



ITA_User_Instruction_Manual

CI/CD For IaC

—Version 1.9 —

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 manual are subject to change without notice.

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, and Tomcat are registered trademarks or trademarks of the Apache Software Foundation.
- Ansible is a registered trademark or trademark of Red hat Inc.
- Terraform is a registered trademark or trademark of HashiCorp.

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.

Terminology and definitions

This document uses the following terminology.

Terminology	Definition
ITA	Abbreviation for “Exastro IT Automation”
Link source file	Git repository source files that are linked using the CI/CD for IaC function
Link destination file	Refers to the files uploaded from the following ITA drivers (Ansible-driver and Terraform-Driver). <ul style="list-style-type: none">• Ansible-Legacy/Playbook file collection• Ansible-Pioneer/Dialogue file collection• Ansible-LegacyRole/Role package list• Ansible common/File list• Ansible common/Template list• Terraform/Module files• Terraform/Policy list

Table of contents

Introduction	3
1. CI/CD For IaC overview	4
1.1 About CI/CD For IaC.....	4
1.2 Function overview figure	5
2. CI/CD For IaC Menu configuration	6
1.3 Menu/Screen.....	6
3. Using the CI/CD For IaC function	7
3.1 Workflow	7
4. CI/CD For IaC function menu.....	8
4.1 CI/CD For IaC standard menu.....	8
4.1.1. Remote repository	8
4.1.2 Register account	13
4.1.3 File link	14
4.2 CI/CD For IaC hidden menus	22
4.2.1. Interface information.....	22
4.2.2. Remote repository files.....	24
5. Operating.....	25
5.1 About performing maintenance	25
5.2 Change log levels	26
6. Appendix	27
6.1 Points to note when registering files to Git repositories	27
6.2 Points to note when registering file that are linked in the role package list to Git repositories.	27

Introduction

This document explains the CI/CD For IaC function and how to use it.

1. CI/CD For IaC overview

1.1 About CI/CD For IaC

The CI/CD For IaC can be divided into two functions.

(1) Git link function

This function creates a clone of the Git repository within ITA.

It will also use the clone to check any updates that might have happened to the source files which will be listed in ITA's "Remote repository file" menu.

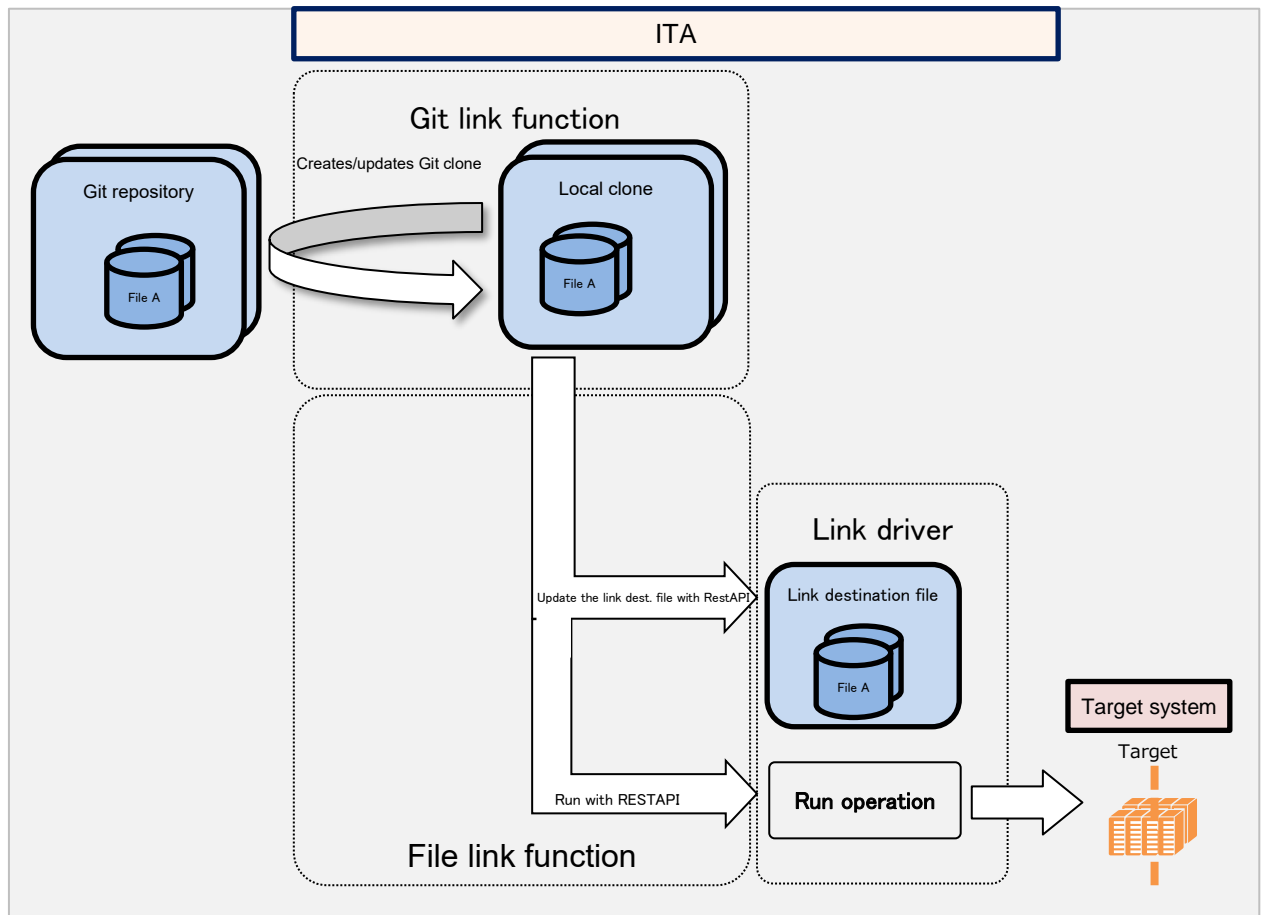
(2) File link function

Links the link source file and the link destination file and then register an Operation and Movement in order to verify the link source file.

If the link source file gets updated, the link destination file will be automatically updated. An Operation and Movement will be run to check the files.

1.2 Function overview figure

The is an overview figure of the CI/CD For IaC function.



2. CI/CD For IaC Menu configuration

This section will explain the menu configuration of the CI/CD For IaC function menu.

For information regarding basic controls, such as logging in to the system from the web, the menu components, please see the XXX manual

1.3 Menu/Screen

The following table lists the different menus and screen for the CI/CD For IaC function.

Table 2.1-1 CI/CD For IaC function menu list

No	Menu group	Menu/Screen	Chapter	Description
1	CI/CD For IaC	Interface information	4.2.1	Manages the interface information needed to access the link destination files with ITA's RestAPI
2		Remote repository	4.1.1	Manages the Git repository information
3		Registered account	4.1.2	Manages the account information needed to access the link destination files with ITA's RestAPI
4		Remote repository files	4.2.2	Manages the Git repository file information
5		File link	4.1.3	Manages the link information between the link source files and the link destination files.

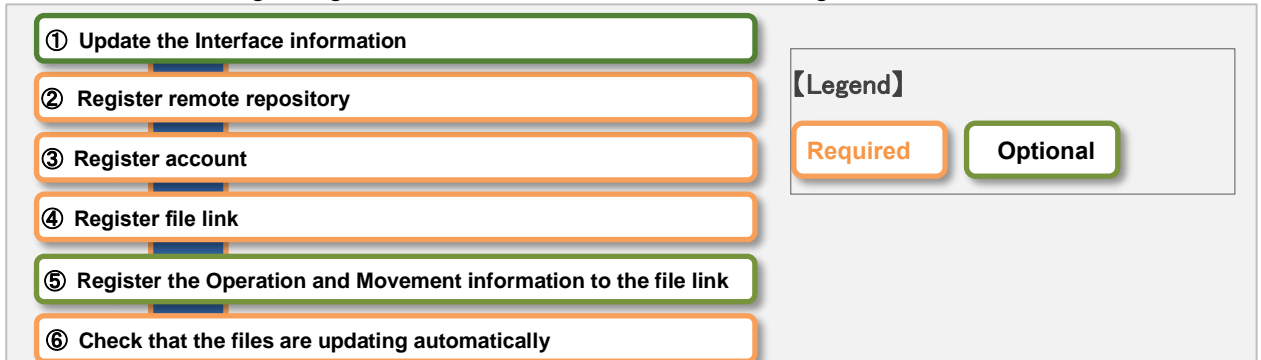
3. Using the CI/CD For IaC function

This section explains the process of using the CI/CD For IaC function.

3.1 Workflow

The basic workflow for the CI/CD For IaC is as follows.

More information regarding each of the tasks is listed below the figure.



Workflow details and references

- ① Update the Interface information
You will need to register interface information if you want to access the link destination file from ITA's RestAPI. Interface information needed to access the install destination ITA is installed by default. If you want the link destination files to be stored a place other than the install destination ITA, please change the interface information.
Please see the "4.2.1 Interface Information" menu for more details.
- ② Register Remote repository
Register the information of the Git you want to link to.
Please see the "4.1.1 Remote Repository" menu for more details.
- ③ Register account
Register the account information needed to access the link destination files from ITA's RestAPI.
Please see the "4.1.2 Registered Account" menu for more details.
- ④ Register file link
Register the link between the link source files and the link destination files
Please see the "4.1.3 File link" menu for more details.
- ⑤ Register the Operation and Movement information to the file link
Register Operation and Movement if you want to check that the files are updating automatically.
Please see the "4.1.3 File link" menu for more details.
- ⑥ Check that the files are updating automatically
Check that the link destination files updates automatically when the link source files have changed. This includes checking that the operation and movement is running automatically if they are registered
Please see the "3.4.1 File link" menu for more information

4. CI/CD For IaC function menu

This section explains the CI/CD For IaC menu and how to navigate it.

4.1 CI/CD For IaC standard menu

The following will explain the standard CI/CD for IaC menu, meaning that it will show the functions as it is after being installed.

4.1.1. Remote repository

- (1) In the “Remote repository” menu, users can register the information of the Git repository they want to link with.

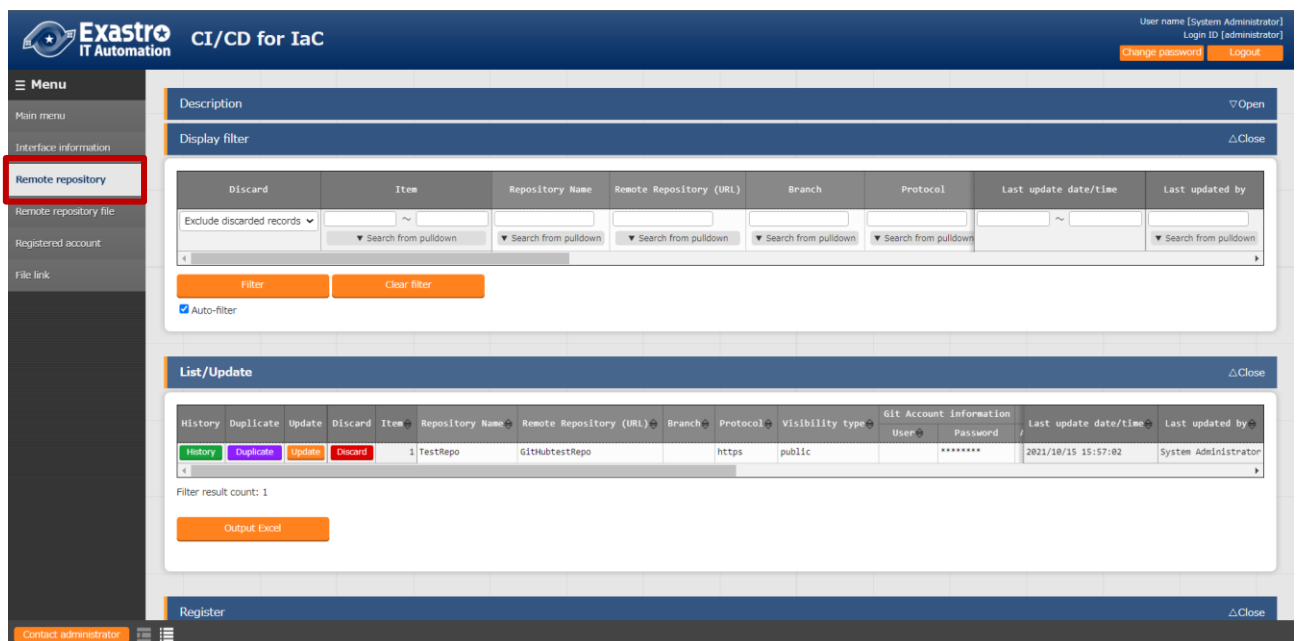


Figure 4.1-1 Menu screen (Remote repository)

- (2) The Remote repository menu items are as following.

Table 4.1-1 Remote repository input item list (Remote repository)

Item	Description	Input necessary	Input format	Restrictions
Repository name	Input the name of the repository. The text input in this field will be the display name for the remote repository in the CI/CD For IaC menus.	○	Manual	Max length256 bytes
Remote repository (URL)	Please input the Git repository URL specified for the Git clone command.	○	Manual	Max length256 bytes
Branch	Please input the Git repository branch name specified for the Git clone command. If nothing is input, the default branch will be specified.	—	Manual	Max length256 bytes

Protocol		<p>Please select the protocol used when connecting to the Git repository.</p> <p>Select the protocol used when connecting to the Git repository.</p> <p>●https Select if you are connecting to a remote Git repository with https.</p> <p>●ssh password authentication Select if you are connecting to a remote Git repository with an ssh password.</p> <p>●ssh key authentication (Without passphrase) Select if you are connecting to a remote Git repository with an ssh key</p> <p>●ssh key authentication (With passphrase) Select if you are connecting to a Remote Git repository with an ssh passphrase key authentication.</p> <p>●Local Select if you are connecting to a local Git.</p>	○	List selection	
Visibility type		<p>Please select the Visibility type for the Git repository (Public/Private).</p> <p>If the connection protocol is set to "https", "Visibility type" is an input necessary item.</p>	—	List selection	
Git account information	User	<p>Input the Git user name.</p> <p>If the Visibility type is set to "Private", this field is an input necessary item.</p>	—	Manual	Max length128 bytes
	Password	<p>Input the password needed when running the Git clone command.</p> <p>If the Visibility type is set to "Private", this field is an input necessary item.</p> <p>Note that GitHub stopped support for password authentication in August 2021. See this url for more information.</p> <p>If you are using GitHub with a password authentication, you will need to create a personal access token. See here for more information.</p>	—	Manual	Max length128 bytes
SSH Connection information	Password	<p>Input the password for the Linux user that will run the Git clone command.</p> <p>You must input a password if "ssh password authentication" is set as protocol.</p>	—	Manual	Max length128 bytes
	Passphrase	<p>Input the passphrase for the key file used when running the Git clone command.</p> <p>You must input a passphrase if "SSH key authentication" is set as protocol.</p>	—	Manual	Max length128 bytes

	Connection Parameter	<p>Input the parameter configured to the GIT_SSH_COMMAND environment variable when running the Git clone command.</p> <p>The GIT_SSH_COMMAND is a environment variable that can be configured with Git version 2.3 or later.</p> <p>Any configured parameters will be deactivated if the ITA server is installed with a version of Git earlier than version 2.3.</p> <p>The environment variable GIT_SSH_COMMAND has the following parameters set by default.</p> <p>UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no</p> <p>If there is no core.sshCommand set to git config -global, set the following parameter. ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no</p> <p>If core.sshCommand is set to git config -global, include the following parameter. -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no</p>	—	Manual	Max length512 bytes
Proxy	Address	<p>Input the proxy server address.</p> <p>If ITA is under a proxy environment, you may need to configure it in order to communicate to the Git server.</p> <p>If the Proxy server URL is http://procy.gate.co.jp:8080 Input http://procy.gate.co.jp into the Address field.</p> <p>The port will then be 8080.</p>	—	Manual	Max length128 bytes
	Port	<p>Input the port for the proxy server.</p> <p>If ITA is under a proxy environment, you may need to configure it in order to communicate to the Git server.</p>	—	Manual	
Remote Repository Synchronization Information	Automatic synchronizati on	<p>Select if you want to synchronize with the Git repository automatically or not.</p> <p>Activate: Will synchronize the repository with the Git repository with the “Cycle(Seconds)” specifying the interval.</p> <p>Deactivate: Will not be automatically synchronized with the Git repository.</p>	○	List selection	Default: Activated
	Cycle(Seconds)	<p>Input the cycle time in which the repository will be synchronized with the Git repository.</p> <p>If nothing is input, the default value (60s) will be used.</p>	—	Manual	Unit: seconds
Conn ection	Retry number	<p>Input how many times the system will retry connecting to Git if it fails.</p>	—	Manual	

		If nothing is input, the default value (3 times) will be used.			
	Cycle(ms)	Input the interval time between when the system retries to connect to Git if it fails. If nothing is input, the default value (1000ms) will be used.	—	Manual	Unit:ms
Remarks		Free description field	—	Manual	Max length4000 bytes

(1) SSH Key authentication.

Create a key with the root user and swap the key.

秘密鍵ファイル(id_ras)のパスはデフォルト(/root/.ssh/id_ras)を使用して下さい。デフォルト以外にしたい場合、ssh 接続情報／接続パラメータに秘密鍵ファイル(id_ras)のパスを設定して下さい。

-i 秘密鍵ファイル(id_ras)のパス

(2) ITA サーバにリモートリポジトリを作成する場合の Git バージョンに関する注意事項

Git バージョンが 1.8.4.3 より古い場合、リモートリポジトリに新規ブランチを作成してからコミットするまでの間、リモートリポジトリの HEAD ブランチ(Default branch)の特定が出来なくなる為、この間にブランチ切り替えを行った場合にブランチ切り替えの検出できません。この事により切り替え先ブランチの資材が「資材紐付」メニューの資材パスに表示されない事があります。

ITA サーバにリモートリポジトリを作成する場合の Git バージョンは 1.8.4.3 以上を使用して下さい。

(3) The following items displays the Git repository synchronization status.

Table 4.1-2 Git repository synchronization status items (Remote repository)

Item	Description	Remarks
Status	The Git synchronization has 4 different statuses. Blank: When the record is newly registered, updated or restored after being abolished. Normal: When it has successfully synchronized with the Git repository. Abnormal: When an error occurred while synchronizing with the Git repository. Resumed: When the process has been resumed by clicking the "Resume" button. If the status displays "Abnormal", synchronization with the Git repository will stop. If you want to resume the process, either click the "Resume" button or update the record.	Please see" エラー! 参照元が見つかりません。 Points to note when registering files to Git repositories."
Detailed information	Displays the error information if the status changes to "Abnormal". If you want to clear the "Detailed information" item, either click the "Resume" button or update the record.	
Last date/time	Displays the date/time of the last synchronization. If you want to clear the "Detailed information" item, either click the "Resume" button or update the record.	

Item	Description	Remarks
Resume button	<p>This button will only be displayed if the synchronization status changes to "Abnormal".</p> <p>Press this button to resume the synchronization process.</p>	

4.1.2 Register account

- (1) The “Register account” menu is used for users to register the account information needed to access the link destination files from ITA’s RestAPI.
Account information registered to this information must also be registered in the “Manage console - User list” menu.

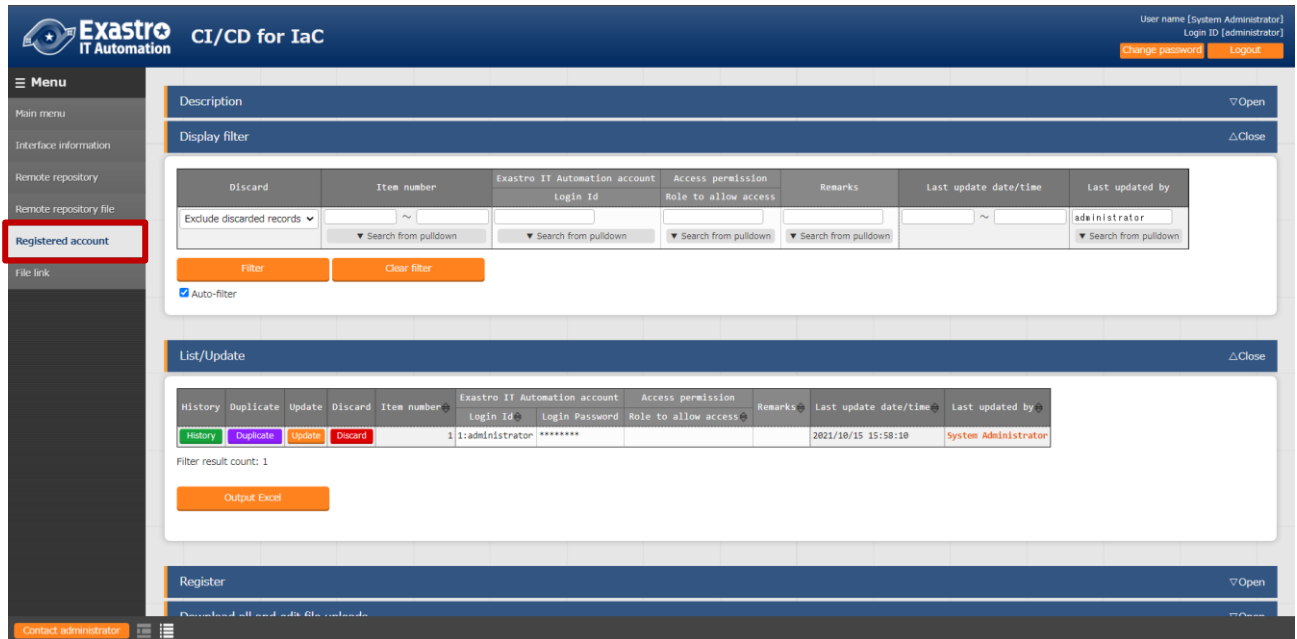


Figure 4.1-2Menu screen (Register account)

- (2) The Register account menu items are as following.

Table 4.1-3 Register account screen item list (Register account)

Item	Description	Input required	Input format	Restrictions
Login ID	Displays the login ID of the users registered in the “Management console – User list”. Select the user that is going to be used with RestAPI.	○	List selection	
Login PW	Input the login password of the user selected in the “Login ID” item. The password is the same as the one registered in the “Management console – User list” menu.	○	Manual	Max length 128 bytes
Remarks	Free description field	—	Manual	Max length 4000 bytes

4.1.3 File link

- (1) This menu allows users to link the link source files and the link destination files and register Operation and Movements that checks the validity of the link destination files.
When the link source file is updated, the backyard function automatically updates the link destination file, executes the registered operation and movement to verify the process and displays the results.

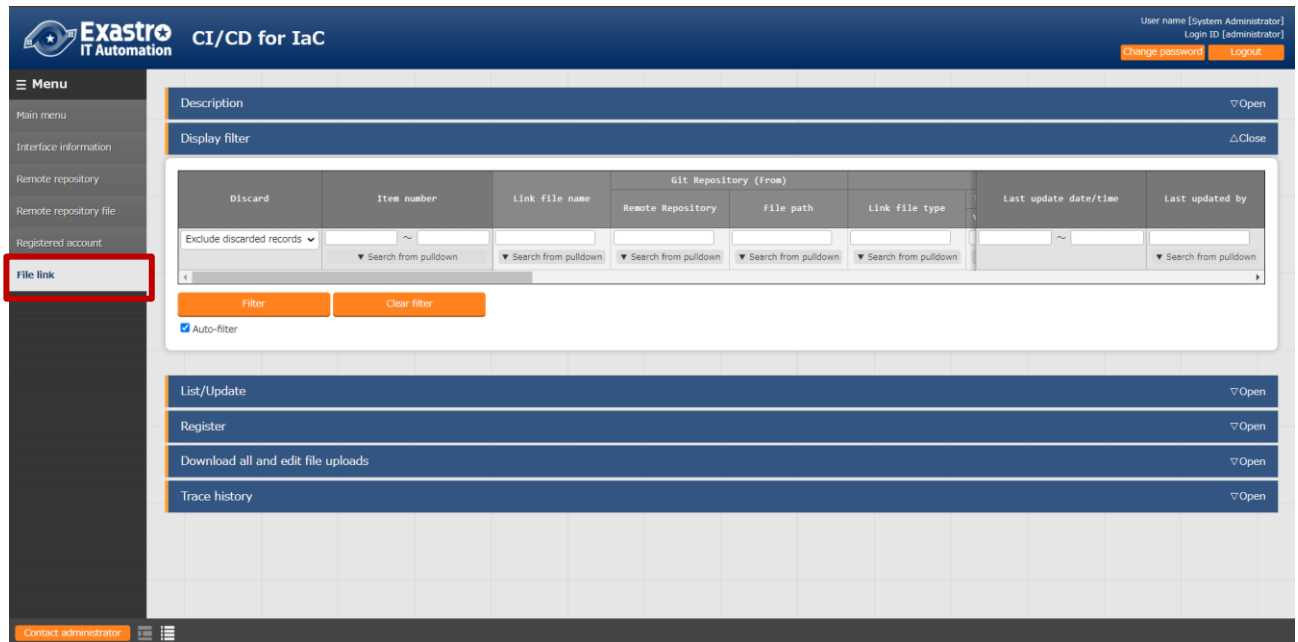


Figure 4.1-3 Menu screen (File link)

- (2) The File link screen items are as following.

Table 4.1-4 File link screen item list (File link)

Table 1-1 The link collection list (no link)

Item	Description	Input required	Input method	Restrictions										
Link file name	<p>Input the file name registered in the link destination file. This name is linked to one of the menus listed below depending on the link destination file type. Note that there the name must meet the same regulations as the items for the different menus. Use the regulations for the different items and input the file name</p> <p>*</p> <table><tr><th>Menu name</th><th>Item name</th></tr><tr><td>Ansible-Legacy/Playbook file collection</td><td>Paybook file name</td></tr><tr><td>Ansible-Pioneer/dialogue file collection</td><td>No target item</td></tr><tr><td>Ansible-LegacyRole/Role package list</td><td>Role package name</td></tr><tr><td>Ansible Common/File list</td><td>File embedded variable name</td></tr></table>	Menu name	Item name	Ansible-Legacy/Playbook file collection	Paybook file name	Ansible-Pioneer/dialogue file collection	No target item	Ansible-LegacyRole/Role package list	Role package name	Ansible Common/File list	File embedded variable name	<div>○</div>	Manual	Max length256 bytes
Menu name	Item name													
Ansible-Legacy/Playbook file collection	Paybook file name													
Ansible-Pioneer/dialogue file collection	No target item													
Ansible-LegacyRole/Role package list	Role package name													
Ansible Common/File list	File embedded variable name													

Item		Description		Input required	Input method	Restrictions											
		<table><tr><td>Ansible Common/Template list</td><td>Template embedded variable name</td></tr><tr><td>Terraform/Module files</td><td>Module file name</td></tr><tr><td>Terraform/Policy list</td><td>Policy name</td></tr></table> <p>The link process differs depending on conditions such as if the file name input in the link destination file name is registered or not. For more information, please see”エラー！ 参照元が見つかりません。 Updating the Link destination files.”.</p> <p>If you want to change the link destination file name, please see、”(5) Important points regarding changing the Link destination file name.”.</p>	Ansible Common/Template list	Template embedded variable name	Terraform/Module files	Module file name	Terraform/Policy list	Policy name									
Ansible Common/Template list	Template embedded variable name																
Terraform/Module files	Module file name																
Terraform/Policy list	Policy name																
Git repository (From)	Remote repository	Displays the remote repositories registered in the” エラー！ 参照元が見つかりません。エラー！ 参照元が見つかりません。 “. Select the remote repository that contains the source link files.		○	List selection												
	File path	Displays the link source files. Please select the path of the link source file. If the link destination file type is “Ansible-LegacyRole/Role package list” menu, Please see” エラー！ 参照元が見つかりません。 Points to note when registering file that are linked in the role package list to Git repositories.”		○	List selection												
Exastro IT automation(To)	Link file type	Select the link destination file type (Menu). The available file types are listed in the table below. Note that the different types needs to have their driver installed in order to be usable. <table><tr><th>Menu name</th><th>Driver</th></tr><tr><td>Ansible-Legacy/Playbook file collection</td><td rowspan="5">Ansible-Driver</td></tr><tr><td>Ansible-Pioneer/Dialogue file collection</td></tr><tr><td>Ansible-LegacyRole/Role package list</td></tr><tr><td>Ansible common/File list</td></tr><tr><td>Ansible common/Template list</td></tr><tr><td>Terraform/Module files</td><td rowspan="2">Terraform-Driver</td></tr><tr><td>Terraform/Policy list</td></tr></table>		Menu name	Driver	Ansible-Legacy/Playbook file collection	Ansible-Driver	Ansible-Pioneer/Dialogue file collection	Ansible-LegacyRole/Role package list	Ansible common/File list	Ansible common/Template list	Terraform/Module files	Terraform-Driver	Terraform/Policy list	○	List selection	
	Menu name	Driver															
Ansible-Legacy/Playbook file collection	Ansible-Driver																
Ansible-Pioneer/Dialogue file collection																	
Ansible-LegacyRole/Role package list																	
Ansible common/File list																	
Ansible common/Template list																	
Terraform/Module files	Terraform-Driver																
Terraform/Policy list																	
	Template Management	Variable definition	If the link destination file type is “Ansible common/Template list”, make sure to input the variable definitions needed for the file. This is not needed if the selected menu is something other than “Ansible common/Template list”	—	Manual	Max length 8192 bytes											

Item			Description	Input required	Input method	Restrictions
	Ansible-Pioneer	Dialogue type	Displays the dialogue file types registered in the "Ansible-Pioneer/Dialogue file type". This is not needed if the selected menu is something other than "Ansible-Pioneer/Dialogue file type"	—	List selection	
		OS type	Displays the OS types registered in the "Ansible-Pioneer/OS type" menu. If the "Ansible-Pioneer/Target file collection" menu is selected for the link destination file select the link destination file OS type from the list. This is not necessary if the link destination file is something else than the "Ansible-Pioneer/Target file collection" menu.	—	List selection	
	Execution login ID		Displays the Login ID of the users registered in the "エラー! 参照元が見つかりません。Registered Account" menu. Select a User login ID in order to access the link destination source files or execute operations from RestAPI. The User login ID selected must belong to a role that has the link "Can perform maintenance" for the menu specified in "Link destination file type". This link can be found in "Management console/Role - Menu link list" menu.	○	List selection	
	Assign Access permission role.		If you have added or updated a link destination file record, select the Access permission role setting contents. Without: blank (No access permission role) With: Set an access permission role that has the "Default access permission" regarding "Execution login ID". If there is not role with "Default access permission" set, the access permission role will set to be blank. If nothing is selected, the default value, "Without" will be selected.	—	List selection	
File Synchronization information	Automatic synchronization		Select if you want the link destination files automatically synchronize with the Git repository files when they change. Activated: Automatically updates the link destination files when the files in the Git repository change. Deactivated: Does not automatically update the link destination files when the files in the Git repository change.	○	List selection	Default: 有効

Item		Description	Input required	Input method	Restrictions
Delivery information	Operation	Displays the Operations registered in the "Basic console/Input operation list". Select the Operation that runs the movement you want to be executed when the link destination file changes.	—	List selection	
	Movement	Displays the Movements registered in the "Basic console/Movement list" menu. Select the Movement that will be run when the link destination file changes.	—	List selection	
	Dry run	Select the Movement execution mode. If marked (●), the movement will be executed in Dry run mode. The movement will not be executed in Dry run mode if nothing is selected.	—	List selection	
Remarks		Free description field	—	Manual	Max length 4000 bytes

- (3) The Synchronization status items for the Git repository items and the link destination file are as follows.

Table 4.1-5 File link Status item list (File link)

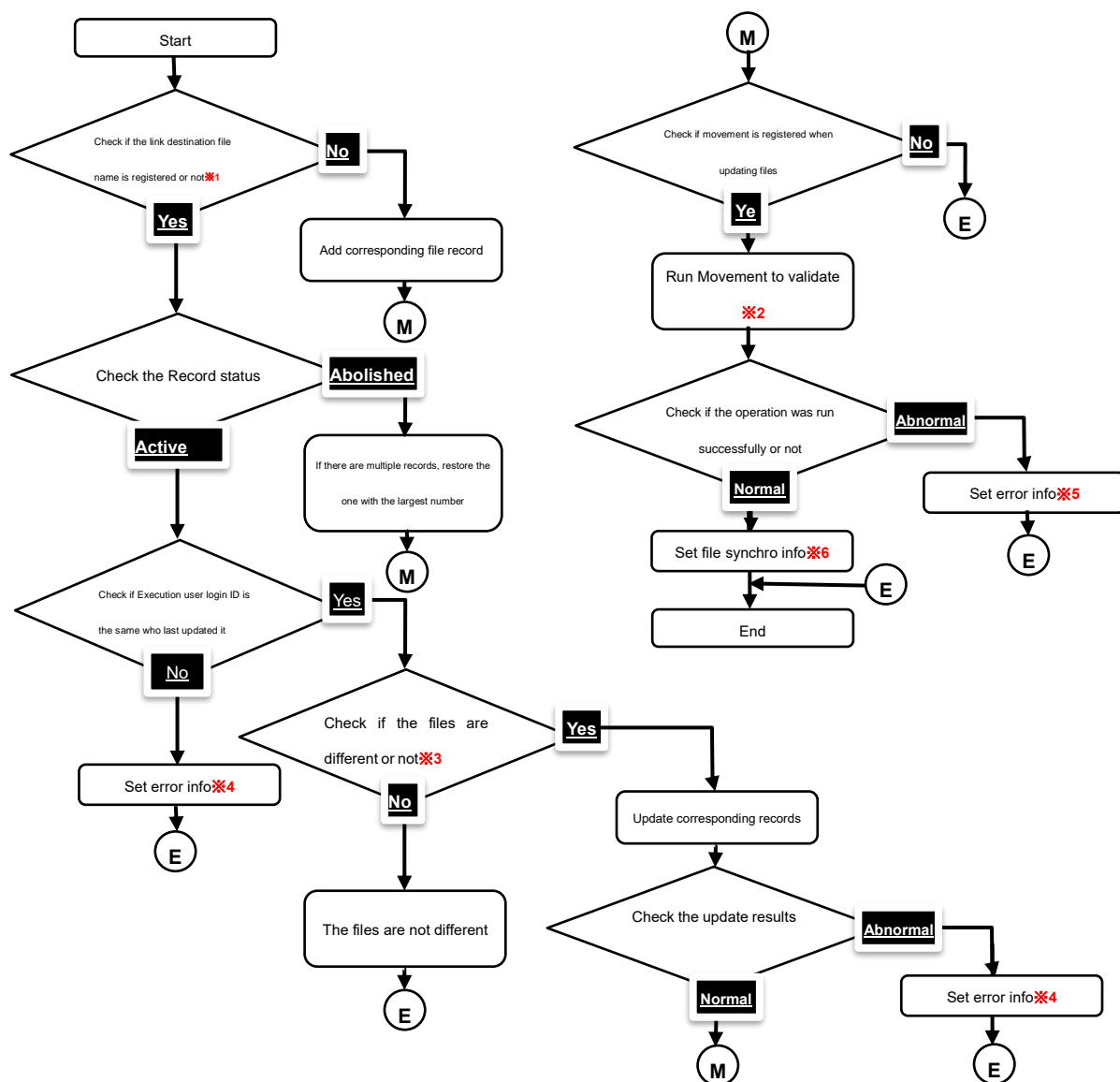
Item	Description	Remarks
File synchronization Information	State Displays the Synchronize status for the link source file and link destination file. There are 3 different statuses. Blank : When the record is newly registered, updated, restored after being abolished or resumed by clicking the "Resume" button. Normal: When it has successfully synchronized with the Git repository. Abnormal: When one of the following events have happened. <ul style="list-style-type: none"> • An error occurred when synchronizing. • The configured Operation and Movement failed to run because the link destination file was changed.※1 If the status displays "Abnormal", the synchronization process between the link source files and link destination files stops. If you want to resume the process, click the "Resume" button or update the record.	
	Detailed Information Displays detailed information regarding the synchronization when the process ends with the status "Abnormal". If you want to clear the "Detailed information" item, either click the "Resume" button or update the record	
	Last date/time Displays the date/time of when the last Synchronization happened. If you want to clear the "Last date/time" item, either click the "Resume" button or update the record	
	Last Executer's Login ID Displays Login ID of the user that last ran the synchronization process. If you want to clear the "Last Executer's Login ID" item, either click the "Resume" button or update the record	
	Resume button This button will only be displayed if the synchronization status changes to "Abnormal". Press this button to resume the synchronization process.	
Delivery Information	Detailed information Displays why an error occurred if the configured Operation and Movement failed to run because the link destination file was changed ※1 If the operation was run successfully, this item will not decide if the status will be "Normal" or "Abnormal"※2. Press the "Check Operation state" button in order to see the Execution results in the "X-Driver/ Execution status confirmation" menu. If you want to clear the "Detailed information" item, either click the "Resume" button or update the record.	
	Operation instance No Displays the operation instance number if the Operation and Movement was run successfully. If you want to clear the "Operation instance No" item, either click the "Resume" button or update the record.	
	Check Operation state button This button will only be displayed if the Operation and Movement was run successfully. Clicking this button will display "X-Driver/Execution status confirmation" menu where users can confirm the execution status.	

Item	Description	Remarks
	<p>For more information regarding the “Execution status confirmation” menu for the different drivers, please see the driver’s respective user manual.</p> <p>If you want to hide the button, either “Click the “Resume” button or update the record.</p>	

※1 For example: The Operation or movement has been deleted.

※2 For example: there is something wrong with the link destination file or there is missing information needed to execute the operation.

(4) Updating the Link destination files.



※1 If the Link destination file type is set to "Ansible-Pioneer/ Dialogue file collection", the link destination file name will be a combination of the dialogue type and the OS type.

※2 If there is missing information, the operation might fail to run.

※3 The item differences also includes the access permission roles and the remarks. The remarks will always be updated being blank.

- ※4 Makes the "4.1.3 File link" menu's "File synchronization information" -> "Status" display "Abnormal" and the "File synchronization information" -> "Detailed information" display the cause of error.
- ※5 Makes the "4.1.3 File link" menu's "File synchronization information" -> "Status" display "Abnormal" and the "Delivery information" -> "Detailed information" display the cause of error.
- ※6 Makes the "4.1.3 File link" menu's "File synchronization information" -> "Status" display "Normal".

- (5) Important points regarding changing the Link destination file name.
If you change the link destination file name, a new version of the same record with a new name will be created. The pre-updated version of the link destination file will still be there.

4.2 CI/CD For IaC hidden menus

This section will explain how to operate the menus that are hidden by default. In order to access the different menus, go to the “Management console/Role – menu link list” and restore the link corresponding to the user and the menu you want to access.

For more information, please see the “ITA_User_Manual_Management_Console” manual.

4.2.1. Interface information

- (1) In the “Interface information” menu, users can register the interface information needed in order to access the link destination files from ITA’s RestAPI. By default, the interface information required in order to access the installed ITA is registered. If you want to set the link file destination files to be somewhere else than the installed ITA, make sure to change the Interface information.

The screenshot shows the Exastro IT Automation CI/CD for IaC interface. The left sidebar contains a 'Menu' section with 'Interface information' highlighted. The main content area is titled 'Interface information' and includes a 'Display filter' section with search criteria for Item number, Host name, Protocol, Port, Access permission, Remarks, Last update date/time, and Last updated by. Below the filter is a table with one entry: Item number 1, Host name 127.0.0.1, Protocol http, Port 80, Access permission Role to allow access, Remarks, Last update date/time 2015/04/01 10:00:00, and Last updated by System Administrator. The table has 'History' and 'Update' buttons for the first row. At the bottom, there is an 'Output Excel' button and a 'Download all and edit file uploads' button.

Figure 4.2-1 Menu screen(Interface information)

- (2) The Interface information menu items are as following.

Table 4.2-1 Interface information menu items(Interface information)

Item	Description	Input required	Input method	Constraints
Host name	Input either the ITA Server host name or the IP Address.	○	Manual	Max length128 bytes Default: 127.0.0.1
Protocol	Input the ITA Server protocol (http/https)	○	Manual	Default: http
Port	Input the Server connection port (80/443).	○	Manual	Default: 80
Remarks	Free description field	—	Manual	Max length4000 bytes

- (1) The “Remote repository files” menu displays a list with the linked files. The information found in this menu is updated by the Backyard function. Do not add/update or delete records.

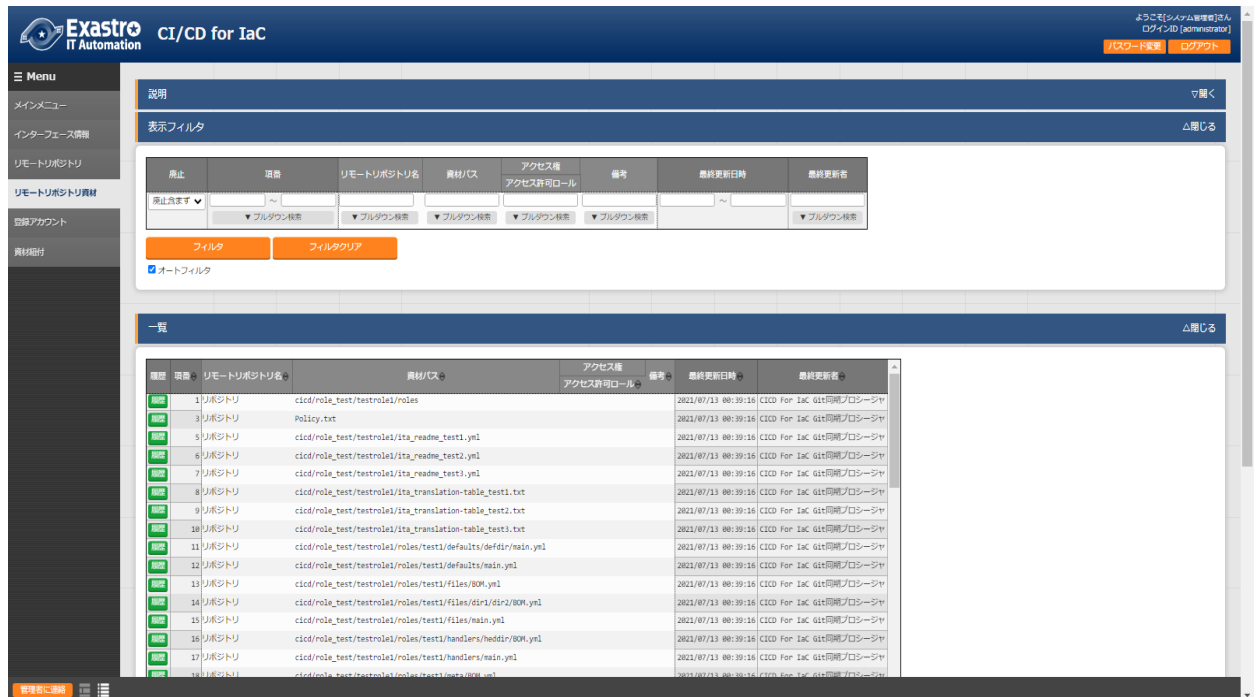


Figure 4.2-2Menu screen (Remote repository files)

- (2) The remote repository files item list are as follows.

Table 4.2-2 Remote repository files Item list(Remote repository files)

Item	Description	Input necessary	Input format	Constraints
Remote repository name	Displays the remote repository name registered in the“エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。”menu	○	List selection	
File path	The link source files' file path are displayed in a list. The file path for remote repositories that has it's synchronization status set to “Abnormal” in the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。”menu. Will not be displayed.	○	List selection	Max length4096 bytes

5. Operating

In addition to operating this function (and ITA) through the client PC's browser screen, it is also possible to operate it through system operation and maintenance. The following operations are available.

- Maintenance
- Changing log levels

5.1 About performing maintenance

The files needed in order to start/stop/restart the CICD For IaC function process are as follows.

Description	Target file name
CICD For IaC function Creates a local clone of the Git repository registered in the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。”menu.※1 Automatically updates the link destination files linked in the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。”menu and runs the operation and movements to verify.	ky_CICD_for_IaC_git_synchronize-workflow.service

※1 The path directory of the Git repository local clone is “/ITA install directory/ita-root/repositories/0000000000x”
 (“x” is the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。”10 numbers from the right.)

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

The commands for starting/stopping and restarting the process are as following.

Make sure to run the command with a user that has root permission.

① Start process

```
# systemctl start ky_CICD_for_IaC_git_synchronize-workflow.service ↵
```

② Stop process

```
# systemctl stop ky_CICD_for_IaC_git_synchronize-workflow.service ↵
```

③ Restart process.

```
# systemctl restart ky_CICD_for_IaC_git_synchronize-workflow.service
```

Change the target file name to fit your environment and run the commands.

5.2 Change log levels

① Change to NORMAL level

Open the following file and rewrite “DEBUG” to “NORMAL” on line 8.

Log level settings file: <install directory>/ita-root/confs/backyardconfs/ita_env

② Change to DEBUG level

Open the following file and rewrite “NORMAL” to “DEBUG” on line 8.

Log level settings file: <Install directory>/ita-root/confs/backyardconfs/ita_env

The settings will take effect after the process has been restarted.

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

6. Appendix

6.1 Points to note when registering files to Git repositories

The following section explains points to note when registering files to Git repositories

- ① Registering files with filenames that exceeds 255 bytes to the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。” menu will lead to an error when running the Git clone command.
- ② Registering files with file path+ file name that exceeds 4096 bytes to the Git repository registered in the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。” menu will lead to an error when running the 、Git clone command.

6.2 Points to note when registering file that are linked in the role package list to Git repositories.

The following section explain points to note when registering files that are linked in the role package list to Git repositories.

- ① Create a directory that contains a directory with the name "roles" and put all the files and directories required by the role packager function in there. Files compressed as role packages are in the upper directory of the roles directory. Note that even if you create a directory called "roles" directly under the Git repository route directory, it will not be recognized as the roles directory linked in the "Ansible-LegacyRole/Role package list" menu.

<Example>

Directories configured like the example below will have its "sample/roles" recognized as a role directory, but the "roles" directory will not be recognized.

Git repository route directory

└─rolesThe “roles” directory will not be recognized as a role directory.

├─ita_readme_test_role.yml
├─ita_translation-table_test_role.txt
├─test_role
│ └─defaults
│ │ main.yml
│ └─tasks
│ │ main.yml

└─sample

├─ita_readme_test_role.yml
├─ita_translation-table_test_role.txt
├─rolesRecognized as a roles directory.
│ └─test_role
│ │ └─defaults
│ │ │ main.yml
│ │ └─tasks
│ │ │ main.yml

Packaged as a
role package file.

The “sample/roles” path will be displayed in the “エラー! 参照元が見つかりません。エラー! 参照元が見つかりません。”menu. Select “Sample/roles” in order to link it in the “Ansible-LegacyRole/Role package list”menu.

