter alphanumeric characters in the following in "File name in link destination" column when selecting this option." | ※1 | List | ※1  Please select only one link destination.  ※2  Ansible-Pioneer Required when linking to Ansible Pioneer "Dialog file materials" menu. |
| File list | Select “●” when linking to the "File list" menu.  ※ Enter "CPF\_" at the beginning and enter alphanumeric characters in the following in "File name in link destination" column when selecting this option. | ※1 | List |
| Ansible-Legacy | Playbook files | Select “●” when linking to the "Playbook file" menu. | ※1 | List |
| Ansible-Pionner | OS type | Please select the OS type of link destination file when linking to the "Dialog file materials" menu. | ※2 | List |
| Dialog file materials | Select “●” when linking to the "Dialog file materials" menu. | ※1 | List |
| Ansible- Legacy Role | Ansible-Role Role package file | Select “●” when linking to the "Role package file" menu.  ※If the linked file is not the specified zip file, the registration data will be discarded and an error message will be displayed in the remarks column. | ※1 | List |
| Remarks | | Free description field | Optional | Manual | Maximum length 4000 bytes |

2. 2. 1. **File link list（OpenStack）**
3. Register the link between the file registered in construction file management and the file used in OpenStack-driver console in "File link list (OpenStack)" menu.

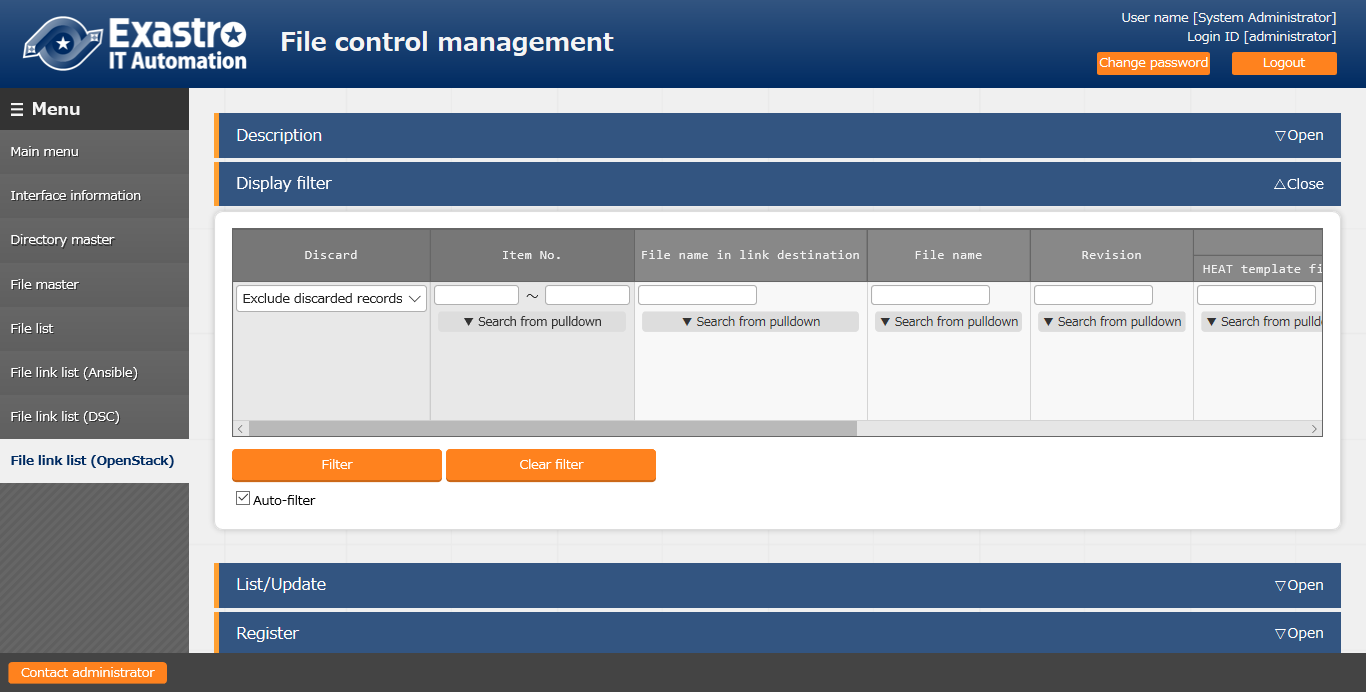
The link destination menu is as below

* Movement list

The registered file will automatically reflected when construction file management is updated.

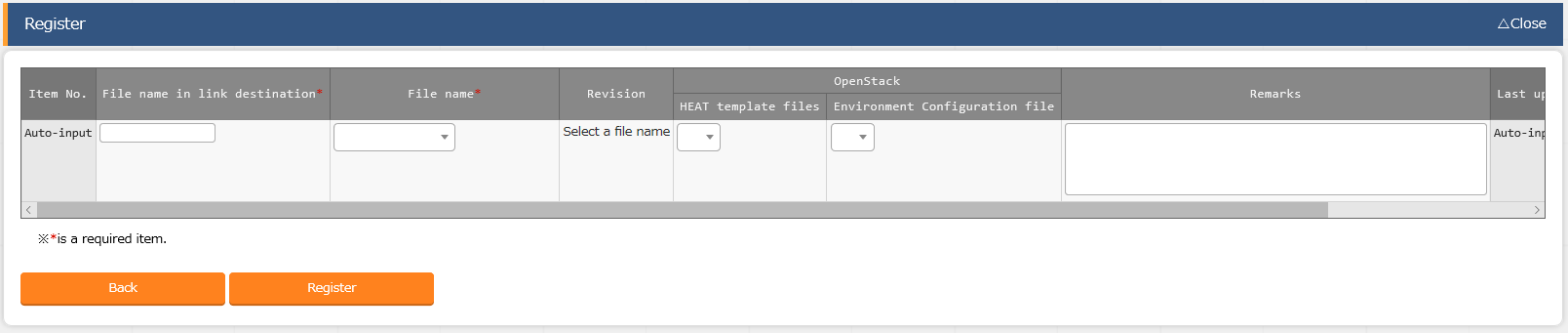
To use this function, OpenStack-driver is required to be installed in ITA.

For details of this function, please refer to chapter 5.



**Figure 4.1-12 Submenu screen (File link list (OpenStack))**

1. Click the “Register” – “Start Registration” button to register the operation information.



**Figure 4.1-13 Registration screen (File link list (OpenStack))**

1. The item list of “File link list (Openstack)” screen is as below.

**Table 4.1-5 Register screen item list (File link list (OpenStack))**

| Item | | Description | Input required | Input type | Restriction |
| --- | --- | --- | --- | --- | --- |
| File name in link destination | | Enter the file name registered in the link destination.  If the file name entered does not exist in the link destination, it will be newly registered. | ○ | Manual |  |
| File name | | Please select the target file to link with. | ○ | List |  |
| Revision | | Select the revision number you want to specify. If no selection is made, the latest revision is automatically specified.  ※If a revision is specified, automatic updating will not be performed. | Optional | List |  |
| OpenStack | HEAT template files | Select “●” when linking to the "HEAT template files" menu. | ※ | List | ※  Please select only one link destination. |
| Environment Configuration file | Select “●” when linking to the "Environment Configuration file" menu. | ※ | List |
| Remarks | | Free comment section | Optional | Manual | Maximum length 4000 bytes |

## File control check-in/check-out console

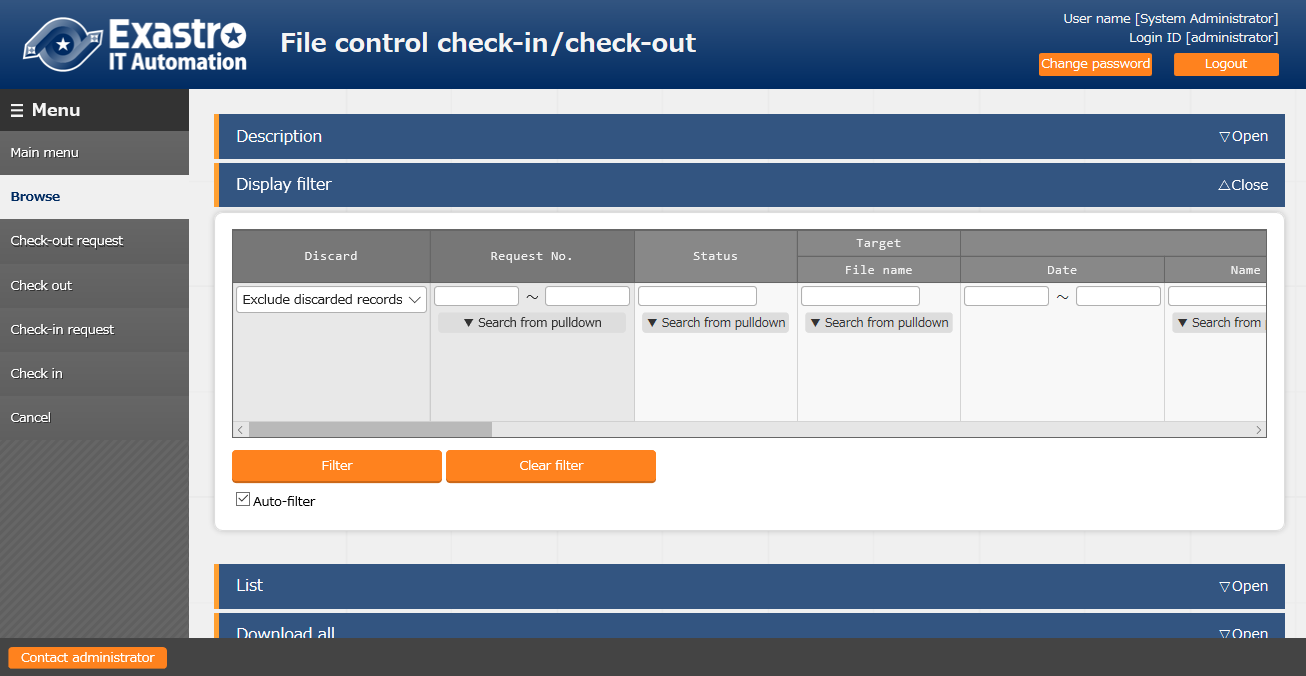
This section describes the operations in file control check-in/check-out console



### **Browse**



1. The reference of the check-in/check-out data of the whole file processed in construction file management file is displayed in "browse". Since this menu is for reference display, data update can't be performed in this menu.



**Figure 4.21 submenu screen (browse)**

1. The management status is displayed in "list".

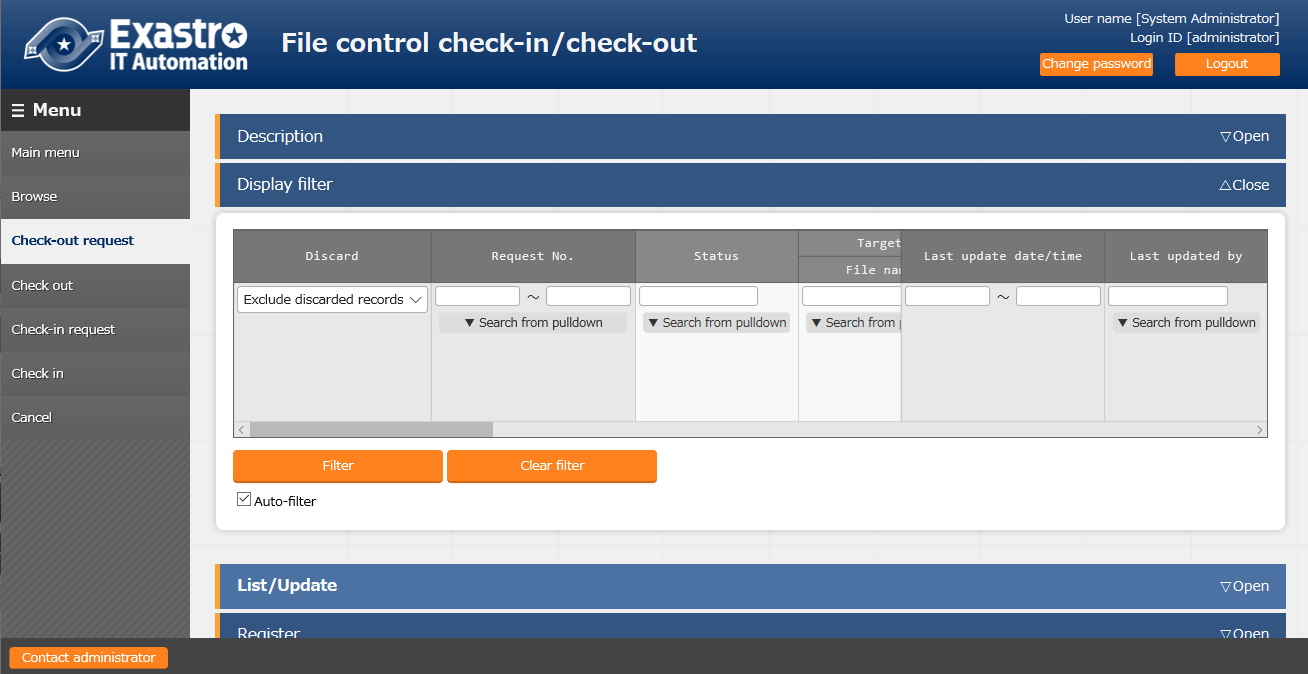
Information such as file status, the user who checked-out the file and the scheduled check-in date of revision or checked-out file are displayed.

Please refer to 2.2 "Construction file management status list" for details of status.



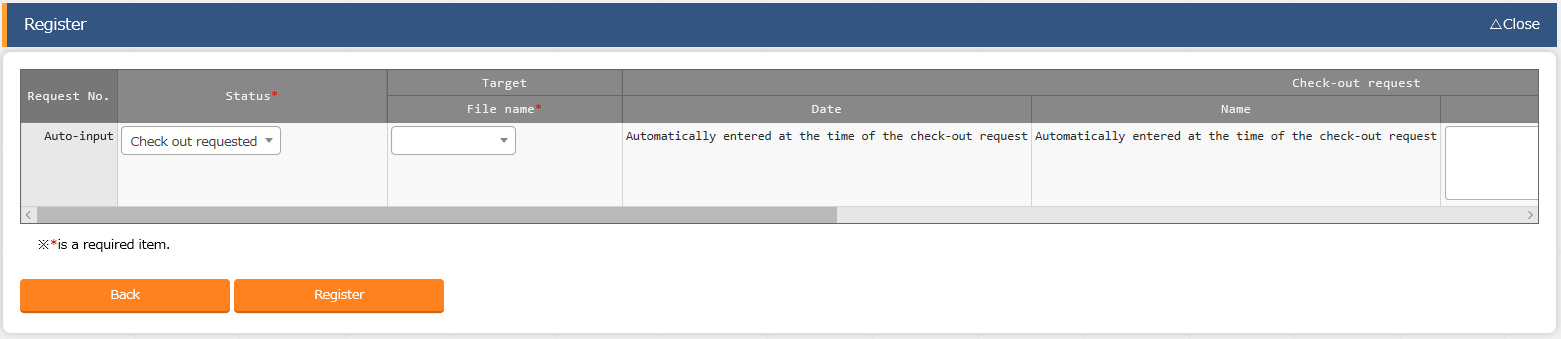
### **Check-out request**

1. The Check-out request is performed in "Check-out request" menu.



**Figure 4.2-2 Submenu screen (Check-out request)**

1. Click the "Register" - "Start Registration" button to register the operation information.



**Figure 4.2-3 Registration screen (Check-out request)**

If the update content has been confirmed, please also enter the "change summary" and "Scheduled date for check-in".

※ The register status will become "Check out requested (duplicate)" if the request target file is already checked-out. The target file will check-out automatically after it is checked-in.

1. The item list of Check-out request screen is as below

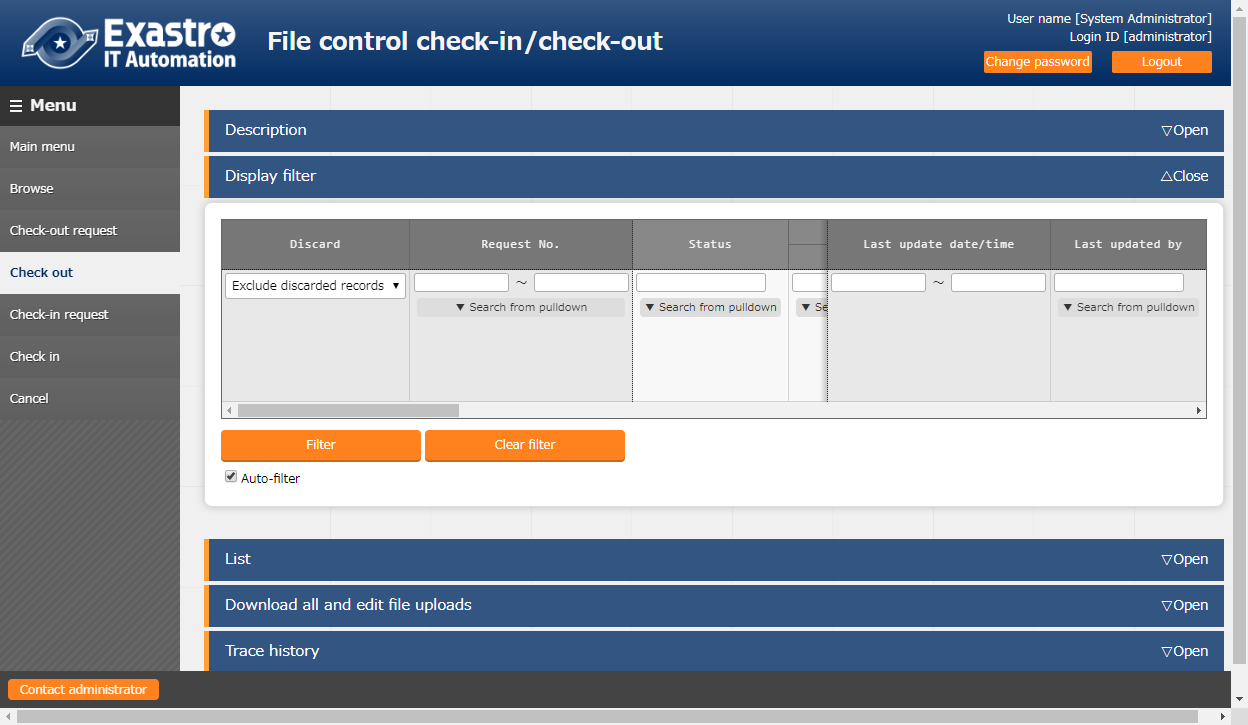
**Table 4.2-1 Register screen item list (Check-out request)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | | Description | Input required | Input type | Restriction |
| Status | | Select the processing status of the file. Please select "Check out requested" when registering. | ○ | List |  |
| Target | File name | Please select the check-out request target file | ○ | List |  |
| Check-out request | Date | Automatically entered when the "Register" button is pressed. | － | Auto |  |
| Name | Automatically entered when the "Register" button is pressed. | － | Auto |  |
| Change summary | Please the summary of change | Optional | Manual |  |
| Scheduled date for check-in | Please enter the scheduled check-in date | Optional | Manual |  |
| Remarks | | Free description field | Optional | Manual | Maximum length 4000 bytes |



### **Check-out**

1. The data in check-out request is displayed in "Check-out" menu.



**Figure 4.2-4 submenu screen (check-out)**

The data registered in the check-out request is automatically checked-out, so no operation is required in this menu.

* It cost maximum 10 seconds for the status to become "Checking out".



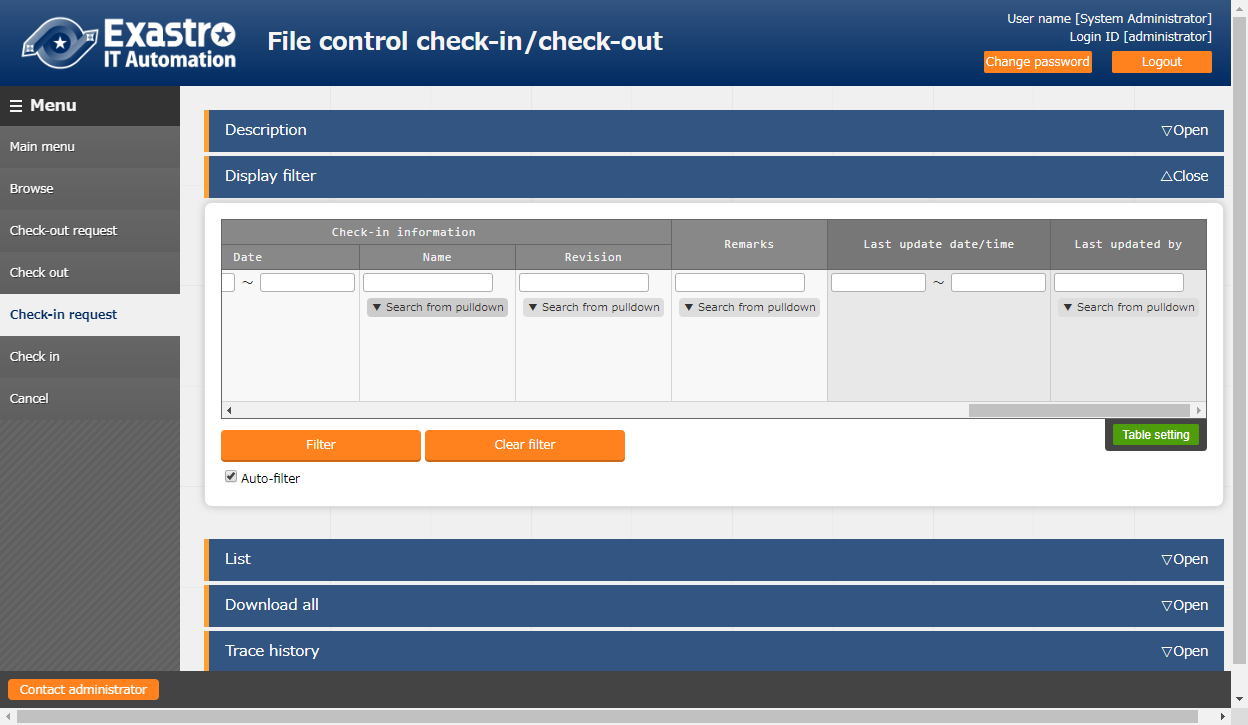
### **Check-in request**

1. The data in checking-out status will be displayed in "Check-in request" menu.

The file operator please update the displayed data and request for check-in.

※If the DIFF file is not registered, the DIFF file will be generated automatically.

For the automatic generation of DIFF file, please refer to 4.2.7Automatic difference extraction function.



**Figure 4.2-5 submenu screen (Check-in request)**

1. The item list of the Check-in request screen is as below

**Table 4.2-2 Register screen item list (Check-in request)**

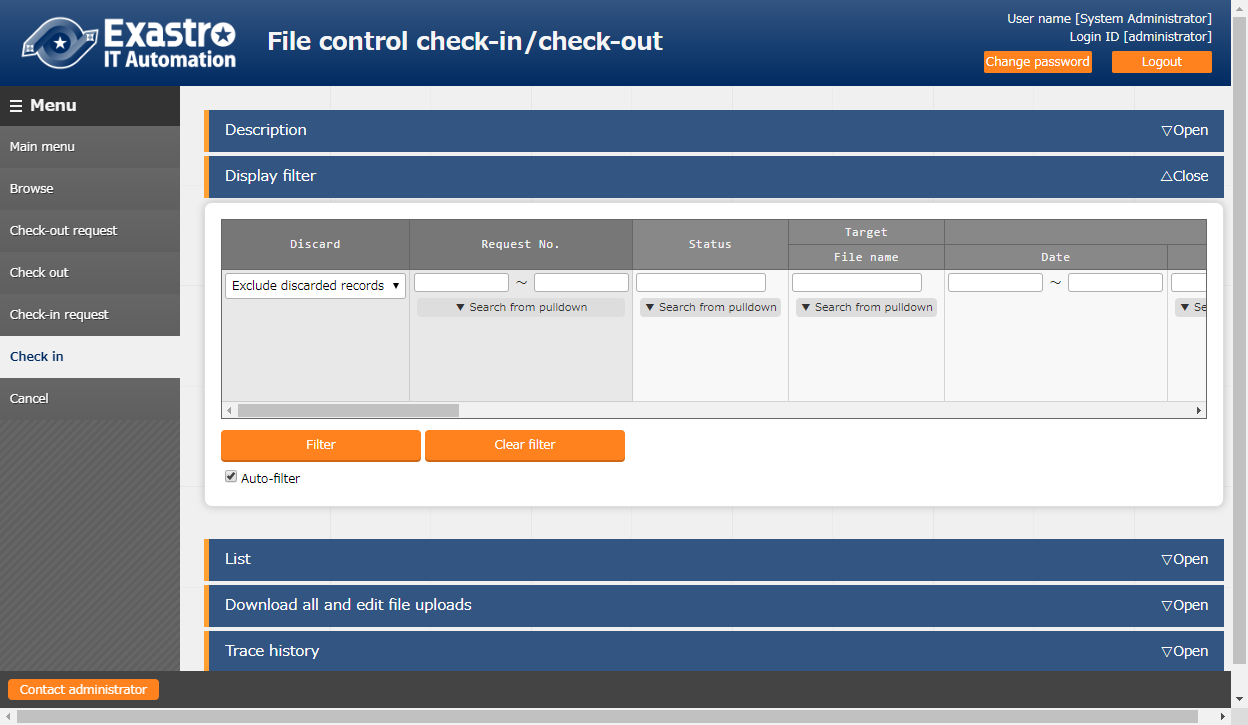
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | | Description | Input required | Input type | Restriction |
| Status | | Select the processing status of the file. Please select "Check in requested" when registering. | ○ | List |  |
| Target | File name | The file in "checking out" status will be displayed. | － | － |  |
| Check-in request | Date | Automatically entered when the "Register" button is pressed. | － | Auto |  |
| Name | Automatically entered when the "Register" button is pressed. | － | Auto |  |
| File | Please upload the file to be checked-in. | ○ | Manual |  |
| DIFF(txt) | Please upload if there is a DIFF file for the target file. | Optional | Manual |  |
| Test item list (xlsx) | Please upload if there is a test item file for the target file. | Optional | Manual |  |
| Evidence (zip) | Please upload if there are other evidence files. | Optional | Manual |  |
| Remarks | | Free description field. | Optional | Manual | Maximum size 4000 bytes |

### **Check-in**

1. The data in checking-in request will be displayed in "Check-in" menu

Administrator please update the displayed data and approve / reject check-in of the request target file.

※If "automatic check-in" is selected in the file master menu, operation are not required in this menu.



**Figure 4.2-6 submenu screen (check-in)**

1. The item list of check-in screen is as below.

**Table 4.2-3 screen item list (check-in)**

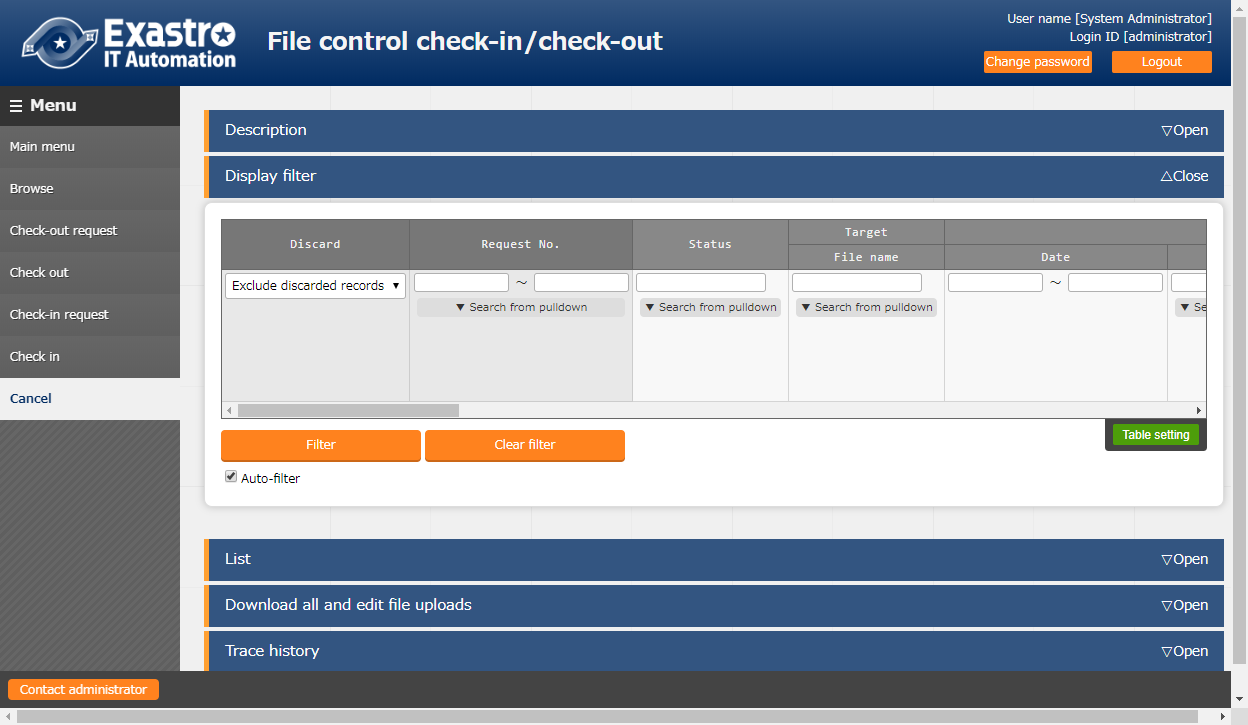
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | | Description | Input required | Input type | Restriction |
| Status | | Select the process status of file.  Select "Checking in" for approving the request. Select "Return (check-in request)" for rejecting the request. | ○ | List |  |
| Target | File | The file in "check-in request" status will be displayed | － | － |  |
| Check-in information | Date | Automatically entered when the "Register" button is pressed. | － | Auto |  |
| Name | Automatically entered when the "Register" button is pressed. | － | Auto |  |
| Revision | Please enter the revision number in the case of manual version control.  ※Even if revision number is entered, it will be overwritten and updated automatically when linked with Git | ○ | Manual |  |
| Remarks | | Free description field | Optional | Manual | Maximum length 4000 bytes |

### **Cancel**

1. The file data in the following status will be displayed in the "cancel" menu. In the case of canceling registered data, please update the target data and register for cancellation.

※The following 7 status can be cancelled

* Check out requested
* Check out requested (duplicate)
* Checking out
* Check in requested
* Checking in
* Return (check-out request)
* Return (check-in request)



**Figure 4.2-7 submenu screen (cancel)**

1. The item list of cancel screen is as below.

**Table 4.2-4 screen item list (cancel)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | | Description | Input required | Input type | Restriction |
| Status | | Select the processing status of file. Please select "cancel" when requesting. | ○ | List |  |
| Target | File name | The target file is displayed. | － | Auto |  |
| Remarks | | Free description field. | Optional | Manual | Maximum length 4000 bytes |

### **Automatic difference extraction function**

The following is the description of the DIFF file that is automatically generated during the check-in request.

* + DIFF file automatic generation

In the following cases, the DIFF file will be generated automatically

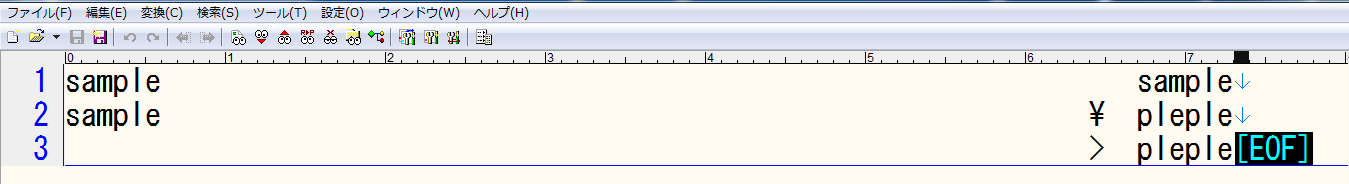
・DIFF file was not uploaded during the check-in request

・The target file type is "text" type.

※It takes maximum 10 seconds for the DIFF file to be generated automatically.

※If the character encoding of the file is other than UTF-8, the DIFF file may not be able to generate automatically.

* + How to read DIFF file (example)



File before update

File after update

\ ：Changed line

> ：Added line

**Figure 4.2-8 DIFF file sample**

# Material Linkage function

The following is the description of the Material Linkage function.



## What is Material Linkage function

Material linkage function is a function that automatically links the file managed in construction file management function with file upload items in various menus of ITA.

## Image of Material Linkage function

The following figure takes registering link to the Ansible driver - playbook list console as an example.

|  |  |
| --- | --- |
| File operator | BackYard |
| If revision is specified during the ②registration, the data will not be updated.  ④File check-in/check-out  Check-in/Check-out service.yml and update the file content  ②Register file linkage  Register the following information  "File link list" menu  Link destination file name: start\_service  File name ：service.yml  Playbook files ：● | ⑤Playbook files  Auto update according to the linkage information in ①  “Playbook files”　Menu  Playbook file name：start\_service  Playbook file ：service.yml  If the information of "Link destination file name" registered in ②  ・exists in link destination→the existing data will be updated  ・not exist in link destination→ new file will be registered  ①Git association  Initial synchronization with Git or register the file that needs check-in / check-out to the file list menu  ③Link with the link destination  Register the following information  “Playbook files”　Menu  Playbook file name：start\_service  Playbook file ：service.yml |

# Application operation

The operation to utilizing this function is not only inputs by user usage from the browser screen of client PC but also operations according to system operation and maintenance. The available operation and maintenance are as follows.

* Maintenance
* Change log level



## Maintenance

The required files for construction file management function process to start/stop/restart

are as below

|  |  |
| --- | --- |
| **Description** | **Target file name** |
| Automatic check-in / check-out function.  Check-in / check-out file and associate with Git according to the data registered in file management function. | ky\_material\_management.service |
| Material linkage function (Ansible)  Automatically associate the file managed in file management menu with Ansible-driver menu | ky\_material\_linkage\_ansible.service |
| Material linkage function (DSC)  Automatically associate the file managed in file management menu with DSC-driver menu. | ky\_material\_linkage\_dsc.service |
| Material linkage function (OpenStack)  Automatically associate the file managed in file management menu with OpenStack-driver menu | ky\_material\_linkage\_openstack.service |

The target file is stored in “/usr/lib/systemd/system”

The method to start/stop/restart the process are as follows.

Please execute the commands with root privilege.

1. Start process

# systemctl start ky\_material\_management.service 

1. Stop process

# systemctl stop ky\_material\_management.service 

1. Restart process

# systemctl restart ky\_material\_management.service 

Please substitute the target file name to start / stop / restart the process.

## Change log level

1. Change to NORMAL level

Rewrite the 8th line of the following file from “DEBUG” to “NORMAL”.  
Log level setting file： <insallation direcotory>/ita-root/confs/backyardconfs/ita\_env

1. Change to DEBUG level
2. Rewrite the 8th line of the following file from “NORMAL” to “DEBUG”.  
   Log level setting file： <installation direcotory>/ita-root/confs/backyardconfs/ita\_env

After rewriting the file, **the change takes effect after restarting the process.**

Please refer to "6.1 Maintenance" for restart.

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