

利用手順マニュアル  
astroll Ansible driver  
別紙 Ansible利用ガイドライン  
astroll追加ルール

第1.0.0版

#### 免責事項

本書の内容はすべて日本電気株式会社が所有する著作権に保護されています。本書の内容の一部または全部を無断で転載および複写することは禁止されています。

本書の内容は将来予告なしに変更することがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。

日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性その他いかなる保証もいたしません。

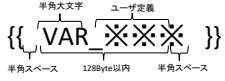

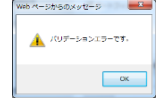
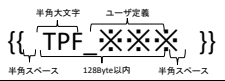
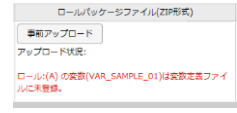
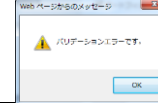
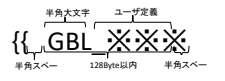
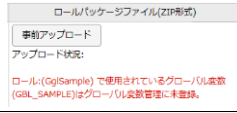
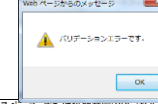
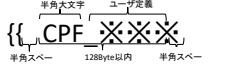
#### 商標

- ・ LinuxはLinus Torvalds氏の米国およびその他の国における登録商標または商標です。
- ・ Red Hatは、Red Hat, Inc.の米国およびその他の国における登録商標または商標です。
- ・ Apache、Apache Tomcat、Tomcatは、Apache Software Foundationの登録商標または商標です。
- ・ Ansibleは、Red Hat, Inc.の登録商標または商標です。

その他、本書に記載のシステム名、会社名、製品名は、各社の登録商標もしくは商標です。

なお、® マーク、TMマークは本書に明記しておりません。

## Ansible利用ガイドライン(astroll追加ルール)

No.	項目	担当モード			astroll取り込み可検例	astroll取り込み不可例	ルールに準拠しない場合の影響機能	影響現象	エラー表示例
1	変数を使用する際、playbook内の変数を以下とする。 ※ユーザ定義範囲の記載ルールはAnsibleルールに準ずる 	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	-- name: ファイル内容確認 command: cat /tmp/[[ VAR_SAMPLE ]]	-「[[または]]」と変数名の間に「半角スペース」がない command: cat /tmp/[[VAR_SAMPLE]] -「[[または]]」と変数名の間の「半角スペース」が4個以上ある command: cat /tmp/[[ VAR_SAMPLE ]] -接続文字(VAR)が小文字になっている command: cat /tmp/[[ var_SAMPLE ]] -接続文字「VAR」が「J」がない command: cat /tmp/[[ VARSAMPLE ]] -変数名が129文字以上ある	代入値管理	・登録の場合 Movementを選択しても変数名に候補として表示しません。 作業実行時に変数未定義のエラーとなります。 ・更新の場合(代入値管理登録後にastroll追加ルール外の記載のplaybookに更新した場合) 変数名に「ID変換失敗(n)」「&nは登録時の管理番号」と表示します。 作業実行時に変数未定義のエラーとなります。	
2	astrollWebUI上のテンプレート管理メニューでテンプレート素材を登録する際、 テンプレート埋込変数名の接続文字をTPFとし、TPLを含め128Byte以内とする。	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	テンプレート埋込変数名: テンプレート素材 TPF_SAMPLE sample.txt	TPE_SAMPLE	テンプレート管理	登録画面でバリデーションチェックを行っているので、astroll追加ルールに準拠しないテンプレート埋込変数名は登録できません。	
3	アップロードしたテンプレート素材をplaybook内で指定する場合は、playbook内の変数を以下とする。 	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	-- name: ファイル配置 template: src= [[ TPF_SAMPLE ]] dest=/tmp/SAMPLE.txt	-「[[または]]」と変数名の間に「半角スペース」がない template: src= [[TPF_SAMPLE]] dest=/tmp/SAMPLE.txt template: src= [[ TPF_SAMPLE ]] dest=/tmp/SAMPLE.txt -「[[または]]」と変数名の間の「半角スペース」が4個以上ある template: src= [[ TPF_SAMPLE ]] dest=/tmp/SAMPLE.txt -接続文字(TPF)が小文字になっている template: src= [[ tpf_SAMPLE ]] dest=/tmp/SAMPLE.txt -接続文字「TPF」が「J」がない template: src= [[ TPFSAMPLE ]] dest=/tmp/SAMPLE.txt -変数名が129文字以上ある	テンプレート管理	登録したテンプレート素材への置換が行われません。	※ 実行状況(エラーログ) フィルタ: <input type="text"/> <input type="checkbox"/> 実行状況の表示 実行状況(エラーログ) 実行状況(エラーログ)
4	tasksディレクトリ内のplaybookやtemplatesディレクトリ内のテキストファイルで使用する変数は、 defaultsディレクトリ内のmain.ymlに定義する。 ※具体値は必須ではない	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	[tasks/main.yml] -- name: テンプレートファイル内容確認 command: cat /tmp/[[ VAR_SAMPLE_01 ]] command: cat /tmp/[[ VAR_SAMPLE_02 ]] command: cat /tmp/[[ VAR_SAMPLE_03 ]] [defaults/main.yml] VAR_SAMPLE_01: SAMPLE_01.txt VAR_SAMPLE_02: SAMPLE_02.txt VAR_SAMPLE_03: SAMPLE_03.txt	[tasks/main.yml] -- name: テンプレートファイル内容確認 command: cat /tmp/[[ VAR_SAMPLE_01 ]] command: cat /tmp/[[ VAR_SAMPLE_02 ]] command: cat /tmp/[[ VAR_SAMPLE_03 ]] [defaults/main.yml] VAR_SAMPLE_01: SAMPLE_01.txt VAR_SAMPLE_02: SAMPLE_02.txt VAR_SAMPLE_03: SAMPLE_03.txt	ロールパッケージ管理	事前アップロード時に警告を表示します。	
5	astrollWebUI上のグローバル変数管理メニューでグローバル変数を登録する際、 グローバル変数名の接続文字をGBLとし、GBLを含め128Byte以内とする。	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	グローバル変数名: 具体値 GBL_SAMPLE sample.txt	GBL_SAMPLE	グローバル変数管理	登録画面でバリデーションチェックを行っているので、astroll追加ルールに準拠しないグローバル変数名は登録できません。	
6	グローバル変数を使用する際、playbook内の変数を以下とする。 ※ユーザ定義範囲の記載ルールはAnsibleルールに準ずる 	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	-- name: ファイル内容確認 command: cat /tmp/[[ GBL_SAMPLE ]]	・接続文字(GBL)が小文字になっている command: cat /tmp/[[ gbl_SAMPLE ]] ・接続文字「GBL」が「J」がない command: cat /tmp/[[ GBLSAMPLE ]] ・変数名が129文字以上ある	グローバル変数管理	作業実行時にグローバル変数の置換が行われなかったためエラーとなります。	・接続文字(GBL)がルールに準拠していない場合 実行状況(エラーログ) フィルタ: <input type="text"/> <input type="checkbox"/> 実行状況の表示 実行状況(エラーログ) 実行状況(エラーログ) ・変数名が129文字以上ある場合 ※ 実行状況(エラーログ) フィルタ: <input type="text"/> <input type="checkbox"/> 実行状況の表示 実行状況(エラーログ) 実行状況(エラーログ)
7	ロールパッケージファイルに含むplaybookでグローバル変数を使用する場合、アップロードする前にグローバル変数を登録する。	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	ロールパッケージを登録する前にグローバル変数を登録している	ロールパッケージを登録する前にグローバル変数を登録していない	ロールパッケージ管理	事前アップロード時に警告を表示します。	
8	astrollWebUI上のファイル管理メニューでファイル埋込変数を登録する際、 ファイル埋込変数名の接続文字をCPFとし、CPFを含め128Byte以内とする。	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	ファイル埋込変数名: ファイル素材 CPF_SAMPLE sample.txt	CPF_SAMPLE	ファイル管理	登録画面でバリデーションチェックを行っているので、astroll追加ルールに準拠しないファイル埋込変数名は登録できません。	
9	ファイル埋込変数を使用する際、playbook内の変数を以下とする。 ※ユーザ定義範囲の記載ルールはAnsibleルールに準ずる 	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	-- name: ファイル配置 copy: src= [[ CPF_SAMPLE ]] dest=/tmp/SAMPLE.txt	-「[[または]]」と変数名の間に「半角スペース」がない copy: src= [[CPF_SAMPLE]] dest=/tmp/SAMPLE.txt copy: src= [[ CPF_SAMPLE ]] dest=/tmp/SAMPLE.txt -「[[または]]」と変数名の間の「半角スペース」が4個以上ある copy: src= [[ CPF_SAMPLE ]] dest=/tmp/SAMPLE.txt -接続文字(CPF)が小文字になっている copy: src= [[ cpf_SAMPLE ]] dest=/tmp/SAMPLE.txt -接続文字「CPF」が「J」がない copy: src= [[ CPFSAMPLE ]] dest=/tmp/SAMPLE.txt -変数名が129文字以上ある	ファイル管理	作業実行時にファイル埋込変数の置換が行われなかったためエラーとなります。	・接続と変数名の間の半角スペース、または接続文字(CPF)がルールに準じていない場合 ※ 実行状況(エラーログ) フィルタ: <input type="text"/> <input type="checkbox"/> 実行状況の表示 実行状況(エラーログ) 実行状況(エラーログ) ・変数名が129文字以上ある場合 ※ 実行状況(エラーログ) フィルタ: <input type="text"/> <input type="checkbox"/> 実行状況の表示 実行状況(エラーログ) 実行状況(エラーログ)

No.	項目	操作モード			astroll取り込み可能例	astroll取り込み不可例	ルールに準拠しない場合の影響機能	影響現象	エラー表示例
		Legacy	Role	Pioneer					
10	ロールパッケージファイルに含むplaybookでファイル埋込変数を使用する場合、アップロードする前にファイル埋込変数を登録する。	x	○	x	ロールパッケージを登録する前にファイル埋込変数を登録している	ロールパッケージを登録する前にファイル埋込変数を登録していない	ロールパッケージ管理	事前アップロード時に警告を表示します。	
11	Ansible-LegacyRoleコンソールでアップロードするロールパッケージファイルは「roles」のある階層のディレクトリをzipにて圧縮したもの(右記記載例参照)を用意する。	x	○	x		・rolesディレクトリがない xxxxx/A/tasks/main.yml xxxxx/A/defaults/main.yml	ロールパッケージ管理	事前アップロード時に警告を表示します。	
12	各ロールのデフォルト変数定義ファイルで配列変数を定義する際の注意事項 ・単一ロールパッケージ ロールを跨って同じ変数名を使用しているが配列変数のメンバー変数の定義が違う際、ロールを跨って同じ変数名を使用しているが配列変数と一般の変数が混在している際。 ⇒アップロード時にエラーとなります。 ・全ロールパッケージ ロールパッケージを跨って同一変数名を使用しているが配列変数のメンバー変数の定義が違う際。 ロールパッケージを跨って同一変数名を使用しているが配列変数と一般の変数が混在している際。 ⇒代入値管理に変数名が表示されません。	x	○	x	別シート【配列変数を定義する際の注意事項】参照	別シート【配列変数を定義する際の注意事項】参照	・単一ロールパッケージ ロールパッケージ管理 ・全ロールパッケージ 代入値管理	・単一ロールパッケージ ロールパッケージを登録できません。 ・全ロールパッケージ 配列変数に値を設定できません。 ロールパッケージに定義されている値で実行します。	・単一ロールパッケージ  ・全ロールパッケージ 
13	・配列変数を使用する際は、defaultsディレクトリのmain.ymlに右記記載例を参考に記入する ※最低1行は定義必須	x	○	x	<pre> [defaults/main.yml] VAR_01: - [ VAR_SAMPLE_01: "001", VAR_SAMPLE_02: "002" ] - [ VAR_SAMPLE_01: "101", VAR_SAMPLE_02: "102" ]           </pre>	<pre> [defaults/main.yml] VAR_01:           </pre>	代入値管理	配列変数として認識されないで、メンバー変数に値を設定することができません。 作業実行時にメンバー変数未定義のエラーとなります。	
14	構築コードの基本書式におけるインデントは2倍で調整する	○	○	○	・△name: サービススクリプト配置 △△△template: △△△△src: "[ item.src ]" △△△△dest: "[ item.dest ]" ～略～ △半角スペース	・インデントが2倍額になっていない ・△name: サービススクリプト配置 △△△template: △△△△src: "[ item.src ]" △△△△dest: "[ item.dest ]" ～略～ △半角スペース	作業実行	作業実行時に想定外エラーとなります。	

## 配列変数を定義する際の注意事項

No.	ロールパッケージ	roles	defaults/main.yml	動作	記載内容
1	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	○	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>配列変数のメンバー変数の定義が同じ</li> <li>メンバー変数の記載順序が同じ</li> </ul>
		B	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }		
2	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	○	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>配列変数のメンバー変数の定義が同じ</li> <li>メンバー変数の記載順序が異なる</li> </ul>
		B	VAR_SAMPLE: - { VAR_002: "bbbb", VAR_001: "aaaa" }		
3	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	×	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>配列変数のメンバー変数の定義が異なる</li> </ul>
		B	VAR_SAMPLE: - { VAR_003: "aaaa", VAR_004: "bbbb" }		
4	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	×	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>一般の変数と配列変数が混在している</li> </ul>
		B	VAR_SAMPLE: aaaa		
5	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	○	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>配列変数のメンバー変数の定義が同じ</li> <li>メンバー変数の記載順序が同じ</li> </ul>
	サンプル02	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }		
6	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	○	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>配列変数のメンバー変数の定義が同じ</li> <li>メンバー変数の記載順序が異なる</li> </ul>
	サンプル02	A	VAR_SAMPLE: - { VAR_002: "bbbb", VAR_001: "aaaa" }		
7	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	×	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>配列変数のメンバー変数の定義が異なる</li> </ul>
	サンプル02	A	VAR_SAMPLE: - { VAR_003: "aaaa", VAR_004: "bbbb" }		
8	サンプル01	A	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" }	×	<ul style="list-style-type: none"> <li>変数名が同じ</li> <li>一般の変数と配列変数が混在している</li> </ul>
	サンプル02	A	VAR_SAMPLE: aaaa		

astrollで取り込めるAnsibleのRole応用例

No.	変数	モジュール/構文	動作内容	ディレクトリ/ファイル		
				defaults/main.yml	tasks/main.yml	templates/※※※.txt
1	一般変数	with_items	一般変数を使用した with_itemsモジュール	VAR_test: SAMPLE	- name: ファイル内容確認 shell: cat /tmp/[[ item ]] with_items: - "[[ VAR.SAMPLE ]]"	-
2	一般変数	if	一般変数を使用した if構文	VAR_SAMPLE: True	- name: 配列変数を仕込んだテンプレートを配置 template: src=var_array_sample.txt dest=/tmp/※※※.txt	{% if VAR.SAMPLE is defined and VAR.SAMPLE == True %} 出力したい文字 {% endif %}
3	配列変数	with_items	配列変数を使用した with_itemsモジュール	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" } - { VAR_001: "cccc", VAR_002: "dddd" } - { VAR_001: "eeee", VAR_002: "ffff" }	- name: ファイル内容確認 shell: cat /tmp/[[ item.VAR_001 ]] with_items: - "[[ VAR.SAMPLE ]]"  - name: ファイル内容確認 shell: cat /tmp/[[ item.VAR_002 ]] with_items: - "[[ VAR.SAMPLE ]]"	-
4	配列変数	for	配列変数を使用した for構文	VAR_SAMPLE: - { VAR_001: "aaaa", VAR_002: "bbbb" } - { VAR_001: "cccc", VAR_002: "dddd" } - { VAR_001: "eeee", VAR_002: "ffff" }	- name: 配列変数を仕込んだテンプレートを配置 template: src=var_array_sample.txt dest=/tmp/※※※.txt	{% for tmp.SAMPLE in VAR.SAMPLE %} [[ tmp.SAMPLE.VAR_001 ]] [[ tmp.SAMPLE.VAR_002 ]] {% endfor %}
5	配列変数	for if	配列変数を使用した for構文及び if構文でのループ制御	VAR_SAMPLE: - { VAR_01: 1, VAR_02: 001 } - { VAR_01: 0, VAR_02: 002 } - { VAR_01: 1, VAR_02: 003 }	- name: ループと条件分岐の応用例 shell: {% for tmp.SAMPLE in VAR.SAMPLE %} {% if tmp.SAMPLE.VAR_01 == 1 %} echo [[ tmp.SAMPLE.VAR_02 ]] {% else %} true {% endif %} {% endfor %}	-
6	配列変数	for if	配列変数を使用した for構文及び if構文でのループ制御	VAR_SAMPLE: - { VAR_01: 1, VAR_02: 001 } - { VAR_01: 0, VAR_02: 002 } - { VAR_01: 1, VAR_02: 003 }	- name: 配列変数を仕込んだテンプレートを配置 template: src=var_array_sample.txt dest=/tmp/※※※.txt	{% for tmp.SAMPLE in VAR.SAMPLE %} {% if tmp.SAMPLE.VAR_01 == 1 %} echo [[ tmp.SAMPLE.VAR_02 ]] {% else %} true {% endif %} {% endfor %}