

JP1 Version 11

JP1/Automatic Operation コマンド・API リファレンス

3021-3-A91-40

## 前書き

### ■ 対象製品

P-2A2C-E1BL JP1/Automatic Operation 11-51 (適用 OS : Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016)

製品構成一覧および内訳形名

- ・ P-CC2A2C-EABL JP1/Automatic Operation - Server 11-51 (適用 OS : Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016)
- ・ P-CC2A2C-EBBL JP1/Automatic Operation - Contents 11-51 (適用 OS : Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016)

P-2A2C-E3BL JP1/Automatic Operation Content Pack 11-51 (適用 OS : Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016)

P-822C-E1BL JP1/Automatic Operation 11-51 (適用 OS : Linux 6 (x64), Linux 7, Oracle Linux 6 (x64), Oracle Linux 7, CentOS 6 (x64), CentOS 7, SUSE Linux 12)

製品構成一覧および内訳形名

- ・ P-CC822C-EABL JP1/Automatic Operation - Server 11-51 (適用 OS : Linux 6 (x64), Linux 7, Oracle Linux 6 (x64), Oracle Linux 7, CentOS 6 (x64), CentOS 7, SUSE Linux 12)
- ・ P-CC822C-EBBL JP1/Automatic Operation - Contents 11-51 (適用 OS : Linux 6 (x64), Linux 7, Oracle Linux 6 (x64), Oracle Linux 7, CentOS 6 (x64), CentOS 7, SUSE Linux 12)

P-822C-E3BL JP1/Automatic Operation Content Pack 11-51 (適用 OS : Linux 6 (x64), Linux 7, Oracle Linux 6 (x64), Oracle Linux 7, CentOS 6 (x64), CentOS 7, SUSE Linux 12)

### ■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

### ■ 商標類

HITACHI, HiRDB, JP1 は、株式会社 日立製作所の商標または登録商標です。

Active Directory は、米国 Microsoft Corporation 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。

AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Intel は、アメリカ合衆国および / またはその他の国における Intel Corporation またはその子会社の商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation 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。Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

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

This product includes software developed by IAIK of Graz University of Technology.

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標がついた製品は、米国 Sun Microsystems, Inc. が開発したアーキテクチャに基づくものです。

SQL Server は、米国 Microsoft Corporation 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。

UNIX は、The Open Group の商標です。

Windows は、米国 Microsoft Corporation 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。

Windows Server は、米国 Microsoft Corporation 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。

XenDesktop は、Citrix Systems, Inc. および／またはその一つもしくは複数の子会社の商標であり、米国の特許商標庁および他の国において登録されている場合があります。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

RSA および BSAFE は、米国 EMC コーポレーションの米国およびその他の国における商標または登録商標です。

JP1/Automatic Operation は、米国 EMC コーポレーションの RSA BSAFE(R) ソフトウェアを搭載しています。

This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project.

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This product includes software developed by the University of California, Berkeley and its contributors.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc (Bellcore).

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The original software is available from <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

This product includes software developed by Ralf S. Engelschall <[rse@engelschall.com](mailto:rse@engelschall.com)> for use in the mod\_ssl project (<http://www.modssl.org/>).

This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>).

This product includes software developed by the Java Apache Project for use in the Apache JServ servlet engine project (<http://java.apache.org/>).

This product includes software developed by Andy Clark.

Java is a registered trademark of Oracle and/or its affiliates.

**HITACHI**  
Inspire the Next

株式会社 日立製作所



## ■ 発行

2020 年 4 月 3021-3-A91-40

## ■ 著作権

All Rights Reserved. Copyright (C) 2016, 2020, Hitachi, Ltd.

## 変更内容

### 変更内容 (3021-3-A91-40) JP1/Automatic Operation 11-51

追加・変更内容	変更箇所
hcnds64ssltool シンタックスを修正した。	1.5.6
backupsystem コマンドの注意事項を変更した。	1.7.1
サポートするメソッドの説明を追加した。	2.2.7
API の使用例を変更した。	2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12

単なる誤字・脱字などはお断りなく訂正しました。

## はじめに

このマニュアルは、JP1/Automatic Operation のコマンドおよび API について説明したものです。なお、このマニュアルでは、JP1/Automatic Operation を JP1/AO と表記します。

JP1/AO マニュアルの参考情報および用語解説については、マニュアル「[JP1/AO 導入・設計ガイド](#)」を参照してください。

### ■ 対象読者

このマニュアルは、次の方にお読みいただくことを前提に説明しています。

- JP1/AO のコマンドを利用する方
- JP1/AO の API を利用する方

### ■ マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

表記	製品名
Active Directory	Microsoft(R) Active Directory
Hyper-V	Microsoft(R) Windows Server(R) 2008 R2 Hyper-V(R)
	Microsoft(R) Windows Server(R) 2012 Hyper-V(R)
	Microsoft(R) Windows Server(R) 2012 R2 Hyper-V(R)
	Microsoft(R) Windows Server(R) 2016 Hyper-V(R)
Windows Server 2008 R2	Microsoft(R) Windows Server(R) 2008 R2 Datacenter
	Microsoft(R) Windows Server(R) 2008 R2 Enterprise
	Microsoft(R) Windows Server(R) 2008 R2 Standard
Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Datacenter
	Microsoft(R) Windows Server(R) 2012 Standard
Windows Server 2012 R2	Microsoft(R) Windows Server(R) 2012 R2 Datacenter
	Microsoft(R) Windows Server(R) 2012 R2 Standard
Windows Server 2016	Microsoft(R) Windows Server(R) 2016 Datacenter
	Microsoft(R) Windows Server(R) 2016 Standard

Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, および Windows Server 2008 R2 を総称して Windows と表記することがあります。

## ■ マニュアルで使用している書式について

このマニュアルで使用する書式について説明します。

### 説明文で使用する書式

書式	説明
文字列	可変の値を示します。 (例) 日付は YYYYMMDD の形式で指定します。
[ ]	ウィンドウ, ダイアログボックス, タブ, パネル, メニュー, ボタンなどの画面上の要素名を示します。
[ ] - [ ]	メニューを連続して選択することを示します。 (例) [ファイル] - [新規作成] を選択します。 上記の例では, [ファイル] メニュー内の [新規作成] を選択することを示します。
[ ]	画面などに入力する文字列を示します。

### 文法で使用する書式

書式	説明
文字列	可変の値を示します。 (例) 日付は YYYYMMDD の形式で指定します。
[ ]	この記号で囲まれている項目は任意に指定できます (省略もできます)。 (例) [A] は「何も指定しない」か「A を指定する」ことを示します。
{ }	この記号で囲まれている複数の項目の中から, 必ず 1 組の項目を選択します。項目の区切りは   で示します。 (例) {A B C} は, 「A, B または C のどれかを指定する」ことを示します。
	この記号で区切られている項目は, 複数項目のうちどれかを指定できます。 (例) A B C は, 「A, B, または C」を示します。

## ■ インストール先フォルダの表記

このマニュアルでは, Windows 版 JP1/AO のデフォルトのインストール先フォルダを次のように表記しています。

JP1/AO のインストール先フォルダ

システムドライブ¥Program Files¥Hitachi¥JP1AO

共通コンポーネントのインストール先フォルダ

システムドライブ¥Program Files¥Hitachi¥HiCommand¥Base64

なお、Linux 版 JP1/AO のインストール先フォルダは次のとおりです。

JP1/AO のインストール先フォルダ

- /opt/jplao/
- /var/opt/jplao/

共通コンポーネントのインストール先フォルダ

/opt/HiCommand/Base64

## ■ マニュアルに掲載している画面図について

製品の改良などによって、このマニュアルに掲載されている画面はご使用の製品画面と一部異なることがあります。あらかじめご了承ください。



# 目次

前書き	2
変更内容	5
はじめに	6

<b>1</b>	<b>コマンド</b>	<b>14</b>
1.1	コマンド一覧	15
1.2	コマンド使用時の注意事項	18
1.3	コマンドの引数に指定できる文字	20
1.4	コマンドの記述形式	21
1.5	構築関連のコマンド	22
1.5.1	encryptpassword (パスワードファイルの作成)	22
1.5.2	hcmds64checkauth (外部認証サーバとの接続確認)	24
1.5.3	hcmds64fwcancel (Windows ファイアウォールの例外登録)	26
1.5.4	hcmds64intg (認証データの削除・確認)	27
1.5.5	hcmds64ldapuser (LDAP 検索用ユーザー登録・削除)	28
1.5.6	hcmds64ssltool (秘密鍵・自己署名証明書の作成)	31
1.5.7	setupcluster (クラスタ環境の構築)	34
1.6	運用関連のコマンド	36
1.6.1	deleteremoteconnection (エージェントレス接続先定義の削除)	36
1.6.2	deleteservicetemplate (サービステンプレートの削除)	38
1.6.3	hcmds64chgurl (URL 情報の更新)	41
1.6.4	hcmds64srv (JP1/AO の開始・停止・状態表示)	42
1.6.5	hcmds64unlockaccount (ユーザーアカウントのロック解除)	46
1.6.6	importservicetemplate (サービステンプレートのインポート)	47
1.6.7	listremoteconnections (エージェントレス接続先定義の一覧出力)	50
1.6.8	listservices (サービス・サービステンプレートの一覧出力)	53
1.6.9	listtasks (タスクの一覧出力, タスク詳細情報出力)	58
1.6.10	setremoteconnection (エージェントレス接続先定義の追加・更新)	68
1.6.11	stoptask (タスクの実行停止)	71
1.6.12	submittask (サービスの実行, タスクの一括再登録)	74
1.7	保守・メンテナンス関連のコマンド	85
1.7.1	backupsystem (JP1/AO システムのバックアップ)	85
1.7.2	hcmds64dbrepair (データベースの再作成)	88
1.7.3	hcmds64dbsrv (データベースの開始・停止)	90
1.7.4	hcmds64dbtrans (データベースのバックアップ・リストア)	92

- 1.7.5 hcmds64getlogs (ログ情報の採取) 95
- 1.7.6 restoresystem (JP1/AO システムのリストア) 99

## 2 API 103

- 2.1 API 一覧 104
- 2.2 API 共通の仕様 110
  - 2.2.1 通信方式 110
  - 2.2.2 セキュリティと認証 110
  - 2.2.3 入出力形式 111
  - 2.2.4 ネームスペース 111
  - 2.2.5 リクエスト形式 112
  - 2.2.6 レスポンス形式 113
  - 2.2.7 サポートするメソッド 114
  - 2.2.8 API で管理できるドメイン名とリソース 114
  - 2.2.9 クエリパラメータ 115
  - 2.2.10 リクエストヘッダー 118
  - 2.2.11 HQL 規格の使用 119
  - 2.2.12 ドメインオブジェクトの形式 121
  - 2.2.13 レスポンスヘッダー 123
  - 2.2.14 リソースのメンバー 123
  - 2.2.15 JP1/AO を操作する API で返却されるメンバー 147
  - 2.2.16 実行できる操作を取得する API で返却されるメンバー 148
  - 2.2.17 ステータスコード 149
  - 2.2.18 エラー情報 150
- 2.3 API の記述形式 152
- 2.4 サービステンプレート関連の API 153
  - 2.4.1 サービステンプレート一覧の取得 153
  - 2.4.2 サービステンプレートの情報取得 156
  - 2.4.3 サービステンプレートの削除 158
  - 2.4.4 サービステンプレートに対する操作一覧の取得 160
  - 2.4.5 サービステンプレートのインポートに必要な HTML ファイルの取得 162
  - 2.4.6 サービステンプレートのインポート 164
  - 2.4.7 サービステンプレートのエクスポートに必要な情報の取得 166
  - 2.4.8 サービステンプレートのエクスポート 168
  - 2.4.9 サービステンプレートの詳細説明を表示する URL の取得 170
  - 2.4.10 サービステンプレートを基にしたサービスの作成に必要な情報の取得 171
  - 2.4.11 サービステンプレートを基にしたサービスの作成 174
- 2.5 サービス関連の API 177
  - 2.5.1 サービス一覧の取得 177

2.5.2	サービスの情報取得	179
2.5.3	サービスの編集	182
2.5.4	サービスの削除	185
2.5.5	サービスに対する操作一覧の取得	186
2.5.6	サービス実行に必要な情報の取得	190
2.5.7	サービスの実行	192
2.5.8	サービスのカウンターのリセットに必要な情報の取得	195
2.5.9	サービスのカウンターのリセット	197
2.5.10	サービスの状態を「リリース」に変更する操作に必要な情報の取得	199
2.5.11	サービスの状態を「リリース」に変更	201
2.5.12	サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得	202
2.5.13	サービスの状態を「メンテナンス」に変更	204
2.5.14	サービスの状態を「無効」に変更する操作に必要な情報の取得	206
2.5.15	サービスの状態を「無効」に変更	208
2.5.16	サービスの詳細説明の URL を取得	210
2.5.17	サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得	211
2.5.18	サービスが使用するサービステンプレートのバージョン変更	213
2.6	スケジュール関連の API	217
2.6.1	スケジュール一覧の取得	217
2.6.2	スケジュールの情報取得	220
2.6.3	スケジュールに対する操作一覧の取得	222
2.6.4	スケジュールのキャンセルに必要な情報の取得	224
2.6.5	スケジュールのキャンセル	226
2.6.6	スケジュールの一時停止に必要な情報の取得	228
2.6.7	スケジュールの一時停止	230
2.6.8	スケジュールの再開に必要な情報の取得	232
2.6.9	スケジュールの再開	234
2.7	タスク関連の API	237
2.7.1	タスク一覧の取得	237
2.7.2	タスクの情報取得	240
2.7.3	タスクの編集	242
2.7.4	タスクの削除	245
2.7.5	タスク操作一覧の取得	246
2.7.6	タスク実行停止に必要な情報の取得	249
2.7.7	タスクの実行停止	251
2.7.8	タスクの強制停止に必要な情報の取得	253
2.7.9	タスクの強制停止	255
2.7.10	タスクの再実行に必要な情報の取得	257
2.7.11	タスクの再実行	259

2.7.12	応答待ち中のタスクに応答するために必要な情報の取得	262
2.7.13	応答待ち中のタスクに応答	264
2.7.14	タスクのリトライに必要な情報の取得 (失敗したステップからリトライ)	267
2.7.15	失敗したステップからリトライ	269
2.7.16	タスクのリトライに必要な情報の取得 (失敗した次のステップからリトライ)	271
2.7.17	失敗した次のステップからリトライ	273
2.7.18	タスクのアーカイブに必要な情報の取得	275
2.7.19	タスクのアーカイブ	276
2.7.20	ステップ一覧の取得	279
2.7.21	タスクログの取得	281
2.8	履歴関連の API 一覧	287
2.8.1	履歴一覧の取得	287
2.8.2	履歴の削除 (条件指定)	289
2.8.3	履歴の取得	291
2.8.4	履歴の削除 (ID 指定)	293
2.8.5	履歴に対する操作一覧の取得	295
2.9	プロパティ関連の API 一覧	297
2.9.1	プロパティ定義一覧の取得	297
2.9.2	プロパティ定義情報の取得	304
2.9.3	プロパティ定義に対する操作一覧の取得	306
2.9.4	プロパティ定義およびプロパティ値一覧の取得	308
2.9.5	プロパティ値一覧の取得	311
2.9.6	プロパティ値の一括更新	315
2.9.7	プロパティ値の取得	319
2.9.8	プロパティ値の更新	320
2.9.9	プロパティ値に対する操作一覧の取得	323
2.9.10	プロパティグループ一覧の取得	324
2.10	サービスグループ関連の API	327
2.10.1	サービスグループ一覧の取得	327
2.10.2	サービスグループ情報の取得	329
2.10.3	サービスグループに対する操作一覧の取得	330
2.11	タグ関連の API	333
2.11.1	タググループ一覧の取得	333
2.11.2	タグ一覧の取得	335
2.12	情報管理の API	339
2.12.1	ユーザー情報の取得	339
2.12.2	バージョン情報の取得	340
2.13	API の使用例	343

## 付録 349

付録 A 参考情報 350

付録 A.1 各バージョンの変更内容 350

## 索引 367

# 1

## コマンド

JP1/AO で使用できるコマンドについて説明します。

## 1.1 コマンド一覧

JP1/AO で使用できるコマンドの一覧を次に示します。

表 1-1 構築関連のコマンド

コマンド名	機能	参照先
encryptpassword (パスワードファイルの作成)	コマンドの引数に指定するパスワードファイルを作成します。	1.5.1 encryptpassword (パスワードファイルの作成)
hcnds64checkauth (外部認証サーバとの接続確認)	外部認証サーバと連携する場合に、外部認証サーバ連携コンフィグファイルの設定および外部認証サーバとの接続を確認します。	1.5.2 hcnds64checkauth (外部認証サーバとの接続確認)
hcnds64fwcancel (Windows ファイアウォールの例外登録)	Windows ファイアウォールによって JP1/AO サーバと Web ブラウザーの間の通信が遮断されないように例外登録します。JP1/AO と Web ブラウザー間のポート番号を変更する場合に使用します。	1.5.3 hcnds64fwcancel (Windows ファイアウォールの例外登録)
hcnds64intg (認証データの削除・確認)	ユーザーアカウントを管理しているサーバのリポジトリに登録されている認証データを削除します。また、認証データが登録されているサーバのアドレスを表示します。JP1/AO のアンインストール時に認証データを削除できなかった場合に、このコマンドを使用して認証データを削除します。	1.5.4 hcnds64intg (認証データの削除・確認)
hcnds64ldapuser (LDAP 検索用ユーザー登録・削除)	Active Directory と連携する場合、Active Directory の登録情報検索に必要なユーザー情報を登録します。また、登録したユーザーを削除できます。	1.5.5 hcnds64ldapuser (LDAP 検索用ユーザー登録・削除)
hcnds64ssltool (秘密鍵・自己署名証明書の作成)	SSL 接続に必要な秘密鍵、CSR、自己署名証明書、および自己署名証明書の内容ファイルを作成します。	1.5.6 hcnds64ssltool (秘密鍵・自己署名証明書の作成)
setupcluster (クラスタ環境の構築)	JP1/AO をクラスタ環境に構築します。	1.5.7 setupcluster (クラスタ環境の構築)

表 1-2 運用関連のコマンド

コマンド名	機能	参照先
deleteremoteconnection (エージェントレス接続先定義の削除)	JP1/AO に登録されているエージェントレス接続先定義を削除します。	1.6.1 deleteremoteconnection (エージェントレス接続先定義の削除)
deleteservicetemplate (サービステンプレートの削除)	JP1/AO に登録されているサービステンプレートを削除します。	1.6.2 deleteservicetemplate (サービステンプレートの削除)
hcnds64chgurl (URL 情報の更新)	共通コンポーネントのリポジトリに格納されている、アプリケーションの起動に使用するアクセス情報 (URL 情報) を更新します。JP1/AO の運用を開始したあと、システム構成を変更した場合に使用します。	1.6.3 hcnds64chgurl (URL 情報の更新)

コマンド名	機能	参照先
hcnds64srv (JP1/AO の開始・停止・状態表示)	JP1/AO のサービスとデータベースを開始および停止します。また、JP1/AO のサービスの状態を表示します。	1.6.4 hcnds64srv (JP1/AO の開始・停止・状態表示)
hcnds64unlockaccount (ユーザーアカウントのロック解除)	ユーザーアカウントのロックを解除します。すべてのユーザーアカウントがロックされて JP1/AO にログインできない場合に使用します。	1.6.5 hcnds64unlockaccount (ユーザーアカウントのロック解除)
importservicetemplate (サービステンプレートのインポート)	サービステンプレートを JP1/AO に追加します。	1.6.6 importservicetemplate (サービステンプレートのインポート)
listremoteconnections (エージェントレス接続先定義の一覧出力)	JP1/AO に登録されているエージェントレス接続先定義の一覧を CSV 形式で出力します。	1.6.7 listremoteconnections (エージェントレス接続先定義の一覧出力)
listservices (サービス・サービステンプレートの一覧出力)	JP1/AO に登録されているサービスまたはサービステンプレートの一覧を CSV 形式で出力します。	1.6.8 listservices (サービス・サービステンプレートの一覧出力)
listtasks (タスクの一覧出力, タスク詳細情報出力)	タスクまたは履歴の一覧を CSV 形式で出力します。 タスク詳細情報を、指定したフォルダに出力します。	1.6.9 listtasks (タスクの一覧出力, タスク詳細情報出力)
setremoteconnection (エージェントレス接続先定義の追加・更新)	エージェントレス接続先定義情報ファイル (CSV 形式) を使用して、エージェントレス接続先定義を登録または更新します。	1.6.10 setremoteconnection (エージェントレス接続先定義の追加・更新)
stoptask (タスクの実行停止)	タスク ID を指定することで、指定したタスクを実行停止します。	1.6.11 stoptask (タスクの実行停止)
submittask (サービスの実行, タスクの一括再登録)	実行するサービス名やプロパティ値を指定して、サービスを実行します。 listtasks コマンドで出力したタスク詳細情報を元に、指定日時実行タスクおよび定期実行タスクを一括で再登録します。	1.6.12 submittask (サービスの実行, タスクの一括再登録)

表 1-3 保守・メンテナンス関連のコマンド

コマンド名	機能	参照先
backupsystem (JP1/AO システムのバックアップ)	JP1/AO の設定情報やデータベース情報をバックアップして指定したフォルダにデータを格納します。	1.7.1 backupsystem (JP1/AO システムのバックアップ)
hcnds64dbrepair (データベースの再作成)	データベースを強制削除して再作成し、バックアップデータからデータベースを回復します。データベースが破損して restoresystem コマンド、および hcnds64dbtrans コマンドの import オプションを使用しても復旧できない場合に、このコマンドを使用します。	1.7.2 hcnds64dbrepair (データベースの再作成)
hcnds64dsrv (データベースの開始・停止)	JP1/AO のデータベースを開始および停止します。データベースのメンテナンスを実施するときに使用します。	1.7.3 hcnds64dsrv (データベースの開始・停止)



コマンド名	機能	参照先
hcnds64dbtrans (データベースのバックアップ・リストア)	JP1/AO のデータベースをバックアップおよびリストアします。JP1/AO のデータベースを再編成する場合に使用します。	1.7.4 hcnds64dbtrans (データベースのバックアップ・リストア)
hcnds64getlogs (ログ情報の採取)	JP1/AO の運用中に出力されるログ情報を採取してアーカイブファイルに出力します。	1.7.5 hcnds64getlogs (ログ情報の採取)
restoresystem (JP1/AO システムのリストア)	backupsystem コマンドで取得した JP1/AO の設定情報やデータベース情報のバックアップデータをリストアします。	1.7.6 restoresystem (JP1/AO システムのリストア)

## 1.2 コマンド使用時の注意事項

---

コマンド使用時の注意事項を次に示します。

- Windows Server のホストで、Administrators 権限が必要なコマンドを実行する場合は、コマンドプロンプトを管理者として起動する必要があります。Windows の [スタート] メニューの [コマンドプロンプト] を右クリックし、[管理者として実行] を選択すると、管理者として起動できます。なお、UAC 機能が無効の場合、コマンドプロンプトを管理者として起動する必要はありません。
- JP1/AO サーバの OS が Windows の場合、コマンドプロンプトで「簡易編集モード」を有効にして画面上でマウスをクリックすると、簡易編集モードを解除するまで画面出力が停止します。このため、「簡易編集モード」を使用しないことを推奨します。
- クラスタ環境でコマンドを使用する場合は、実行系のホストで実行してください。ただし、hcnds64get logs コマンドは、待機系のホストでも実行できます。
- コマンドの実行を中断したい場合は、[Ctrl] + [C] キーを押してください。コマンドの実行を中断した場合は、中断時のメッセージを確認して問題が発生していないか確認してください。再実行したい場合は、再度コマンドを実行してください。
- コマンドの実行中に [Ctrl] + [S] キーを押さないでください。[Ctrl] + [S] キーを押すと、コマンドの出力が停止します。
- stoptask コマンドおよびsubmittask コマンド以外のコマンドは、JP1/AO のほかのコマンドと同時に実行できません。
- コマンドの種類によっては、コマンドを実行した直後に割り込み操作をした場合、「2 (コマンドの実行を中断しました)」以外の戻り値が返るときがあります。
- stoptask コマンド、submittask コマンドはそれぞれ同時に 10 個まで実行できます。それを超えて実行した場合、次のメッセージが表示されタスクは実行されません。  
KNAE03236-E コマンドの同時実行数を超過しました。実行中のコマンドの終了を待って、再実行してください。
- 次のコマンドを実行する場合、ユーザー設定プロパティファイルの設定を変更することで、監査ログに出力されるサブジェクト識別情報を変更できます。
  - deleteremoteconnection コマンド
  - deleteservicetemplate コマンド
  - importservicetemplate コマンド
  - listremoteconnections コマンド
  - listservices コマンド
  - listtasks コマンド
  - setremoteconnection コマンド
  - stoptask コマンド
  - submittask コマンド

- JP1/AO サーバの OS が Linux の場合、標準設定でコアファイル出力時の最大サイズが「0」になっていると、コアダンプは出力されません。障害時にコアダンプを出力するには、ulimit コマンドを実行して、各コマンドを実行する前に最大サイズを無制限(unlimited)に設定してください。
- JP1/AO サーバの OS が Windows の場合、指定したファイルパスの大文字・小文字は区別されません。JP1/AO サーバの OS が Linux の場合、指定したファイルパスの大文字・小文字は区別されます。

---

## 関連項目

- マニュアル「JP1/AO 構築ガイド」－「ユーザー設定プロパティファイル (config\_user.properties)」のトピック
-

## 1.3 コマンドの引数に指定できる文字

---

コマンドの引数に指定できる文字を次に示します。

- コマンドの引数の指定方法は、OS のコマンドプロンプトやシェルの仕様に従います。そのため、引数の値に空白や特殊記号を含む場合、「"」で囲むなどしてエスケープする必要があります。
- 各コマンドの引数でパスを指定するときに使用できる文字の種類は次のとおりです。  
半角英数字, 「\_」, 「.」, 「-」, 「 」, 「(」, 「)」, 「#」, 「@」, 「:」, 「¥」  
ただし、submittask コマンドの propertyfile オプションで指定するパスについては、制限はありません。  
なお、JP1/AO サーバの OS が Linux の場合は、フォルダ区切りとしてだけ「/」も使用できます。
- JP1/AO サーバの OS が Windows の場合、「:」はドライブ区切りとしてだけ使用できます。
- JP1/AO サーバの OS が Windows の場合、「¥」はフォルダ区切りとしてだけ使用できます。
- 引数でパスを指定する場合、UNC 形式のパスは指定できません。
- 引数でパスを指定する場合、フォルダ名の先頭または末尾が空白のパスは指定できません。空白だけのフォルダ名の場合も同様です。
- 引数でパスを指定する場合、フォルダ名の先頭または末尾が「.」のパスは指定できません。ピリオドだけのフォルダ名の場合も同様です。
- 特に断りがない場合、パスの長さは絶対パスで 1~230 文字で指定できます。
- 特に断りがない場合、各コマンドの引数は大文字・小文字が区別されます。
- 次に示す名称は OS の予約語のため、ファイル名およびフォルダ名に指定しないでください。  
CON, PRN, AUX, CLOCK\$, NUL, COM0, COM1, COM2, COM3, COM4, COM5,  
COM6, COM7, COM8, COM9, LPT0, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7,  
LPT8, LPT9

## 1.4 コマンドの記述形式

---

コマンドの記述形式について説明します。

各コマンドで説明する項目は次のとおりです。ただし、コマンドによっては説明しない項目もあります。

### 機能

コマンドの機能について説明しています。

### 形式

コマンドの指定形式を次の形式で説明しています。

コマンド名 [/オプション [値]] …]

/オプションと値の組み合わせを「オプション」と呼びます。各コマンドのオプションをまとめて「引数」と呼びます。

### 引数

コマンドの引数について説明しています。

なお、JP1/AO サーバの OS が Linux の場合、「/」は「-」に読み替えてください。

### 格納先

コマンドの格納場所について説明しています。

なお、JP1/AO サーバの OS が Linux の場合、「¥」は「/」に読み替えてください。

### 実行権限

コマンドの実行に必要なユーザーの権限について説明しています。

### 注意事項

コマンド使用時の注意事項を説明しています。

### 戻り値

コマンドの戻り値について説明しています。

なお、コマンド実行時に表示されるメッセージについては、マニュアル「[JP1/AO メッセージ](#)」を参照してください。

コマンドによっては監査ログを出力します。監査ログを出力するコマンド、監査の対象となる操作、および出力されるメッセージ ID については、マニュアル「[JP1/AO 運用ガイド](#)」の「[監査ログに出力される事象の種別](#)」のトピックを参照してください。

### 使用例

コマンドの使用例について説明しています。

## 1.5 構築関連のコマンド

---

### 1.5.1 encryptpassword (パスワードファイルの作成)

#### 機能

JP1/AO のコマンドの引数に指定するパスワードファイルを作成します。

JP1/AO に登録済みのユーザーのユーザー ID、パスワード、および作成するパスワードファイルのパスを引数に指定してコマンドを実行すると、暗号化されたパスワードファイルを作成できます。

各コマンドの実行時に、パスワードを指定する代わりに、作成したパスワードファイルを指定することで、パスワードの指定が不要になります。

#### 形式

```
encryptpassword
  /user ユーザーID
  /password パスワード
  /passwordfile パスワードファイルパス
```

#### 引数

`/user ユーザーID`

パスワードファイルを作成したい JP1/AO のユーザー ID を指定します。

指定できる文字数は、1～256 文字です。

使用できる文字の種類は、半角英数字、[!]、[#]、[\$]、[%]、[&]、[']、[(, ()], [\*]、[+]、[-]、[.]、[=]、[@]、[¥]、[^]、[\_]、[|] です。

大文字と小文字は区別しません。

`/password パスワード`

`user` オプションで指定した、ユーザーのパスワードを指定します。

指定できる文字数は、1～256 文字です。

使用できる文字の種類は、`user` オプションと同じです。

`/passwordfile パスワードファイルパス`

作成するパスワードファイルのパスを絶対パスまたは相対パスで指定します。指定したパスが存在する場合はエラーとなります。

#### 格納先

Windows の場合

`JP1/AO のインストール先フォルダ¥bin`

Linux の場合

/opt/jplao/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	排他エラーが発生しました。
5	通信に失敗しました。
6	認証に失敗しました（指定された値が不正です）。
7	パスの指定が不正です。
8	出力パスが存在しています。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
14	コマンドの実行権限がありません。
200	パスワードファイルの作成に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、指定したユーザーのパスワードファイルを作成する場合のコマンドの使用例を示します。

```
encryptpassword /user user01 /password pass01 /passwordfile passfile
```

## 関連項目

- 1.3 コマンドの引数に指定できる文字

## 1.5.2 hcnds64checkauth (外部認証サーバとの接続確認)

### 機能

外部認証サーバと連携する場合に、外部認証サーバ連携コンフィグファイルの設定および外部認証サーバとの接続を確認します。

JP1/AO では外部認証サーバとして JP1/Base または Active Directory と連携できます。

このコマンドを実行すると次の項目がチェックされます。

- 外部認証サーバ連携コンフィグファイル (exauth.properties) の、外部認証サーバ連携時に共通で使用するキーの値
- 外部認証サーバ連携コンフィグファイル (exauth.properties) の auth.server.type キーに値が正しく設定されているかどうか

JP1/Base の認証機能と連携する場合は auth.server.type キーに「jplbase」を、Active Directory と連携する場合は「ldap」を指定してください。大文字と小文字は区別されます。auth.server.type キーのデフォルトを示す「internal」が指定された場合には、外部認証サーバの設定が有効になっていないことを示すエラーメッセージが表示されます。

- JP1/Base の認証機能と連携する場合、次の項目がチェックされる
  - JP1/Base と共通コンポーネントが同一ホストに存在するかどうか
  - JP1/Base がサポートしているバージョンかどうか
  - JP1/Base のユーザーが正しく認証できるかどうか
- Active Directory と連携する場合、次の項目がチェックされる
  - 外部認証サーバ連携コンフィグファイル (exauth.properties) の、Active Directory 連携時に使用するキーの値
  - Active Directory と接続できるかどうか
  - Active Directory と接続できる場合、グループ検索は可能か

### 形式

```
hcnds64checkauth
  /user ユーザー名
  /pass パスワード
  [/summary]
```

### 引数

/user ユーザー名

外部認証サーバに登録済みのユーザー名を指定します。ただし、JP1/Base の認証機能と連携する場合には、JP1/AO に登録済みのユーザー名と重複しないユーザー名を指定してください。



## /pass パスワード

外部認証サーバに登録済みのユーザー名に対応するパスワードを指定します。

## /summary

コマンド実行時に表示される確認メッセージを簡略化します。このオプションを指定した場合、表示されるメッセージは、処理フェーズごとの成否を示すメッセージ、エラーメッセージ、および結果を示すメッセージだけとなります。ただし、結果を示すメッセージと同様のエラーメッセージがある場合、途中のエラーメッセージが省略されて、結果を示すメッセージだけが表示されます。

## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 戻り値

戻り値	説明
0	正常に終了しました。
1~99	シンタックスエラーの総数です。
100	シンタックスエラーの行数が 100 を超えた場合の終了コードです。
101~199	接続エラーまたは認証エラーが発生しました。 1 の位：接続エラーの数 10 の位：認証エラーの数 それぞれの位の最大は 9 です。9 以上発生しても 9 のままとなります。
247	user オプションに指定したユーザー ID が、JP1/AO に登録されているユーザー ID と重複しているため、認証できません。重複していないユーザー ID を指定してください。
248	JP1/Base が、このコマンドを実行したホストと同一ホスト上にインストールされていません。
249	サポートしていないバージョンの JP1/Base が使用されています。
250	セカンダリーサーバからコマンドが実行されました。
252	定義ファイルの共通項目の設定が不正です。
253	外部認証連携が設定されていません。
254	引数が不正です。
255	異常終了しました。

## 使用例

Windows で、外部認証サーバとの接続を確認する場合のコマンドの使用例を示します。

```
hcnds64checkauth /user test01 /pass TTdate00 /summary
```

---

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
- 

## 1.5.3 hcnds64fwcancel (Windows ファイアウォールの例外登録)

### 機能

Windows ファイアウォールによって JP1/AO サーバと Web ブラウザーの間の通信が遮断されないように例外登録します。JP1/AO と Web ブラウザー間のポート番号をデフォルト値から変更する場合に使用します。

### 形式

```
hcnds64fwcancel
```

### 格納先

共通コンポーネントのインストール先フォルダ¥bin

### 実行権限

Administrators 権限を持つユーザーで実行してください。

### 戻り値

このコマンドには戻り値がありません。このため、処理が正常に終了したかどうかは、Windows ファイアウォールに正しく例外登録されているかどうかで確認してください。

Windows ファイアウォールは、Windows の [コントロールパネル] の [Windows ファイアウォール] で確認できます。

## 1.5.4 hcnds64intg (認証データの削除・確認)

### 機能

ユーザーアカウントを管理しているサーバのリポジトリに登録されている認証データを削除します。また、認証データが登録されているサーバのアドレスを表示します。

JP1/AO のアンインストール時に認証データを削除できなかった場合に、このコマンドを使用して認証データを削除します。

### 形式

```
hcnds64intg
  {/delete /type Automation | /print | /primary }
  /user ユーザーID
  /pass パスワード
```

### 引数

`/delete`

認証データを削除します。

`/type Automation`

認証データが登録されているサーバの製品名として「Automation」を指定します。

`/print`

認証データが登録されているプログラム名称を表示します。

`/primary`

認証データが登録されているサーバのホスト名、または IP アドレスを表示します。

`/user ユーザーID`

認証データが登録されているサーバに接続するためのユーザー ID を指定します。User Management 権限を持つアカウントのユーザー ID を指定します。

`/pass パスワード`

User Management 権限を持つアカウントのパスワードを指定します。

### 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	すでに、認証データが削除されています。
2	コマンドを実行したサーバに、認証データが登録されています。
3	コマンドを実行したサーバに、認証データが登録されていません。
4	コマンドを実行したサーバに、認証データが登録されていません。また、認証データが登録されているサーバで認証エラーが発生しました。
253	認証データが登録されているサーバで認証エラーが発生しました。
254	認証データが登録されているサーバとの通信に失敗しました。
255	異常終了しました。

## 使用例

Windows で、ユーザーアカウントを管理しているサーバから認証データを削除する場合のコマンドの使用例を示します。

```
hcnds64intg /delete /type Automation /user user1 /pass pass1
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)

## 1.5.5 hcnds64ldapuser (LDAP 検索用ユーザー登録・削除)

### 機能

Active Directory と連携する場合、Active Directory の登録情報検索に必要なユーザー情報を登録します。また、登録したユーザー情報を削除できます。

このコマンドでユーザー情報を登録した場合は、引き続きhcnds64checkauth コマンドを実行して正しく認証できるかを確認してください。

## 形式

```
hcnds64ldapuser
  {/set /dn ユーザー識別子 /pass パスワード | /delete}
  /name {サーバ識別名 | ドメイン名}
  | /list
```

## 引数

`/set`

ユーザー情報を登録します。

`/dn ユーザー識別子`

登録するユーザーのユーザー識別子を指定します。使用可能な文字は RFC 4514 に従ってください。

「&」「|」「^」「(」「)」「<」「>」は「"」で囲むか、「^」でエスケープしてください。

末尾に「¥」があり、値として指定したい場合は、「¥」でエスケープしてください。

`/pass パスワード`

dn オプションで指定した、ユーザーのパスワードを指定します。

`/delete`

登録されているユーザー情報を削除します。name オプションで指定したサーバ識別名またはドメイン名を含むユーザーの情報を削除します。

`/name {サーバ識別名 | ドメイン名}`

ユーザー情報の登録をする場合、登録先になるサーバ識別名またはドメイン名を指定します。

ユーザー情報の削除をする場合、削除するユーザーの登録されているサーバ識別名またはドメイン名を指定します。

ただし、Active Directory とのグループ連携が無効で、LDAP 検索用ユーザーが登録されている場合、ドメイン名は指定できません。サーバ識別名を指定してください。

`/list`

登録したユーザー情報に含まれている、サーバ識別名とドメイン名の一覧を表示します。

## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	引数に指定できない文字が含まれています。
3	登録された情報が見つかりません。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

- Windows で、ユーザー情報を登録する場合  
ユーザー名 [suzuki]、所属グループ [Users]、サーバドメイン名 [example.com]、パスワード [qweasd00] のユーザー情報を登録する。  

```
hcms64ldapuser /set /dn "CN=suzuki,CN=Users,DC=Example,DC=com" /pass qweasd00 /name example.com
```
- Windows で、ユーザー情報を削除する場合  
サーバ名 [tokyo] のユーザーの情報を削除する。  

```
hcms64ldapuser /delete /name tokyo
```
- Windows で、登録されているサーバ識別名とドメイン名の一覧を表示する場合  

```
hcms64ldapuser /list
```

## 出力例

登録されているサーバ識別名とドメイン名の一覧を出力した場合の例

```
[ServerName]  
tokyo  
osaka  
kyoto
```

---

## 関連項目

- 1.3 コマンドの引数に指定できる文字
-

## 1.5.6 hcmds64ssltool (秘密鍵・自己署名証明書の作成)

### 機能

SSL 接続に必要な秘密鍵, CSR, 自己署名証明書および自己署名証明書の内容ファイルを作成します。作成したファイルは, 次の用途で使用します。

- CSR を CA に提出して SSL サーバ証明書を取得します。取得した SSL サーバ証明書と秘密鍵を組み合わせ、SSL 接続環境を構築できます。
- 自己署名証明書と秘密鍵を組み合わせ、SSL 接続環境を構築できます。ただし、セキュリティ強度が低い場合、テスト目的での利用をお勧めします。
- 自己署名証明書の内容ファイルで、自己署名証明書の登録内容が確認できます。

### 形式

```
hcmds64ssltool
  [/key 秘密鍵ファイル名]
  [/csr CSRファイル名]
  [/cert 自己署名証明書ファイル名]
  [/certtext 自己署名証明書の内容ファイル名]
  [/validity 自己署名証明書有効期限 /dname 識別名 (DN) /sigalg 署名アルゴリズム]
```

### 引数

#### /key 秘密鍵ファイル名

秘密鍵を格納するパスを絶対パスで指定します。絶対パスには、秘密鍵のファイル名を含めてください。このオプションを省略した場合、デフォルトの出力先パスに「httpsdkey.pem」ファイルおよび「ecc-httpsdkey.pem」ファイルが出力されます。

#### /csr CSRファイル名

CSR を格納するパスを絶対パスで指定します。絶対パスには、CSR のファイル名を含めてください。このオプションを省略した場合、デフォルトの出力先パスに「httpsd.csr」ファイルおよび「ecc-httpsd.csr」ファイルが出力されます。

#### /cert 自己署名証明書ファイル名

自己署名証明書を格納するパスを絶対パスで指定します。絶対パスには、自己署名証明書のファイル名を含めてください。

このオプションを省略した場合、デフォルトの出力先パスに「httpsd.pem」ファイルおよび「ecc-httpsd.pem」ファイルが出力されます。

#### /certtext 自己署名証明書の内容ファイル名

自己署名証明書の内容をテキスト形式で出力します。そのファイルを格納するパスを絶対パスで指定します。絶対パスには、テキストファイル名を含めてください。

このオプションを省略した場合、デフォルトの出力先パスに「httpsd.txt」ファイルおよび「ecc-httpsd.txt」ファイルが出力されます。

#### /validity 自己署名証明書有効期限

自己署名証明書の有効期限を日数で指定します。このオプションを省略した場合、有効期限は3,650日となります。指定できる値は、9999年12月31日までの日数です。

#### /dname 識別名 (DN)

SSL サーバ証明書に記述する識別名 (DN) を「属性型=属性値」の形式で指定します。「,」で区切ることで、複数の属性型の値を指定できます。属性型は、大文字、小文字を区別しません。属性値に「"」 「¥」は使用できません。

文字のエスケープは RFC 2253 に従ってください。

次の文字は「¥」でエスケープしてください。

- 「+」 「,」 「;」 「<」 「=」 「>」
- 文字列の先頭の空白
- 文字列の末尾の空白
- 文字列の先頭の「#」

このオプションを省略した場合、コマンド実行時に画面に従って属性値を応答入力します。

このオプションに指定できる属性型について、次の表に示します。

表 1-4 識別名 (DN) に指定できる属性型一覧

属性型	属性型の意味	応答入力時の画面の表記	属性値
CN	Common Name	Server Name	ホスト名、IP アドレス、ドメイン名など JP1/AO サーバの識別名*
OU	Organizational Unit Name	Organizational Unit	部門、部署名など小さな単位の組織名
O	Organization Name	Organization Name	会社、団体の組織名*
L	Locality Name	City or Locality	都市または地域名（日本では市町村名）
ST	State or Province Name	State or Province	州名または地方名（日本では都道府県名）
C	Country Name	two-character country-code	国コード（日本は「JP」）

注※

応答入力では必ず入力してください。

応答入力例を次に示します。

```
Enter Server Name [default=MyHostname]:example.com
Enter Organizational Unit:Device Manager Administration
Enter Organization Name [default=MyHostname]:HITACHI
Enter your City or Locality:Yokohama
Enter your State or Province:Kanagawa
```



```
Enter your two-character country-code:JP
Is CN=example.com,OU=Device Manager Administration,O=HITACHI,L=Yokohama,
ST=Kanagawa,C=JP correct? (y/n) [default=n]:y
```

入力に誤りがあった場合は、確認で「n」を入力すると再度応答入力が可能です。

## /sigalg 署名アルゴリズム

署名アルゴリズムとして、次のどれかを指定してください。このオプションを省略した場合は、「SHA256withRSA」となります。

- SHA1withRSA
- SHA256withRSA

## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 注意事項

- SSL サーバ証明書の属性型「CN」と、Web ブラウザーから JP1/AO サーバの接続先として指定したホスト名、IP アドレスまたはドメイン名が一致しない場合、サーバ名の不一致の警告またはエラーが発生します。
- key, csr, cert または certtext オプションを省略してコマンドを実行した場合、各ファイルは次の場所に出力されます。

Windows の場合

共通コンポーネントのインストール先フォルダ¥uCPSB¥httpsd¥conf¥ssl¥server

Linux の場合

共通コンポーネントのインストール先ディレクトリ/Base64/uCPSB/httpsd/conf/ssl/server

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。

戻り値	説明
250	キーストアの削除に失敗しました。
251	秘密鍵の作成に失敗しました。
252	自己署名証明書の作成に失敗しました。
253	CSR の作成に失敗しました。
254	自己署名証明書の内容ファイルの作成に失敗しました。
255	異常終了しました。

---

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
- 

## 1.5.7 setupcluster (クラスタ環境の構築)

### 機能

JP1/AO をクラスタ環境に構築します。実行系ホストと待機系ホストでそれぞれコマンドを実行する必要があります。

データベースおよびデータの退避先となる共有ディスク上のパスを指定してコマンドを実行すると、クラスタ環境を構築できます。

コマンドの実行後は、途中経過を示すメッセージが表示されます。

### 形式

```
setupcluster  
/exportpath データベースおよびデータの退避先パス
```

### 引数

*/exportpath データベースおよびデータの退避先パス*

このコマンドを実行するホストのデータベースおよびデータを退避させるフォルダを絶対パスまたは相対パスで指定します。十分に空き容量のある、共有ディスク上のフォルダを指定します。指定できるパスの長さは 49 文字までです。

### 格納先

Windows の場合

*JP1/AO のインストール先フォルダ¥bin*

Linux の場合

/opt/jplao/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
7	パスの指定が不正です。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
11	指定したフォルダが空ではありません。
14	コマンドの実行権限がありません。
120	クラスタのセットアップに失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、実行系ホストで指定した共有フォルダのパスに、データベースの再作成およびデータの回避を行い、クラスタ環境を構築する場合のコマンドの使用例を示します（待機系ホストの場合も同じコマンドを実行します）。

```
setupcluster /exportpath Z:¥share
```

## 関連項目

- 1.3 コマンドの引数に指定できる文字

## 1.6 運用関連のコマンド

### 1.6.1 deleteremoteconnection (エージェントレス接続先定義の削除)

#### 機能

JP1/AO に登録されているエージェントレス接続先定義を削除します。

削除したいエージェントレス接続先定義の ID を指定してコマンドを実行すると、指定したエージェントレス接続先定義を削除できます。

#### 形式

```
deleteremoteconnection
  /id エージェントレス接続先定義ID
  /user ユーザーID
  {/password パスワード | /passwordfile パスワードファイルパス}
```

#### 引数

/id

削除するエージェントレス接続先定義の ID を指定します。なお、事前に `listremoteconnections` コマンドを実行して、エージェントレス接続先定義の ID を確認しておく必要があります。

指定できる文字数は、1~64 文字です。

使用できる文字の種類は、半角数字です。

/user

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字、`[!]`、`[#]`、`[$]`、`[%]`、`[&]`、`[']`、`[(,)]`、`[*]`、`[+]`、`[-]`、`[.]`、`[=]`、`[@]`、`[¥]`、`[^]`、`[_]`、`[|]` です。

大文字と小文字は区別しません。

/password

`user` オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたは `passwordfile` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、`user` オプションと同じです。

/passwordfile

user オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルは`encryptpassword` コマンドで作成します。

このオプションまたは`password` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 格納先

Windows の場合

`JP1/AO` のインストール先フォルダ¥bin

Linux の場合

`/opt/jplao/bin`

## 実行権限

OS の Administrators 権限または root 権限と、JP1/AO の Admin ロールを持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 注意事項

このコマンドで削除できるエージェントレス接続先定義は、1 回の実行につき 1 件です。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
14	コマンドの実行権限がありません。
240	エージェントレス接続先定義の削除に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、指定したエージェントレス接続先定義（エージェントレス接続先定義 ID が 12345）を削除する場合のコマンドの使用例を示します。

```
deleteremoteconnection /id 12345 /user user01 /password pass01
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)

## 1.6.2 deleteservicetemplate（サービステンプレートの削除）

### 機能

JP1/AO に登録されているサービステンプレートを削除します。

削除したいサービステンプレートのサービステンプレート ID、ベンダー ID、バージョン番号を指定してコマンドを実行すると、指定したサービステンプレートを削除できます。

### 形式

```
deleteservicetemplate  
  /name サービステンプレートID  
  /vendor サービステンプレートのベンダーID  
  /version サービステンプレートのバージョン番号  
  /user ユーザーID  
  {/password パスワード | /passwordfile パスワードファイルパス}
```

### 引数

**/name** サービステンプレートID

削除するサービステンプレートの ID を指定します。

大文字と小文字は区別しません。

指定できる文字数は、1～64 文字です。

使用できる文字の種類は、半角英数字、[-], [\_], [.] です。

**/vendor** サービステンプレートのベンダーID

削除するサービステンプレートのベンダー ID を指定します。

大文字と小文字は区別しません。

指定できる文字数は、1～64 文字です。

使用できる文字の種類は、半角英数字、[-], [\_], [.] です。

### `/version` サービステンプレートのバージョン番号

削除するサービステンプレートのバージョン番号を「*XX.YY.ZZ*」の形式で指定します。

*XX*, *YY*, *ZZ* に指定できる文字は、00 から 99 の 2 桁の半角数字です。

- *XX*: メジャーバージョン番号
- *YY*: マイナーバージョン番号
- *ZZ*: リビジョン番号

### `/user` ユーザーID

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字、`!`, `#`, `$`, `%`, `&`, `'`, `(`, `)`, `*`, `+`, `-`, `.`, `=`, `@`, `¥`, `^`, `_`, `|` です。

大文字と小文字は区別しません。

### `/password` パスワード

`user` オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたは `passwordfile` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、`user` オプションと同じです。

### `/passwordfile` パスワードファイルパス

`user` オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルは `encryptpassword` コマンドで作成します。

このオプションまたは `password` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 格納先

Windows の場合

`JP1/AO` のインストール先フォルダ¥bin

Linux の場合

`/opt/jplao/bin`

## 実行権限

OS の Administrators 権限または root 権限と、JP1/AO の Admin ロールまたは Develop ロールの両方を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 注意事項

リリースサービステンプレートは、このコマンドを使用して削除してください。なお、開発サービステンプレートは [エディタ] 画面から削除してください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
9	指定したパスが存在しません。
14	コマンドの実行権限がありません。
190	サービステンプレートの削除に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、指定したサービステンプレート（サービステンプレート ID が nameA、ベンダー ID が vendorB、バージョン番号が 01.00.00）を削除する場合のコマンドの使用例を示します。

```
deleteservicetemplate /name nameA /vendor vendorB /version 01.00.00 /user user01 /password pass01
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
- マニュアル「JP1/AO サービステンプレート開発ガイド」 - 「開発サービステンプレートを削除する手順」のトピック



## 1.6.3 hcnds64chgurl (URL 情報の更新)

### 機能

共通コンポーネントのリポジトリに格納されている、アプリケーションの起動に使用するアクセス情報 (URL 情報) を更新します。

JP1/AO の運用を開始したあと、次のどれかの構成変更を行った場合に使用します。

- 共通コンポーネントがインストールされているホストが使用するポートを変更した場合
- 共通コンポーネントがインストールされているホストのホスト名または IP アドレスを変更した場合

### 形式

```
hcnds64chgurl  
  {/list |  
   /change 変更前のURL 変更後のURL |  
   /change 変更後のURL /type Automation}
```

### 引数

`/list`

現在設定されている URL と製品名のリストを表示します。

`/change 変更前のURL 変更後のURL`

現在登録されている URL に関する情報を、変更後の URL に関する情報で上書きします。

現在登録されている URL と変更後の URL を指定します。 `type` オプションと組み合わせて使用する場合は、変更後の URL だけを指定します。

IPv6 アドレスを含む URL を指定する場合は、IP アドレスを [ ] で囲んでください。

`/type Automation`

URL を変更する製品名として「Automation」を指定します。

### 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

### 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	URLがありません。
253	リポジトリのリストアに失敗しました。
254	リポジトリのバックアップに失敗しました。
255	異常終了しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、現在設定されている URL と製品名のリストを表示する場合  
`hcnds64chgurl /list`
- Windows で、現在登録されている URL に関する情報を、変更後の URL で上書きする場合  
`hcnds64chgurl /change "http://192.168.11.33:22015" "http://192.168.11.55:22015"`

## 関連項目

- 1.3 コマンドの引数に指定できる文字

## 1.6.4 hcnds64srv (JP1/AO の開始・停止・状態表示)

### 機能

JP1/AO のサービスとデータベースを開始または停止します。また、JP1/AO のサービスの状態を表示したり、サービスの開始方法を変更したりできます。

このコマンドの `server` オプションに「AutomationWebService」を指定して実行すると、次のサービスを開始・停止・状態表示します。

表 1-5 開始・停止・状態表示の対象サービス一覧

サービス表示名およびプロセス	開始	停止	状態表示
HAutomation Engine Web Service	○	○	○
HBase 64 Storage Mgmt SSO Service	○	○※1	○

サービス表示名およびプロセス	開始	停止	状態表示
HBase 64 Storage Mgmt Web Service	○	○ ※1	○
HBase 64 Storage Mgmt Web SSO Service	○	○ ※1	○
データベースのプロセス※2	○	○ ※1	○

(凡例)

○：処理されます。×：処理されません。

注※1

Hitachi Command Suite 製品のサービスが開始している場合は停止されません。

注※2

JP1/AO の内部プロセスです。hcnds64srv コマンドでは、データベースのサービスを示す「HiRDB/EmbeddedEdition\_HD1」は開始および停止しません。

## 形式

```
hcnds64srv
  {/start | /stop | /check | /status}
  [/server サービス名]
```

JP1/AO および全 Hitachi Command Suite 製品のサービスの状態を確認する場合：

```
hcnds64srv
  /statusall
```

サービスの開始方法を変更する場合：

```
hcnds64srv
  /starttype {auto | manual}
  {/server サービス名 | /all}
```

## 引数

/start

server オプションで指定したサービスとデータベースを開始します。

/stop

server オプションで指定したサービスとデータベースを停止します。

/check

server オプションで指定したサービスとデータベースの状態を表示します。

/status

server オプションで指定したサービスとデータベースの状態を表示します。

## `/server サービス名`

JP1/AO 製品のサービスだけを開始・停止・状態表示する場合、サービス名には「AutomationWebService」を指定します。このオプションを省略した場合、インストールされている JP1/AO およびすべての Hitachi Command Suite 製品のサービスが対象となります。

## `/statusall`

サービスとデータベースの状態、および共通コンポーネントに登録された Hitachi Command Suite 製品のサービスの状態を表示します。

## `/starttype {auto | manual}`

`server` オプションで指定したサービスの開始種別を指定します。

自動開始の場合は `auto` を指定します。手動開始の場合は `manual` を指定します。

## `/all`

このオプションを指定した場合は、インストールされている JP1/AO およびすべての Hitachi Command Suite 製品のサービスが対象となります。

## 格納先

### Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

### Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 注意事項

- 日常運用で JP1/AO のサービスを開始・停止する場合は、`server` オプションを省略してすべてのサービスを開始・停止してください。`server` オプションを指定して JP1/AO 製品のサービスだけを開始する場合は、事前に共通コンポーネントのサービスを開始しておく必要があるため、`server` オプションに「HBase」を指定して共通コンポーネントのサービスを開始してください。
- タスクの処理中に `stop` オプションを指定した場合、エージェントレス接続先で実行中の処理は強制終了します。そのため、実行状態（実行中、応答待ち中、異常検出、停止中、または長期実行中）のタスクがある場合は、停止状態（正常終了、失敗、またはキャンセル）に遷移するまで待つか、すべてのタスクを実行停止してから使用してください。
- `stop` オプションを指定して実行したあと、3分以内に停止処理が終わらない場合は、タイムアウトを示すメッセージを出力してエラー終了します。この場合は、しばらく待ってから、再度 `stop` オプションを指定して実行してください。

## 戻り値

start オプション, stop オプション指定時のコマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	サービスは開始済みです (start オプション指定時)。 サービスは停止済みです (stop オプション指定時)。
255	失敗しました。

check オプション, status オプション, statusall オプション指定時のコマンドの戻り値を次の表に示します。

戻り値	説明
0	開始していません。
1	開始しています。
255	失敗しました。

starttype オプション指定時のコマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
255	失敗しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で, JP1/AO 製品のサービスを開始する場合  
`hcnds64srv /start /server AutomationWebService`
- Windows で, JP1/AO 製品のサービスを停止する場合  
`hcnds64srv /stop /server AutomationWebService`
- Windows で, JP1/AO 製品のサービスの状態を確認する場合  
`hcnds64srv /status /server AutomationWebService`

---

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
-

## 1.6.5 hcnds64unlockaccount (ユーザーアカウントのロック解除)

### 機能

ユーザーアカウントのロックを解除します。すべてのユーザーアカウントがロックされてJP1/AOにログインできない場合に使用します。

### 形式

```
hcnds64unlockaccount
  /user ユーザーID
  /pass パスワード
```

### 引数

`/user ユーザーID`

ロックを解除するユーザーアカウントのユーザー ID を指定します。User Management 権限を持つユーザー ID を指定します。

`/pass パスワード`

ロックを解除するユーザーアカウントのパスワードを指定します。

### 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

### 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

### 注意事項

- hcnds64unlockaccount コマンドでロックを解除できるのは、User Management 権限があるユーザーアカウントだけです。
- オプションに指定するユーザー名またはパスワードに「&」、「|」、または「^」を含む場合、「"」で囲むか、「^」でエスケープします。例えば、Windows で、パスワードが「^a^b^c^」の場合、「hcnds64unlockaccount /user system /pass ^"a"^"b"^"c"^」, または「hcnds64unlockaccount /user system /pass ^a^b^c^」と記述します。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
251	認証エラー（ログインに失敗）です。
252	認証エラー（User Management 権限がない）です。
253	認証サーバとの通信に失敗しました。
254	セカンダリーサーバ側で実行されました。
255	異常終了しました。

## 使用例

Windows で、指定したユーザー（ユーザー ID が test01）のロックを解除する場合のコマンドの使用例を示します。

```
hcnds64unlockaccount /user test01 /pass TTdate00
```

## 関連項目

- 1.3 コマンドの引数に指定できる文字

## 1.6.6 importservicetemplate（サービステンプレートのインポート）

### 機能

サービステンプレートを JP1/AO に追加します。JP1/AO にサービステンプレートを追加することを、サービステンプレートのインポートといいます。

1つのサービステンプレートパッケージ、または複数のサービステンプレートパッケージをまとめた zip ファイルを指定してコマンドを実行すると、指定したサービステンプレートパッケージを JP1/AO にインポートできます。

### 形式

```
importservicetemplate  
  /file サービステンプレートパッケージまたは複数のサービステンプレートパッケージをまとめたzipファイル  
  /user ユーザーID  
  {/password パスワード | /passwordfile パスワードファイルパス}
```

## 引数

*/file* サービステンプレートパッケージまたは複数のサービステンプレートパッケージをまとめたzipファイル

インポートするサービステンプレートパッケージまたは複数のサービステンプレートパッケージをまとめた zip ファイルを絶対パスまたは相対パスで指定します。

*/user* ユーザーID

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字および「!」、 「#」、 「\$」、 「%」、 「&」、 「'」、 「(」、 「)」、 「\*」、 「+」、 「-」、 「.」、 「=」、 「@」、 「¥」、 「^」、 「\_」、 「|」です。

大文字と小文字は区別しません。

*/password* パスワード

*user* オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたは*passwordfile* オプションのどちらかを指定する必要があります。両オプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、*user* オプションと同じです。

*/passwordfile* パスワードファイルパス

*user* オプションで指定した、ユーザーのパスワードが格納されているファイルのパスを絶対パスまたは相対パスで指定します。

このオプションまたは*password* オプションのどちらかを指定する必要があります。両オプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 格納先

Windows の場合

*JP1/AO* のインストール先フォルダ¥bin

Linux の場合

/opt/jplao/bin

## 実行権限

Admin ロールまたは Develop ロールかつ OS の Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。



## 注意事項

複数のサービステンプレートパッケージをまとめた zip ファイルをインポートする場合、一部のサービステンプレートパッケージのインポートに失敗してもインポートを続行します。特定のサービステンプレートパッケージのインポートに失敗したことを示すメッセージが表示されたときは、メッセージの内容を確認し、失敗の原因を取り除いた上で、インポートに失敗したサービステンプレートパッケージを再度インポートしてください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
9	指定したパスが存在しません。
14	コマンドの実行権限がありません。
180	サービステンプレートのインポートに失敗しました。
255	異常終了しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、指定したサービステンプレートパッケージ (C:¥temp¥aaa.st) を JP1/AO にインポートする場合  

```
importservicetemplate /file C:¥temp¥aaa.st /user user1 /password pass1
```
- Windows で、指定した複数のサービステンプレートパッケージをまとめた zip ファイル (C:¥temp ¥bbb.zip) を JP1/AO にインポートする場合  

```
importservicetemplate /file C:¥temp¥bbb.zip /user user1 /password pass1
```

---

## 関連項目

- 1.3 コマンドの引数に指定できる文字

- マニュアル「JP1/AO サービステンプレート開発ガイド」 – 「サービス共有プロパティを定義する場合の注意事項」のトピック

## 1.6.7 listremoteconnections (エージェントレス接続先定義の一覧出力)

### 機能

JP1/AO に登録されているエージェントレス接続先定義の一覧を CSV 形式で出力します。

### 形式

```
listremoteconnections
  /file 出力ファイルパス
  /user ユーザーID
  {/password パスワード | /passwordfile パスワードファイルパス}
```

### 引数

#### /file

エージェントレス接続先定義情報を出力するファイルのパスを絶対パスまたは相対パスで指定します。指定したファイルが存在する場合はエラーとなります。

#### /user

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字、「!」、「#」、「\$」、「%」、「&」、「'」、「(」、「)」、「\*」、「+」、「-」、「.」、「=」、「@」、「¥」、「^」、「\_」、「|」です。

大文字と小文字は区別しません。

#### /password

`user` オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたは `passwordfile` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、`user` オプションと同じです。

#### /passwordfile

`user` オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルは `encryptpassword` コマンドで作成します。

このオプションまたは `password` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 出力形式

エージェントレス接続先定義ごとに1行ずつ、出力項目がCSV形式で出力されます。

なお、出力項目の値は「"」で囲まれて出力されます。値に「"」が含まれる場合、直前に「\"」を付加してエスケープされます。

表 1-6 エージェントレス接続先定義情報ファイルの出力形式

出力項目	意味
Id	エージェントレス接続先定義の ID です。
Method	接続先種別として次のどれかが出力されます。 <ul style="list-style-type: none"><li>• IPv4</li><li>• IPv6</li><li>• HostName</li></ul>
IP Address/Host Name	接続先ホストの IP アドレスまたはホスト名です。
Service Group	エージェントレス接続先定義に割り当てられたサービスグループです。
Authentication	認証情報を設定しているかどうか出力されます。 <ul style="list-style-type: none"><li>• Enable 認証情報を設定している。</li><li>• Disable 認証情報を設定していない。</li></ul>
Protocol	接続先ホストとの通信で使用する認証用プロトコルが出力されます。 <ul style="list-style-type: none"><li>• Windows</li><li>• SSH</li><li>• Telnet</li></ul>
SSH Authentication Method <sup>※1</sup>	接続先ホストとの通信で使用する認証方式が出力されます。 <ul style="list-style-type: none"><li>• Password Authentication パスワード認証</li><li>• Public Key Authentication 公開鍵認証</li><li>• Keyboard Interactive Authentication キーボードインタラクティブ認証</li></ul>
User ID	接続先ホストにログインするためのユーザー ID です。
Password <sup>※2</sup>	接続先ホストにログインするためのパスワードです。
Super User's Password <sup>※2</sup>	接続先ホストのスーパーユーザーのパスワードです。
Connection Status	最後にホストに接続した時の状態が出力されます。 <ul style="list-style-type: none"><li>• Connection Successful 接続成功</li><li>• Error 接続失敗</li></ul>

出力項目	意味
Connection Status	<ul style="list-style-type: none"> <li>Unknown 未接続</li> <li>- 対象外</li> </ul> 接続先ホストを範囲指定している場合は、「-」が表示されます。
Connected Time	最後にホストに接続した時刻が出力されます。

#### 注※1

「Protocol」が「Windows」または「Telnet」の場合は、空文字が出力されます。

#### 注※2

パスワードが設定されているかどうかに関係なく、「\*\*\*\*\*」が出力されます。

## 格納先

Windows の場合

*JP1/AO* のインストール先フォルダ¥bin

Linux の場合

/opt/jplao/bin

## 実行権限

OS の Administrators 権限または root 権限と、JP1/AO の Admin ロールを持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
8	すでに出力先パスに同名のファイルが存在します。

戻り値	説明
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
13	ファイル出力に失敗しました。
14	コマンドの実行権限がありません。
220	エージェントレス接続先定義一覧の取得に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、エージェントレス接続先定義情報を「C:¥temp¥list01.csv」として出力する場合のコマンドの使用例を示します。

```
listremoteconnections /file C:¥temp¥list01.csv /user user01 /password pass01
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)

## 1.6.8 listservices (サービス・サービステンプレートの一覧出力)

### 機能

JP1/AO に登録されているサービスまたはサービステンプレートの一覧を CSV 形式で出力します。

ベンダー名やバージョン情報を含んだサービスの一覧またはサービステンプレートの一覧を出力できます。なお、デバッグサービスについては出力されません。

### 形式

```
listservices
  /output {services | servicetemplates}
  /file 出力ファイルパス
  [/encoding {UTF-8 | Shift_JIS}]
  /user ユーザーID
  {/password パスワード | /passwordfile パスワードファイルパス}
```

### 引数

```
/output {services | servicetemplates}
```

出力する一覧情報の種類として、次のどちらかを値に指定します。

- `services`  
サービス一覧を出力します。
- `servicetemplates`  
サービステンプレート一覧を出力します。

#### `/file` 出力ファイルパス

出力ファイルのパスを絶対パスまたは相対パスで指定します。指定したファイルが存在する場合はエラーとなります。

#### `/encoding {UTF-8 | Shift_JIS}`

出力ファイルのエンコーディングを指定します。このオプションを省略した場合は、OS のデフォルトエンコーディングとなります。

#### `/user ユーザーID`

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字、`!`, `#`, `$`, `%`, `&`, `'`, `(`, `)`, `*`, `+`, `-`, `.`, `=`, `@`, `¥`, `^`, `_`, `|` です。

大文字と小文字は区別しません。

#### `/password パスワード`

`user` オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたは`passwordfile` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、`user` オプションと同じです。

#### `/passwordfile パスワードファイルパス`

`user` オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルは`encryptpassword` コマンドで作成します。

このオプションまたは`password` オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 出力形式

サービスまたはサービステンプレートごとに 1 行ずつ、出力項目が CSV 形式で出力されます。

なお、出力項目の値は「`"`」で囲まれて出力されます。値に「`"`」が含まれる場合、直前に「`"`」を付加してエスケープされます。

表 1-7 サービス一覧およびサービステンプレート一覧の出力形式

出力情報の種類	出力項目	内容
サービス一覧	Name	サービス名

出力情報の種類	出力項目	内容
サービス一覧	Favorite	お気に入り設定の状態
	Description	説明
	Service Group	サービスグループ名
	Service Template	サービステンプレート名
	Vendor Name	ベンダー名
	Version	サービスバージョン
	Tags	サービスに設定されているタグ
	Status	状態
	Create Date	作成日時
	Modify Date	最終更新日時
	Submit Date	最終実行日時
	Reset Date	リセット日時
	Executed Count	タスク実行回数
	Completed Count	正常終了回数
	Last Failed Date	最終失敗日時
	Failed Count	失敗回数
	Submit Count	サービス実行回数
	ID	サービス ID
	Latest	サービステンプレートが最新バージョンかどうか
	Supported Schedule Type	選択できるスケジュール
Supported Action Type	タスクに対して実行できる操作	
サービステンプレート一覧	Name	サービステンプレート名
	Vendor	ベンダー名
	Version	サービステンプレートバージョン
	Description	説明
	Service Template Key Name	サービステンプレート ID
	Vendor ID	ベンダー ID
	Tags	サービステンプレートに設定されているタグ
	Registered	作成時刻

出力情報の種類	出力項目	内容
サービスプレート一覧	Updated	最終更新時刻
	Latest Version	サービスプレートが最新バージョンかどうか
	Used Services	サービスプレートを使用しているサービスの数
	Used Service Templates	このサービスプレートをコンポーネントとして使用しているサービスプレートの数
	Outdated Services	古いバージョンのサービスプレートを使用しているサービスがあるかどうか
	Outdated Component	古いコンポーネントを使用しているかどうか
	Supported Schedule Type	選択できるスケジュール
	Supported Action Type	このサービスプレートを使用しているタスクに対して実行できる操作
	Release State	リリース状態

ファイルの出力例を次に示します。

- サービス一覧の場合

```
"Name", "Favorite", "Description", "Service Group", "Service Template", "Vendor
Name", "Version", "Tags", "Status", "Create Date", "Modify Date", "Submit Date", "Reset Date"
, "Executed Count", "Completed Count", "Last Failed Date", "Failed Count", "Submit
Count", "ID", "Latest", "Supported Schedule Type", "Supported Action Type"
"リモートコマンド実行", "false", "リモートの実行対象サーバ上のコマンドを実行します。
", "DefaultServiceGroup", "リモートコマンド実行", "(株)日立製作所
", "01.12.00", "Basic, OS Operations"
, "リリース", "2015-08-28 13:07:25", "2015-08-28 13:07:25", "2015-08-28
13:20:26", "", "3", "1", "2015-08-28 13:17:58", "2", "3", "4005", "はい
", "immediate, schedule, recurrence", "forciblyStop, retry"
```

- サービスプレート一覧の場合

```
"Name", "Vendor", "Version", "Description", "Service Template Key Name", "Vendor
ID", "Tags", "Registered", "Updated", "Latest Version", "Used Services", "Used Service
Templates", "Outdated Services", "Outdated Component", "Supported Schedule Type", "Release
State", "Supported Action Type"
"OSユーザーの一覧取得", "(株)日立製作所", "02.00.00", "Windows/UNIXのOSユーザーの一覧を取得
します。", "osShowUsers", "com.hitachi.software.dna.cts.jp1", "AIX, Gather OS
information, Linux, Windows", "2016-11-17 15:04:12", "2016-11-17 15:04:12", "はい", "0", "0", "い
いえ", "いいえ", "immediate, schedule, recurrence", "リリース", ""
```



## 格納先

Windows の場合

`JPI/AO` のインストール先フォルダ¥bin

Linux の場合

`/opt/jplao/bin`

## 実行権限

OS の Administrators 権限または root 権限を持つユーザーで実行してください。

サービス一覧を出力する場合、コマンド実行ユーザーが所属するユーザーグループから、操作対象のサービスグループに対して Admin ロール、Develop ロール、Modify ロールまたは Submit ロールが設定されている必要があります。ロールが設定されていないサービスグループのサービスについては出力されません。

サービステンプレート一覧を出力する場合、コマンド実行ユーザーが所属するユーザーグループから、操作対象のサービスグループに対して Admin ロール、Develop ロールまたは Modify ロールが設定されている必要があります。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
8	すでに出力先パスに同名のファイルが存在します。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
12	エンコーディングの指定が不正です。
13	ファイル出力に失敗しました。
14	コマンドの実行権限がありません。
160	サービス一覧の取得に失敗しました。

戻り値	説明
161	サービステンプレート一覧の取得に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、登録されているサービスの一覧を OS のデフォルトエンコーディングでファイルに出力する場合  

```
listservices /output services /file list01 /user user01 /password pass01
```
- Windows で、登録されているサービステンプレートの一覧を UTF-8 エンコーディングでファイルに出力する場合  

```
listservices /output servicetemplates /file list02 /encoding UTF-8 /user user02 /password pass02
```

---

## 関連項目

- 1.3 コマンドの引数に指定できる文字
- 

## 1.6.9 listtasks (タスクの一覧出力, タスク詳細情報出力)

### 機能

listtasks コマンドの機能は次のとおりです。

### タスクまたは履歴の一覧出力

listtasks コマンドのoutput オプションにtasks を指定して実行すると、[タスク] 画面のタスク一覧に表示されるタスク情報を CSV 形式で出力できます。またはoutput オプションにhistories を指定して実行すると、[タスク] 画面の履歴一覧に表示されるタスク情報を CSV 形式で出力できます。なお、デバッグタスクについてはどちらの CSV ファイルにも出力されません。

期間を指定すると、出力するタスク情報を絞り込むこともできます。

### タスク詳細情報の出力

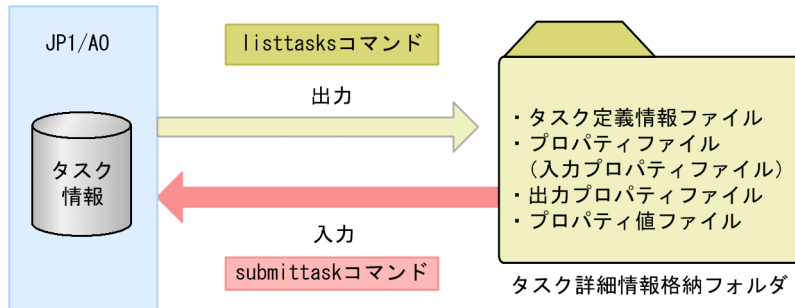
listtasks コマンドのoutput オプションにtaskdetails を指定して実行すると、タスク詳細情報格納フォルダに、入力プロパティおよび出力プロパティの値を含むタスク詳細情報が出力されます。なお、デバッグタスクの詳細情報については出力されません。

出力されたタスク詳細情報<sup>※</sup>を元に、`submittask` コマンドを実行すると、指定日時実行タスクと定期実行タスクを、同一の設定で一括再登録できます。

## 注※

このタスク詳細情報には、定義情報（サービス、サービステンプレート、ユーザー、ユーザーグループ、サービスグループ、エージェントレス接続先定義、サービス共有プロパティ）および定義ファイルは含まれません。これらの情報をバックアップしたい場合は、`backupsystem` コマンドを使用してください。

図 1-1 タスク詳細情報の出力と入力



## 形式

```
listtasks
  [/startrange {yyyy-mm-dd|,yyyy-mm-dd|yyyy-mm-dd,yyyy-mm-dd}]
  /output {tasks | histories | taskdetails}
  {/file出力ファイルパス | /taskdetaildir タスク詳細情報格納フォルダのパス}
  [/encoding {UTF-8 | Shift_JIS}]
  /user ユーザーID
  {/password パスワード | /passwordfile パスワードファイルパス}
```

## 引数

`/startrange {yyyy-mm-dd|,yyyy-mm-dd|yyyy-mm-dd,yyyy-mm-dd}`

タスクの開始日または開始予定日を指定することで、出力するタスクを絞り込みます。

「yyyy」には西暦を4桁の半角数字で指定します。「mm」には月を1（または01）から12の半角数字で指定します。「dd」には日を1（または01）から31の半角数字で指定します。

- `yyyy-mm-dd`  
指定した日付以後に開始したタスクまたは開始予定のタスクを出力します。
- `,yyyy-mm-dd`  
指定した日付以前に開始したタスクまたは開始予定のタスクを出力します。
- `yyyy-mm-dd,yyyy-mm-dd`  
指定した期間内に開始したタスクまたは開始予定のタスクを出力します。「,」の右側には、左側に指定した日付以降の日付だけを指定できます。

定期実行タスクの場合、次に開始予定のタスクまで出力されます。その次以降に開始予定のタスクは出力されません。

待機中のタスクを出力したい場合、定期実行タスクおよび指定日時実行タスクのときは開始日時ではなく開始予定日時を指定してください。即時実行タスクのときは実行操作日時を指定してください。実行操作日時は [タスク詳細] 画面で確認できます。

このオプションを省略した場合は、ユーザーが参照できるすべてのタスクが出力されます。

なお、output オプションにtaskdetails を指定した場合に、startrange オプションを指定するとエラーになります。

`/output {tasks | histories | taskdetails}`

出力する一覧情報の種類として、次のどれかを値に指定します。

- **tasks**  
[タスク] 画面のタスク一覧を出力します。
- **histories**  
[タスク] 画面の履歴一覧を出力します。
- **taskdetails**  
入力プロパティおよび出力プロパティの値を含むタスク詳細情報を出力します。

`{/file 出力ファイルパス | /taskdetaildir タスク詳細情報格納フォルダのパス}`

- **/file 出力ファイルパス**  
一覧として出力するファイルのパスを絶対パスまたは相対パスで指定します。指定したファイルがすでにある場合はエラーとなります。  
output オプションにtasks またはhistories を指定した場合は、必須で指定します。output オプションにtaskdetails を指定した場合に、このオプションを指定するとエラーになります。
- **/taskdetaildir タスク詳細情報格納フォルダのパス**  
タスク詳細情報を出力するための、空のフォルダを絶対パスまたは相対パスで指定します。指定できるのはローカルディスクのフォルダだけです。指定できる文字数は、絶対パスの長さで 180 文字までです。相対パスの場合、絶対パスの長さに換算したときに 180 文字までとなるように指定します。  
指定したフォルダがない場合や、指定したフォルダ内にファイルまたはフォルダがすでにある場合はエラーになります。  
output オプションにtaskdetails を指定した場合は、必須で指定します。output オプションにtasks またはhistories を指定した場合に、このオプションを指定するとエラーになります。

`/encoding {UTF-8 | Shift_JIS}`

出力ファイルのエンコーディングを指定します。このオプションを省略した場合は、OS のデフォルトエンコーディングとなります。

output オプションにtaskdetails を指定した場合、ここで指定したエンコーディングは、タスク詳細情報格納フォルダの直下にあるタスク一覧ファイル (listtasks.csv) だけに適用されます。タスク詳細情報、プロパティファイル (入力プロパティファイル)、および出力プロパティファイルは、常に UTF-8 で出力されます。

## /user ユーザーID

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1～256 文字です。

使用できる文字の種類は、半角英数字、[!], [#], [\$], [%], [&], ['], [(, ()], [\*], [+], [-], [.] , [=], [@], [¥], [^], [\_], [|] です。

大文字と小文字は区別しません。

## /password パスワード

user オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたはpasswordfile オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1～256 文字です。

使用できる文字の種類は、user オプションと同じです。

## /passwordfile パスワードファイルパス

user オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルはencryptpassword コマンドで作成します。

このオプションまたはpassword オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 出力形式

listtasks コマンドの出力形式は次のとおりです。

### タスクおよび履歴の一覧を出力する場合 (output オプションにtasks またはhistories を指定した場合)

タスクごとに 1 行ずつ、出力項目が CSV 形式で出力されます。

なお、出力項目の値は「"」で囲まれて出力されます。値に「"」が含まれる場合、直前に「\"」を付加してエスケープされます。

表 1-8 タスク一覧の出力項目

出力項目	内容
Task Name	タスク名
To Do	To Do 設定の状態
Status	タスクの状態
Scheduled Time	開始予定日時
Start Time	開始日時
Completion Time	終了日時
Schedule Type	タスク種別

出力項目	内容
Task ID	タスク ID
Description	タスクの説明
Service	サービス名
Service Group	サービスグループ名
Tags	サービスに設定されているタグ
Submitted By	実行者
Submit Time	実行操作日時
Schedule Interval	定期実行周期
Recurrence Time	定期実行時刻
Schedule Start Date	定期実行適用開始日
Notes	メモ
Step Start Time	ステップ開始日時
Supported Action Type	タスクに対して実行できる操作
Service status	サービスの状態

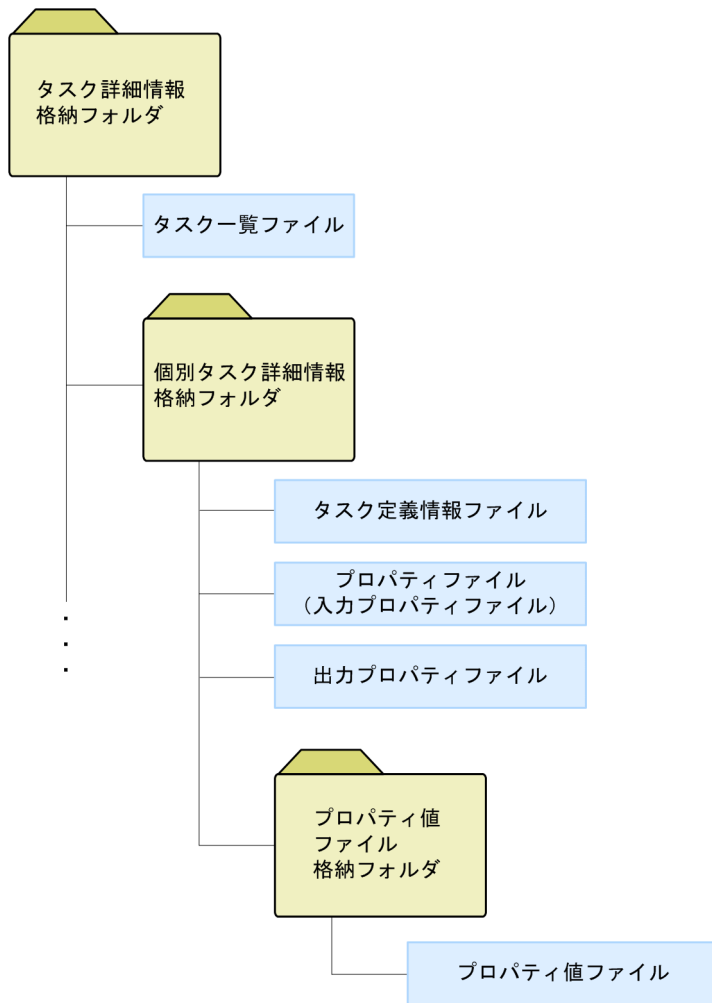
ファイルの出力例を次に示します。

```
"Task Name","To Do","Status","Scheduled Time","Start Time","Completion Time","Schedule
Type",
"Task ID","Description","Service","Service Group","Tags","Submitted By","Submit
Time","Schedule Interval",
"Recurrence Time","Schedule Start Date","Notes","Step Start Time","Supported Action
Type","Service Status"
"リモートコマンド実行_20150828130932","FALSE","失敗","","2015/8/28 13:09","2015/8/28
13:09","即時実行","4015",
"", "リモートコマンド実行", "DefaultServiceGroup", "Basic, OS_Operations", "System", "2015/8/28
13:09", "", "", "", "", "", "", "forciblyStop, retry", "リリース"
```

## タスク詳細情報を出力する場合 (output オプションにtaskdetails を指定した場合)

タスク詳細情報格納フォルダに出力されるデータを次に示します。

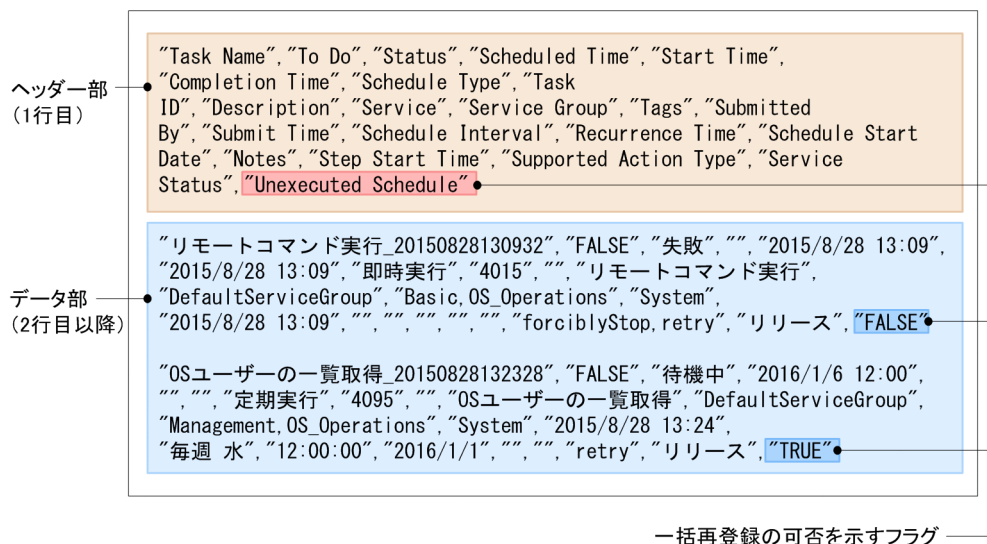
図 1-2 タスク詳細情報格納フォルダの構成



タスク詳細情報格納フォルダの内容を次に示します。

- タスク詳細情報格納フォルダ  
フォルダ名：任意  
taskdetaildir オプションに指定するフォルダです。
- タスク一覧ファイル  
ファイル名：listtasks.csv  
タスク詳細情報に含まれるタスクの一覧と、その一覧のうちsubmittask コマンドで一括再登録する対象となるタスクを確認するためのファイルです。  
output オプションに、tasks を指定して出力されるタスク一覧の各行の末尾に、submittask コマンドで一括再登録するタスクかどうかのフラグが付加されています。  
タスク一覧の出力例を次に示します。

図 1-3 タスク一覧の出力例



ヘッダー部とデータ部の内容は次のとおりです。

ヘッダー部 (1 行目)

Unexecuted Schedule

データ部 (2 行目以降)

true:一括再登録されるタスクです。

false:一括再登録されないタスクです。

データ部 (2 行目以降) のフラグがtrueになるのは、予定されたタスク (指定日時実行タスクまたは定期実行タスク) のうち未実行のタスクです。

- listtasks コマンドを実行する時点で、実行が開始されていない指定日時実行タスク
- listtasks コマンドを実行する時点で、キャンセルされていない定期実行タスク
- listtasks コマンドを実行する時点で、保留中の指定日時実行タスクと定期実行タスク

なお、このファイルは、submittask コマンドで一括再登録するタスクを確認するためのものです。このファイルは編集しないでください。

- 個別タスク詳細情報格納フォルダ

フォルダ名:タスク ID

個々のタスクの定義情報、プロパティファイル (入力プロパティファイル)、および出力プロパティファイルを格納するフォルダです。タスク ID がフォルダ名となり、出力するタスクの数だけフォルダが生成されます。

- タスク定義情報ファイル

ファイル名:taskdef.xml

タスクの定義情報を XML で出力したファイルです。このファイルは編集しないでください。

- プロパティファイル (入力プロパティファイル)

ファイル名:input.properties



タスクに設定された入力プロパティの情報が「key=value」の形式で出力されます。プロパティファイル（入力プロパティファイル）の形式については、マニュアル「*JPI/AO 運用ガイド*」を参照してください。なお、出力されるのは、可視性が「設定画面と実行画面」の入力プロパティだけです。該当する入力プロパティが1つもない場合は、空のファイルが作成されます。

submittask コマンドでタスクの一括再登録を行う場合、このファイルを編集して実行しないでください。編集した場合のコマンドの実行結果は製品サポートの対象外です。ただし、元の設定と異なる設定で登録する場合は、このファイルをコピーして使用してください。ファイルのコピー後に、指定日時やタスク名など、サービスの実行時に指定できる項目を指定してから、個別にサービスを実行してください。入力プロパティの値を変更したい場合は、コピーしたファイルを必要に応じて編集し、submittask コマンドにpropertyfile オプションを指定して実行してください。

なお、ホスト名、IP アドレス、ユーザー名などの情報が入力プロパティの値に含まれている場合、そのまま出力されます。ただし、プロパティのデータ型が「password」の場合、難読化された値が出力されます。

- 出力プロパティファイル

ファイル名：output.properties

タスクに設定された出力プロパティの内容が「key=value」の形式で出力されます。該当する出力プロパティが1つもない場合は、空のファイルが作成されます。

なお、ホスト名、IP アドレス、ユーザー名などの情報が出力プロパティの値に含まれている場合、そのまま出力されます。ただし、プロパティのデータ型が「password」の場合、難読化された値が出力されます。

- プロパティ値ファイル格納フォルダ

フォルダ名：value\_files

プロパティ値ファイルを格納するフォルダです。

- プロパティ値ファイル

- 入力プロパティのプロパティ値ファイル名

input\_番号\*.txt

- 出力プロパティのプロパティ値ファイル名

output\_番号\*.txt

注※

番号は0001から始まる連番で、プロパティ種別（入力プロパティまたは出力プロパティ）ごとに採番されます。

サービスの入力/出力プロパティに composite 型のプロパティが含まれる場合、そのプロパティ値がテキストファイルで出力されます。プロパティ値ファイルの形式については、マニュアル「*JPI/AO 運用ガイド*」を参照してください。

## 格納先

Windows の場合

`JP1/AO` のインストール先フォルダ¥bin

Linux の場合

`/opt/jplao/bin`

## 実行権限

OS の Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

`user` オプションに指定するユーザーに必要な権限は、`output` オプションに指定する引数によって異なります。

`output` オプションに `tasks` または `histories` を指定する場合（タスクまたは履歴の一覧出力）

`user` オプションに指定するユーザーが所属するユーザーグループから、操作対象のサービスグループに対して Admin ロール、Develop ロール、Modify ロールまたは Submit ロールが設定されている必要があります。ロールが設定されていないサービスグループのタスクについては出力されません。

`output` オプションに `taskdetails` を指定する場合（タスク詳細情報の出力）

`user` オプションには、Admin ロールを持つユーザーを指定してください。JP1/AO に登録されているすべてのタスクの情報が一括で出力されるほか、入力プロパティや出力プロパティの値に、ホスト名、IP アドレス、ユーザー名、パスワードなどの情報が含まれている場合も、そのまま出力されるため、すべてのリソースにアクセスできる Admin ロールを指定する必要があります。出力されたタスク詳細情報は、適切にアクセス制御された場所に保管してください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
8	すでに出力先パスに同名のファイルが存在します。

戻り値	説明
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
11	指定したフォルダが空ではありません。
12	エンコーディングの指定が不正です。
13	ファイル出力に失敗しました。
14	コマンドの実行権限がありません。
150	タスク一覧の取得に失敗しました。
151	履歴一覧の取得に失敗しました。
152	タスク詳細情報の取得に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、登録されているタスクの一覧を OS のデフォルトエンコーディングでファイルに出力する場合  

```
listtasks /output tasks /file list01.csv /user user01 /password pass01
```
- Windows で、2012 年 1 月 1 日から 2012 年 3 月 31 日までの間に開始または開始予定の履歴一覧のタスクを UTF-8 エンコーディングでファイルに出力する場合  

```
listtasks /startrange 2012-01-01,2012-03-31 /output histories /file list02.csv /encoding UTF-8 /user user02 /password pass02
```
- Windows で、タスク詳細情報を出力する場合  

```
listtasks /output taskdetails /taskdetaildir "C:¥data¥taskdetail" /user user03 /password pass03
```

## 関連項目

- 1.3 コマンドの引数に指定できる文字
- 1.6.12 `submittask` (サービスの実行, タスクの一括再登録)
- 1.7.1 `backupsystem` (JP1/AO システムのバックアップ)
- 1.7.6 `restoresystem` (JP1/AO システムのリストア)
- マニュアル「JP1/AO 運用ガイド」 - 「プロパティファイルについて」のトピック

## 1.6.10 setremoteconnection (エージェントレス接続先定義の追加・更新)

### 機能

エージェントレス接続先定義情報ファイル (CSV 形式) を使用して、エージェントレス接続先定義を登録または更新します。

このコマンドを実行する前に、listremoteconnections コマンドを実行して、出力されたエージェントレス接続先定義情報ファイルを編集してください。

### 形式

```
setremoteconnection
  /file エージェントレス接続先定義情報ファイルのパス
  /user ユーザーID
  {/password パスワード | /passwordfile パスワードファイルパス}
```

### 引数

#### /file

エージェントレス接続先定義情報ファイルのパスを絶対パスまたは相対パスで指定します。指定したファイルが存在しない場合はエラーとなります。

#### /user

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字、[!], [#], [\$], [%], [&], ['], [(], [)], [\*], [+], [-], [.] , [=], [@], [¥], [^], [\_], [||] です。

大文字と小文字は区別しません。

#### /password

user オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたはpasswordfile オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、user オプションと同じです。

#### /passwordfile

user オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルはencryptpassword コマンドで作成します。

このオプションまたはpassword オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## エージェントレス接続先定義情報ファイルの形式

エージェントレス接続先定義情報ファイルの形式を次の表に示します。

表 1-9 エージェントレス接続先定義情報ファイルの形式

項目	指定内容※1
Id	エージェントレス接続先定義を追加したい場合 空文字を指定します。  エージェントレス接続先定義を更新したい場合 情報を更新したいエージェントレス接続先定義の ID を指定します。
Method	接続先種別として次のどれかを指定します。大文字と小文字は区別しません。 <ul style="list-style-type: none"><li>• IPv4</li><li>• IPv6</li><li>• HostName</li></ul>
IP Address/Host Name	接続先ホストの IP アドレスまたはホスト名を指定します。
Service Group	エージェントレス接続先定義に割り当てるサービスグループを指定します。
Authentication	認証情報を設定するかどうかを指定します。大文字と小文字は区別しません。 <ul style="list-style-type: none"><li>• Enable 認証情報を設定する。</li><li>• Disable 認証情報を設定しない。</li></ul>
Protocol※2	接続先ホストとの通信で使用する認証用プロトコルを指定します。大文字と小文字は区別しません。 <ul style="list-style-type: none"><li>• Windows</li><li>• SSH</li><li>• Telnet</li></ul>
SSH Authentication Method※2	「Protocol」が「SSH」の場合に、接続先ホストとの通信で使用する認証方式を指定します。() 内に示す文字列で指定することもできます。大文字と小文字は区別しません。 <ul style="list-style-type: none"><li>• Password Authentication (PW) パスワード認証</li><li>• Public Key Authentication (PK) 公開鍵認証</li><li>• Keyboard Interactive Authentication (KI) キーボードインタラクティブ認証</li></ul>
User ID※2	接続先ホストにログインするためのユーザー ID を指定します。「Protocol」が「Windows」または「SSH」の場合は必ず指定してください。
Password※2	接続先ホストにログインするためのパスワードを指定します。次の場合に、必ず指定してください。 <ul style="list-style-type: none"><li>• 「Protocol」に「Windows」を指定している場合</li><li>• 「SSH Authentication Method」に「Password Authentication」または「Keyboard Interactive Authentication」を指定している場合</li></ul> また、「Id」に値が指定されているかどうかで、次のように動作します。

項目	指定内容※1
Password※2	<p>[Id] に値が指定されていない場合</p> <ul style="list-style-type: none"> <li>「Password」に「*****」を指定するとエラーになります。</li> </ul> <p>[Id] に値が指定されている場合</p> <ul style="list-style-type: none"> <li>「Password」に「*****」を指定すると、パスワードは変更されません。</li> <li>「Password」に空文字を指定すると、パスワードが削除されます。</li> </ul>
Super User's Password※2	<p>接続先ホストのスーパーユーザーのパスワードを指定します。「Protocol」に「SSH」または「Telnet」を指定している場合に指定します。</p> <p>また、「*****」を指定すると、パスワードは変更されません。空文字を指定した場合は、パスワードが削除されます。</p>
Connection Status	<p>最後にホストに接続した時の状態を指定します。</p> <ul style="list-style-type: none"> <li>Connection Successful 接続成功</li> <li>Error 接続失敗</li> <li>Unknown 未接続</li> <li>- 対象外</li> </ul>
Connected Time	最後にホストに接続した時刻を指定します。

#### 注※1

値は「"」で囲まれていなくてもエラーになりません。ただし、値に「"」が含まれる場合は、直前に「\"」を付加してエスケープしてください。

#### 注※2

「Authentication」で「Disable」を指定している場合は、空文字を指定してください。

## 格納先

Windows の場合

*JPI/AO* のインストール先フォルダ¥bin

Linux の場合

/opt/jplao/bin

## 実行権限

OS の Administrators 権限または root 権限と、JPI/AO の Admin ロールを持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 戻り値

コマンドの戻り値を次の表に示します。

### 1. コマンド

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
14	コマンドの実行権限がありません。
230	エージェントレス接続先定義の指定内容が不正です。
231	一部のエージェントレス接続先定義の登録に失敗しました。
232	すべてのエージェントレス接続先定義の登録に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、list01.csv の内容に従ってエージェントレス接続先定義を追加・更新する場合のコマンドの使用例を示します。

```
setremoteconnection /file list01.csv /user user01 /password pass01
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)

### 1.6.11 stoptask (タスクの実行停止)

#### 機能

タスク ID を指定することで、指定したタスクを実行停止します。ただし、デバッグタスクについては実行停止できません。

なお、このコマンドでは、タスクの強制停止はできません。

## 形式

```
stoptask  
  /taskid タスクID  
  /user ユーザーID  
  {/password パスワード | /passwordfile パスワードファイルパス}
```

## 引数

### */taskid* タスクID

実行停止するタスクのタスク ID を指定します。

指定できる値は、16桁までの半角数字（10進数）です。

### */user* ユーザーID

JP1/AO のユーザー ID を指定します。

指定できる文字数は、1～256文字です。

使用できる文字の種類は、半角英数字、「!」、「#」、「\$」、「%」、「&」、「'」、「(」、「)」、「\*」、「+」、「-」、「.」、「=」、「@」、「¥」、「^」、「\_」、「|」です。

大文字と小文字は区別しません。

### */password* パスワード

*user* オプションで指定した、ユーザーのパスワードを指定します。

このオプションまたは*passwordfile* オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

指定できる文字数は、1～256文字です。

使用できる文字の種類は、*user* オプションと同じです。

### */passwordfile* パスワードファイルパス

*user* オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルは*encryptpassword* コマンドで作成します。

このオプションまたは*password* オプションのどちらかを指定する必要があります。両方のオプションが指定されている場合、およびどちらのオプションも指定されていない場合はエラーとなります。

## 格納先

Windows の場合

*JP1/AO* のインストール先フォルダ¥bin

Linux の場合

/opt/jplao/bin



## 実行権限

OS の Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

コマンド実行ユーザーが所属するユーザーグループから、実行停止したいタスクのサービスグループに対して、Admin ロール、Develop ロール、Modify ロールまたは Submit ロールが設定されている必要があります。ロールが設定されていないサービスグループのタスクについては実行停止できません。

## 注意事項

実行停止したいタスクの状態が実行中、応答待ち中、または異常検出の場合に、このコマンドを実行してください。これ以外の状態の場合にコマンドを実行すると、コマンドの戻り値が 140 で失敗します。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
9	指定したパスが存在しません。
14	コマンドの実行権限がありません。
140	タスクの実行停止に失敗しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、指定したタスク（タスク ID が 1）を実行停止する場合のコマンドの使用例を示します。

```
stoptask /taskid 1 /user user01 /password pass01
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
- マニュアル「JP1/AO 運用ガイド」－「タスクを停止する手順（実行停止）」のトピック

## 1.6.12 submittask (サービスの実行, タスクの一括再登録)

### 機能

submittask コマンドの機能は次のとおりです。

### サービスの実行

サービス名, サービスグループ名, およびプロパティ値を指定してコマンドを実行することで, 指定したサービスを実行します。タスクが正常に実行された場合, タスク ID がメッセージに出力されます。デバッグサービスについては実行できません。オプションを指定することで, サービスを定期実行したり, 実行日時を指定して実行したりできます。何も指定しない場合, サービスは即時実行されます。

### タスクの一括再登録

listtasks コマンドで出力したタスク詳細情報の内容を元に, 指定日時実行タスクおよび定期実行タスクを一括で再登録します。再登録したタスクには, 元のタスクの設定およびタスクの状態が引き継がれます。これは, タスク詳細情報格納フォルダの情報を参照して, 同一の設定で指定日時実行または定期実行でサービスを実行する機能であり, 同一のタスクを復元する機能ではありません。そのため, 再登録されたタスクは, 再登録前のタスクとは別のタスクであり, タスク ID も異なります。デバッグタスクについては再登録できません。

タスクを一括再登録する前の作業について

タスクの一括再登録をする前に, 次の作業を実施してください。

- タスク詳細情報格納フォルダには, タスクの一括再登録を行う JP1/AO と, バージョンおよびリビジョンが同一の JP1/AO で出力したフォルダを指定してください。バージョンまたはリビジョンが異なる JP1/AO で出力したタスク詳細情報格納フォルダを指定した場合, タスクの一括再登録に失敗します。
- listtasks コマンドで出力されたタスク詳細情報格納フォルダがあることを確認してください。
- 定義情報 (サービス, サービステンプレート, ユーザー, ユーザーグループ, サービスグループ, エージェントレス接続先定義, サービス共有プロパティ) および定義ファイルを別途設定してください。必要に応じて, backupsystem コマンドおよびrestoresystem コマンドを使用して, これらの設定を復元してください。listtasks コマンドでタスク詳細情報格納フォルダを出力したときから, これらの設定を変更したり, サービスを削除したりしないでください。サービスの設定が変更されていた場合, 変更後の設定に従って, 一括再登録が実行されます。また, サービス設定が削除されていた場合は, 該当のタスクの再登録に失敗します。
- 指定日時実行タスクの場合, タスク再登録時点で指定日時が過ぎていないことを確認してください。指定日時が過ぎていると, エラーとなるため, そのままでは再登録できません。

## 一括再登録されるタスクの条件について

一括再登録されるタスクは、タスク詳細情報格納フォルダ内にあるタスク一覧 (listtasks.csv) のうち、未実行の指定日時実行タスクおよび定期実行タスクです。未実行の指定日時実行タスクおよび定期実行タスクは、タスク一覧で「Unexecuted Schedule」の列が「true」となっています。

## 指定日時が過ぎている指定日時実行タスクの再登録について

指定日時が過ぎている指定日時実行タスクはそのままでは再登録できません。一括再登録しても、指定日時が過ぎている旨のメッセージが出力されて、再登録が失敗します。再登録に失敗したタスクの設定内容を確認したい場合は、タスク詳細情報格納フォルダ内のタスク一覧ファイル (listtasks.csv) と、個別タスク詳細情報格納フォルダのプロパティファイル (入力プロパティファイル) および出力プロパティファイルを参照してください。指定日時が過ぎているタスクを登録したい場合は、タスク詳細情報格納フォルダ内にあるタスク一覧 (listtasks.csv) の元の指定日時を確認してから、JP1/AO の [サービス] 画面または submittask コマンドで、新しく日時を指定して個別にサービスを実行してください。なお、開始予定日時は現在の日時以降を指定してください。

## 再登録に失敗したタスクがある場合の対処について

タスクの一括再登録に失敗した場合は、タスク登録が失敗した旨のメッセージと、再登録に失敗したタスクのタスク ID が表示されます。このときのタスク ID とは、listtasks コマンドで出力した時点のタスク ID です。一部、再登録に成功したタスクがある場合、該当するタスクの個別タスク詳細情報格納フォルダをほかの場所に退避させてください。その後、失敗原因を取り除いてからコマンドを再実行してください。フォルダを退避させるのは、再登録に成功したタスクの 2 重登録を避けるためです。この対処をしたあとも同様のエラーが発生する場合は、システム管理者に連絡してください。

## 形式

submittask コマンドの形式は次のとおりです。

## サービスを即時実行する場合

```
submittask
  /servicename サービス名
  [/servicegroup サービスグループ名]
  [/taskname タスク名]
  [/taskdescription タスクの説明]
  [/property プロパティキー プロパティ値 |
  /propertyfile プロパティファイルパス]
  /user ユーザーID
  {/password パスワード | /passwordfile パスワードファイルパス}
  [/wait]
```

## サービスを指定した日時に実行する場合

```
submittask
  /servicename サービス名
  [/servicegroup サービスグループ名]
  [/taskname タスク名]
  [/taskdescription タスクの説明]
```

```
[/property プロパティキー プロパティ値 |  
/propertyfile プロパティファイルパス]  
/user ユーザーID  
{/password パスワード | /passwordfile パスワードファイルパス}  
/scheduledate yyyy-mm-dd /scheduletime hh:mm
```

## サービスを定期的に実行する場合

```
submittask  
  /servicename サービス名  
  [/servicegroup サービスグループ名]  
  [/taskname タスク名]  
  [/taskdescription タスクの説明]  
  [/property プロパティキー プロパティ値 |  
  /propertyfile プロパティファイルパス]  
  /user ユーザーID  
  {/password パスワード | /passwordfile パスワードファイルパス}  
  /recurrencepattern {daily[:{1h|2h|3h|4h|6h|8h|12h|24h}] | weekly:sun, mon, ..., sat |  
  monthly:{dd, dd, ..., dd [,endofmonth] | endofmonth}}  
  /recurrencetime hh:mm /recurrencestart yyyy-mm-dd
```

## タスクを一括再登録する場合

```
submittask  
  /reregister  
  /taskdetaildir タスク詳細情報格納フォルダ  
  [/setoriginalsubmitter]  
  /user ユーザーID  
  {/password パスワード | /passwordfile パスワードファイルパス}
```

## 引数

**/servicename サービス名**

実行するサービスの名称を指定します。

指定できる文字数は、1~128文字です。

**/servicegroup サービスグループ名**

実行するサービスが所属するサービスグループの名称を指定します。

このオプションを省略した場合は、引数に指定したユーザーに関連づいたサービスグループが設定されますが、ユーザーに関連づいたサービスグループが複数存在する場合、エラーとなります。

指定できる文字数は、1~80文字です。使用できる文字の種類は、UnicodeのU+10000~U+10FFFF以外です。

なお、servicegroup オプションの代わりに、JP1/AO 10-52 以前で使用していたresourcegroup オプションを指定して、サービスグループ名を指定することもできます。servicegroup オプションに「All Resources」を指定した場合でも、「DefaultServiceGroup」と指定したとみなされて実行されます。

**/taskname タスク名**

タスクの名称を指定します。

このオプションを省略した場合は、デフォルトの名称として「サービス名\_YYYYMMDDhhmmss」(YYYYMMDDhhmmss：サービス実行時の時刻) が設定されます。

指定できる文字数は 1~128 文字です。使用できる文字の種類は、制御文字 (¥u0000~¥u001F および ¥u007F~¥u009F) 以外です。

#### `/taskdescription` タスクの説明

タスクの説明を指定します。

このオプションを省略した場合は値が設定されません。

指定できる文字数は、1~256 文字です。使用できる文字の種類は、制御文字 (¥u0000~¥u001F および ¥u007F~¥u009F) 以外です。

#### `/property` プロパティキー プロパティ値

実行するサービスで使用するプロパティキーおよびプロパティ値を指定します。指定したプロパティ値はサービステンプレートの仕様に従って妥当かどうか検証されます。

このオプションで設定しなかったプロパティキーについては、[サービス設定] 画面で指定した値が設定されます。指定が必須のプロパティについて、[サービス設定] 画面で値を指定していない場合、かつこのオプションでも値を指定しなかった場合、エラーとなります。

このオプションは、「`/property` プロパティキー1 プロパティ値1 `/property` プロパティキー2 プロパティ値2」のように複数指定できます。指定できるプロパティキーとプロパティ値の組み合わせは、デフォルトでは最大 1,000 個です。指定できるプロパティの個数の上限は、ユーザー設定プロパティファイル (config\_user.properties) で定義できます。

- **プロパティキー**

サービスのプロパティキーを指定します。

指定できる文字数は、1~1,024 文字です。使用できる文字の種類は、半角英数字、「-」、「\_」、「.」、「/」です。

同一のプロパティキーが指定された場合は、エラーとなります。

- **プロパティ値**

プロパティキーに対応するプロパティ値を指定します。

空白と特殊記号を含む場合、「"」で囲んで指定します。

#### `/propertyfile` プロパティファイルパス

実行対象のサービスで使用する入力プロパティの設定を定義したプロパティファイルのパスを、絶対パスまたは相対パスで指定します。

このオプションで設定しなかったプロパティキーおよびプロパティ値については、[サービス設定] 画面 (作成, 編集, または複製), または [サービス実行] 画面で指定した値が設定されます。指定が必須のプロパティについて、[サービス設定] 画面 (作成, 編集, または複製), または [サービス実行] 画面で値を指定していない場合、かつこのオプションで指定したプロパティファイルでも値が定義されていない場合、エラーとなります。

プロパティファイルの形式については、マニュアル「[JP1/AO 運用ガイド](#)」を参照してください。

## `/reregister`

タスクを一括再登録する場合に指定します。`reregister` オプションを指定する場合は、`taskdetaildir` オプションも必ず指定してください。

## `/taskdetaildir` タスク詳細情報格納フォルダ

`reregister` オプションを指定したときに、必ず指定します。再登録したい指定日時実行タスクまたは定期実行タスクの情報が格納された、タスク詳細情報格納フォルダのパスを、絶対パスまたは相対パスで指定します。ただし、指定できるのはローカルディスクのフォルダだけです。指定できる文字数は、絶対パスの長さで 190 文字までです。

## `/setoriginalsubmitter`

タスクを一括再登録する場合にこのオプションを指定すると、一括再登録をした実行者ではなく、元のタスクの実行者を、再登録後のタスク実行者として表示できます。元のタスクの実行者とは、`listtasks` コマンドでタスク詳細情報を出力した時点でのタスクの実行者を指します。再登録後のタスクの実行者は、[タスク] 画面の [実行者] 列に表示されるユーザー ID で確認できます。`listtasks` コマンドでタスク詳細情報を出力した時点でのタスクの実行者は、タスク詳細情報格納フォルダに出力された `listtasks.csv` ファイルの `Submitted By` の列で確認できます。

このオプションを省略した場合は、`submittask` コマンドの `user` オプションに指定したユーザー ID が、再登録後のタスクの実行者となります。

なお、タスクを再登録する時点で、「`listtasks` コマンドでタスク詳細情報を出力した時点でのタスクの実行者」が存在しない場合であってもエラーにはなりません。この場合、タスクの実行者は、「`listtasks` コマンドでタスク詳細情報を出力した時点でのタスクの実行者」になります。

## `/user` ユーザーID

JP1/AO のユーザー ID を指定します。`servicename` オプションで指定したサービスが所属するサービスグループに関連づいたユーザーを指定してください。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、半角英数字、`[!]`、`[#]`、`[$]`、`[%]`、`[&]`、`[']`、`[(, [)]`、`[*]`、`[+]`、`[-]`、`[.]`、`[=]`、`[@]`、`[¥]`、`[^]`、`[_]`、`[|]` です。

大文字と小文字は区別しません。

## `/password` パスワード

`user` オプションで指定した、ユーザーのパスワードを指定します。

指定できる文字数は、1~256 文字です。

使用できる文字の種類は、`user` オプションと同じです。

## `/passwordfile` パスワードファイルパス

`user` オプションで指定したユーザーのパスワードファイルのパスを絶対パスまたは相対パスで指定します。パスワードファイルは `encryptpassword` コマンドで作成します。

## `/wait`

指定した場合、タスクの実行結果（正常終了または失敗）を出力してからコマンドを終了します。`wait` オプションを指定しない場合、タスク実行の終了を待たないで、コマンドが終了します。このとき、タスクの実行が正常に開始された場合にだけ、タスク ID がメッセージで出力されます。



## /scheduledate

サービスを実行する年月日を、「YYYY-MM-DD」の形式で指定します。「YYYY」には西暦を4桁の半角数字で指定します。「MM」には月を1（または01）から12の半角数字で指定します。「DD」には日を1（または01）から31の半角数字で指定します。scheduledate オプションは、scheduleset オプションとあわせて指定します。なお、次のどれかに該当する場合は、コマンドの実行に失敗します。

- 引数の組み合わせが不正な場合  
引数の組み合わせについては「表 1-10 submittask コマンドの引数の組み合わせ」を参照してください。
- 指定した日付の形式が不正な場合
- scheduleset オプションとの組み合わせで決まる日時が過去を指定している場合
- 指定した日付が1994年1月1日から2036年12月31日の範囲外の場合

## /scheduleset

サービスを実行する時分を、「hh:mm」の形式で指定します。「hh」には時を00から23の半角数字で指定します。「mm」には分を00から59の半角数字で指定します。scheduleset オプションは、scheduledate オプションとあわせて指定します。なお、次のどれかに該当する場合は、コマンドの実行に失敗します。

- 引数の組み合わせが不正な場合  
引数の組み合わせについては「表 1-10 submittask コマンドの引数の組み合わせ」を参照してください。
- 指定した時刻の形式が不正な場合
- scheduledate オプションとの組み合わせで決まる日時が過去を指定している場合

/recurrencepattern {daily[:{1h|2h|3h|4h|6h|8h|12h|24h}] | weekly:sun,mon,...,sat | monthly:  
{dd,dd,...,dd [,endofmonth] | endofmonth}}

サービスを定期実行する周期を指定します。recurrencepattern オプションは、recurrencetime オプションおよびrecurrencestart オプションとあわせて指定します。なお、次のどちらかに該当する場合は、コマンドの実行に失敗します。

- 引数の組み合わせが不正な場合  
引数の組み合わせについては「表 1-10 submittask コマンドの引数の組み合わせ」を参照してください。
- 指定した定期実行周期の形式が誤っている場合

定期実行周期の種別は、毎日、毎週、毎月の3つがあります。定期実行周期の書式は、定期実行周期の種別ごとに異なります。

### 毎日

1日に1回実行する場合は、dailyを指定します。

定期実行の間隔を時間単位で指定して実行する場合は、daily:{1h|2h|3h|4h|6h|8h|12h|24h}の形式で指定します。先頭にdaily:を指定し、時間間隔を1h, 2h, 3h, 4h, 6h, 8h, 12h, 24hの中から指定します。

## 毎週

`weekly:sun,mon,...,sat` の形式で指定します。

先頭に `weekly:` を指定し、サービスを実行する曜日を「,」区切りで指定します。曜日の略称には、`sun, mon, tue, wed, thu, fri, sat` を使用します。曜日を指定する際、順序は問いません。同じ曜日が複数回指定されている場合は、引数不正のエラーとなります。

## 毎月

`monthly:{dd,dd,...,dd [,endofmonth] | endofmonth}` を指定します。

先頭に `monthly:` を指定し、サービスを実行する日を「,」区切りで指定します。月末にサービスを実行する場合は、`endofmonth` を指定します。日を指定する際、順序は問いません。ただし、日と月末を同時に指定する場合は、`endofmonth` を最後に指定します。日を指定する場合、1（または01）から31の半角数字で指定します。なお、次の場合は、引数不正のエラーとなります。

- 同じ日が複数回指定されている場合
- 0以下や32以上などの存在しない日を指定した場合
- `endofmonth` が最後に指定されていない場合

日の指定で存在しない月の定期実行はスキップされます。例えば毎月30や31日を指定したとき、2月の定期実行はスキップされます。

## `/recurrencetime hh:mm`

サービスを定期実行する時分を、`[hh:mm]` の形式で指定します。`[hh]` には時を00から23の半角数字で指定します。`[mm]` には分を00から59の半角数字で指定します。`recurrencetime` オプションは、`recurrencepattern` オプションおよび `recurrencestart` オプションとあわせて指定します。なお、次のどちらかに該当する場合は、コマンドの実行に失敗します。

- 引数の組み合わせが不正な場合  
引数の組み合わせについては「[表 1-10 submittask コマンドの引数の組み合わせ](#)」を参照してください。
- 指定した時刻の形式が不正な場合

## `/recurrencestart yyyy-mm-dd`

サービスを定期実行する際の適用開始日を「`yyyy-mm-dd`」の形式で指定します。「`yyyy`」には西暦を4桁の半角数字で指定します。「`mm`」には月を1（または01）から12の半角数字で指定します。「`dd`」には日を1（または01）から31の半角数字で指定します。`recurrencestart` オプションは、`recurrencepattern` オプションおよび `recurrencetime` オプションとあわせて指定します。なお、次のどれかに該当する場合は、コマンドの実行に失敗します。

- 引数の組み合わせが不正な場合  
引数の組み合わせについては「[表 1-10 submittask コマンドの引数の組み合わせ](#)」を参照してください。
- 指定した年月日の形式が不正な場合
- 指定した年月日が1994年1月1日から2036年12月31日の範囲外である場合



## submittask コマンドの引数の組み合わせ

表 1-10 submittask コマンドの引数の組み合わせ

オプション	サービスの即時実行	サービスの指定日時実行	サービスの定期実行	スケジュールされたタスクの一括再登録
/servicename	◎	◎	◎	×
/servicegroup	○	○	○	×
/taskname	○	○	○	×
/taskdescription	○	○	○	×
/property※1	○	○	○	×
/propertyfile※1	○	○	○	×
/reregister	×	×	×	◎
/taskdetaildir	×	×	×	◎
/setoriginalsubmitter	×	×	×	○
/user	◎	◎	◎	◎
/password※2	◎	◎	◎	◎
/passwordfile※2	◎	◎	◎	◎
/wait	○	×	×	×
/scheduledate	×	◎	×	×
/schedulesettime	×	◎	×	×
/recurrencepattern	×	×	◎	×
/recurrencetime	×	×	◎	×
/recurrencestart	×	×	◎	×

### (凡例)

◎：必ず指定します。省略した場合は引数エラーになります。

○：省略できます。

×：指定できません。指定した場合は引数エラーになります。

### 注※1

property オプションまたはpropertyfile オプションのどちらか一方を指定してください。両方を同時に指定するとエラーになります。

### 注※2

password オプションまたはpasswordfile オプションのどちらか一方を指定してください。両方を同時に指定するとエラーになります。

## 格納先

Windows の場合

`JPI/AO` のインストール先フォルダ¥bin

Linux の場合

`/opt/jplao/bin`

## 実行権限

OS の Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

コマンド実行ユーザーが所属するユーザーグループから、実行したいサービスのサービスグループに対して Admin ロール、Develop ロール、Modify ロールまたは Submit ロールが設定されている必要があります。ロールが設定されていないサービスグループのサービスについては実行できません。

user オプションに指定するユーザーに必要な権限を次に示します。

## サービスを実行する場合

user オプションに指定するユーザーが所属するユーザーグループから、操作対象のサービスグループに対して、Admin ロール、Develop ロール、Modify ロールまたは Submit ロールが設定されている必要があります。それぞれ、実行権限のあるサービスだけを実行できます。

## タスクを一括再登録する場合

user オプションに指定するユーザーには、Admin ロールが設定されている必要があります。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
5	通信に失敗しました。
6	認証に失敗しました。
7	パスの指定が不正です。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。

戻り値	説明
14	コマンドの実行権限がありません。
130	サービスの開始に失敗しました。
131	プロパティファイルが存在しません。
132	プロパティファイルの形式が不正です。
133	タスクの状態取得に失敗しました。(wait オプション指定時)
134	タスクの実行に失敗しました。(wait オプション指定時)
136	タスク詳細情報格納フォルダのデータ形式が不正です。
137	予定されたタスクの一括再登録が一部失敗しました。
138	予定されたタスクの一括再登録がすべて失敗しました。
139	タスク詳細情報格納フォルダを出力した JP1/AO と、インストールされている JP1/AO のバージョンまたはリビジョンが異なります。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、サービス名で指定したサービスにプロパティおよびプロパティ値を指定して実行する場合  

```
submittask /servicename service01 /user user01 /password pass01 /property keyA valueA /property keyB "value B" /property keyC valueC,valueD
```
- Windows で、サービスグループとサービス名で指定したサービスに対して、タスク名、タスクの説明、およびプロパティファイルを指定して実行する場合  

```
submittask /servicename service02 /servicegroup servicegroupA /taskname task02 /taskdescription testtask /propertyfile C:¥properties.txt /user user02 /password pass02
```
- Windows で、タスクの実行結果を出力してからコマンドが終了するようにする場合  

```
submittask /servicename service03 /user user03 /password pass03 /wait
```
- Windows で、日時を指定してサービスを実行する場合  

```
submittask /servicename service04 /user user04 /password pass04 /scheduledate 2014-01-01 /schedulesettime 15:30
```
- Windows で、定期実行する場合  

```
submittask /servicename service05 /user user05 /password pass05 /recurrencepattern weekly:sun,mon,fri /recurrencetime 15:30 /recurrencestart 2013-06-17
```
- Windows で、予定されたタスクを一括再登録する場合  

```
submittask /reregister /taskdetaildir "C:¥data¥taskdetail" /user user06 /password pass06
```

---

## 関連項目

- 1.6.9 listtasks (タスクの一覧出力, タスク詳細情報出力)
  - 1.3 コマンドの引数に指定できる文字
  - マニュアル「JP1/AO 構築ガイド」－「ユーザー設定プロパティファイル (config\_user.properties)」のトピック
  - マニュアル「JP1/AO 運用ガイド」－「プロパティファイルについて」のトピック
-

## 1.7 保守・メンテナンス関連のコマンド

---

### 1.7.1 backupsystem (JP1/AO システムのバックアップ)

#### 機能

JP1/AO の設定情報やデータベース情報をバックアップして指定したフォルダにデータを格納します。

#### 形式

```
backupsystem
  /dir バックアップデータのパス
  [/auto]
```

#### 引数

*/dir* バックアップデータのパス

バックアップデータを収集する空のフォルダを絶対パスまたは相対パスで指定します。指定できるのはローカルディスクのフォルダだけです。指定するパスは、50 文字以内をお勧めします。

*/auto*

JP1/AO, JP1/OA および Hitachi Command Suite 製品のサービスとデータベースの停止・開始を自動で実行します。このオプションを省略した場合は、JP1/AO, JP1/OA および Hitachi Command Suite 製品のサービスとデータベースの停止・開始を自動で実行しません。

なお、クラスタ環境でこのオプションを指定する場合は、クラスタソフトに登録したサービスがオフラインである必要があります。

#### 格納先

Windows の場合

*JP1/AO* のインストール先フォルダ¥bin

Linux の場合

*/opt/jp1ao/bin*

#### 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

#### 注意事項

- バックアップファイルを格納するフォルダに十分な空き容量があることを確認してください。必要な空き容量は次のとおりです。

バックアップ対象ファイルの容量+20MB

バックアップ対象ファイルの容量の目安としては、以下の各フォルダの容量を合計して下さい※1。

表 1-11 Windows の非クラスタ構成の場合

製品	フォルダ
JP1/AO	JP1/AO のインストール先フォルダ¥conf 配下すべて
	JP1/AO のインストール先フォルダ¥contents 配下すべて
	JP1/AO のインストール先フォルダ¥system 配下すべて
	JP1/AO のインストール先フォルダ¥data 配下すべて
	JP1/AO のインストール先フォルダ¥develop 配下すべて
共通コンポーネント	システムドライブ¥Program Files¥Hitachi¥HiCommand ¥database※2 配下すべて

表 1-12 Windows のクラスタ構成の場合

製品	フォルダ
JP1/AO	共有フォルダ名¥jp1ao¥conf 配下すべて
	共有フォルダ名¥jp1ao¥contents 配下すべて
	共有フォルダ名¥jp1ao¥system 配下すべて
	共有フォルダ名¥jp1ao¥data 配下すべて
	共有フォルダ名¥jp1ao¥develop 配下すべて
共通コンポーネント	共有フォルダ名¥Base64¥database 配下すべて

表 1-13 Linux の非クラスタ構成の場合

製品	フォルダ
JP1/AO	/opt/jp1ao/conf 配下すべて
	/var/opt/jp1ao/contents 配下すべて
	/var/opt/jp1ao/data 配下すべて
	/var/opt/jp1ao/develop 配下すべて
	/opt/jp1ao/system 配下すべて
共通コンポーネント	/var/opt/HiCommand/database 配下すべて

表 1-14 Linux のクラスタ構成の場合

製品	フォルダ
JP1/AO	共有フォルダ名/jp1ao/conf 配下すべて
	共有フォルダ名/jp1ao/contents 配下すべて

製品	フォルダ
JP1/AO	共有フォルダ名/jp1ao/data 配下すべて
	共有フォルダ名/jp1ao/develop 配下すべて
	共有フォルダ名/jp1ao/system 配下すべて
共通コンポーネント	共有フォルダ名/Base64/database 配下すべて

#### 注※1

Hitachi Command Suite 製品がインストールされている場合は、これらのほかに該当する Hitachi Command Suite 製品分のファイルやフォルダもバックアップされます。

#### 注※2

JP1/AO のデータベースの格納先フォルダを変更すると、「システムドライブ¥Program Files¥Hitachi ¥HiCommand¥database」の部分も変更されます。

- このコマンドでは、次のファイルはバックアップされません。必要に応じて手動でバックアップしてください。
  - https 接続用の SSL サーバ証明書ファイル
  - https 接続用の秘密鍵ファイル
  - 公開鍵認証用の秘密鍵ファイル
  - 作成したクラスタサービス制御コマンド（Linux でクラスタ構成の場合）
- auto オプションを指定しない場合、JP1/AO のサービスが停止していることを確認してからこのコマンドを実行してください。

停止していない場合は、hcnds64srv コマンドにstop オプションを指定して実行し、サービスを停止してください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
7	パスの指定が不正です。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。

戻り値	説明
11	指定したフォルダが空ではありません。
14	コマンドの実行権限がありません。
100	バックアップの実行に失敗しました。
101	サービスの開始/停止に失敗しました。
103	スケジューラーデータベースへのアクセスエラーが発生しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、指定したバックアップフォルダ (C:¥Users¥Backup) にデータをバックアップする場合のコマンドの使用例を示します。

```
backupsystem /dir "C:¥Users¥Backup" /auto
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)

## 1.7.2 hcnds64dbrepair (データベースの再作成)

### 機能

すべてのデータベースを強制削除して再作成し、hcnds64dbtrans コマンドで取得したバックアップデータからデータベースを回復します。このコマンドは、データベースが破損してrestoresystem コマンド、およびhcnds64dbtrans コマンドのimport オプションを使用しても復旧できない場合に使用します。

### 形式

```
hcnds64dbrepair
/trans バックアップデータ
```

### 引数

/trans バックアップデータ

hcnds64dbtrans コマンドで取得したデータベースのバックアップデータを指定します。バックアップデータには、hcnds64dbtrans コマンドのworkpath オプションまたはfile オプションに指定したパスを指定してください。



## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 注意事項

- hcms64dbrepair コマンドを実行する前に、JP1/AO システムを停止してください。
- hcms64dbrepair コマンドの実行後は、JP1/AO システムを開始してください。
- バックアップデータを展開するために必要な領域として「共通コンポーネントのインストール先フォルダ¥tmp」または「var/opt/HiCommand/Base64/tmp」のフォルダを使用します。バックアップデータの容量に応じて、展開するために必要な領域を確保してください。
- コマンドの実行後、ビルトインアカウント（System アカウント）のパスワードが初期値に戻ります。必要に応じて、パスワードを変更してください。
- クラスタシステムの場合、実行系ホストでコマンドを実行してください。待機系ホストでは実行できません。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
245	データベースのインポートに失敗しました。
246	定義ファイルが不正です。
247	データベースのアンセットアップに失敗しました。
248	サービスまたはデータベースの停止に失敗しました。
249	待機系ノードのため実行できません。
250	バックアップデータが不正（ファイル不足、アーカイブファイルの展開失敗）です。
251	製品不一致、製品バージョン不一致のため中断しました。
252	データベースのセットアップに失敗しました。

戻り値	説明
253	データベースのサービスの開始に失敗しました。
254	データベースがセットアップされていないためデータベースを再作成できません。
255	異常終了しました。

## 使用例

Windows で、すべてのデータベースを強制削除して再作成し、バックアップデータからデータベースを回復する場合のコマンドの使用例を示します。

```
hcnds64dbrepair /trans C:¥bkfile1
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
- マニュアル「JP1/AO 運用ガイド」－「JP1/AO システムを開始する手順（非クラスタ構成の場合）」のトピック
- マニュアル「JP1/AO 運用ガイド」－「JP1/AO システムを開始する手順（クラスタ構成の場合）」のトピック
- マニュアル「JP1/AO 運用ガイド」－「JP1/AO システムを停止する手順（非クラスタ構成の場合）」のトピック
- マニュアル「JP1/AO 運用ガイド」－「JP1/AO システムを停止する手順（クラスタ構成の場合）」のトピック
- [1.6.4 hcnds64srv \(JP1/AO の開始・停止・状態表示\)](#)

## 1.7.3 hcnds64dsrv (データベースの開始・停止)

### 機能

JP1/AO のデータベースを開始および停止します。データベースのメンテナンスを実施するときに使用します。

### 形式

```
hcnds64dsrv
  {/start | /stop}
```

### 引数

/start

データベースを開始します。

/stop

データベースを停止します。

## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 注意事項

データベースのメンテナンス手順以外では使用しないでください。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	開始・停止の要求を受け付けました。
254	データベースが初期化されていません。
255	失敗しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、JP1/AO のデータベースを開始する場合  
hcnds64dbsrv /start
- Windows で、JP1/AO のデータベースを停止する場合  
hcnds64dbsrv /stop

---

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)
-

## 1.7.4 hcnds64dbtrans (データベースのバックアップ・リストア)

### 機能

JP1/AO のデータベースをバックアップ (エクスポート) またはリストア (インポート) します。JP1/AO のデータベースを再編成する場合に使用します。

### 形式

JP1/AO のデータベースをバックアップ (エクスポート) する場合：

```
hcnds64dbtrans
  /export
  /workpath 作業用フォルダパス
  /file アーカイブファイルパス
  [/auto]
```

JP1/AO のデータベースをリストア (インポート) する場合：

```
hcnds64dbtrans
  /import
  /type Automation
  /workpath 作業用フォルダパス
  [/file アーカイブファイルパス]
  [/auto]
```

### 引数

#### /export

データベースをエクスポートします。

#### /workpath 作業用フォルダパス

エクスポートまたはインポートするときに一時的に使用する作業用フォルダを絶対パスで指定します。指定できるのはローカルディスクのフォルダだけです。

エクスポート時や、インポート時にfile オプションを指定する場合、作業用フォルダには空のフォルダを指定してください。

#### /file アーカイブファイルパス

エクスポートまたはインポートするアーカイブファイルのパスを絶対パスで指定します。export オプションを指定する場合は必ず指定します。

出力ファイルサイズが2GBを超える場合や、アーカイブファイルを作成する先のディスク容量が不足している場合、アーカイブファイルは作成されません。

#### /auto

JP1/AO, JP1/OA および Hitachi Command Suite 製品のサービスとデータベースの停止・開始を自動で実行します。このオプションを省略した場合は、JP1/AO, JP1/OA および Hitachi Command Suite 製品のサービスとデータベースの停止・開始を自動で実行しません。

`/import`

データベースをインポートします。インポート時に既存の認証データはすべて削除されてからインポートされます。

`/type Automation`

データベースをインポートする対象の製品名として「Automation」を指定します。

## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 注意事項

- エクスポート時に、戻り値「3」が出力された場合、データベースの情報はworkpath オプションに指定したディレクトリに残っています。  
これらの情報をインポートしたい場合は、workpath オプションに、エクスポート時にworkpath オプションに指定したディレクトリを指定してください。このとき、エクスポート時にworkpath オプションに指定したディレクトリのフォルダ構成を変更しないでください。また、インポートするとき、file オプションには何も指定しないでください。
- 次の場合、workpath オプションに指定したディレクトリは空になり、コマンドが終了します。
  - エクスポート時に、戻り値「1」、「2」、「233」、「234」、「235」、「237」、「238」、「239」、「240」または「255」が出力された場合
  - インポート時に、戻り値「3」が出力された場合

## 戻り値

export オプション指定時のコマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	製品のバージョンの取得に失敗しました。
2	データベースが開始していません。
3	データベースのアーカイブに失敗しました。
4	作業用フォルダが空ではありません。

戻り値	説明
233	データベースの再起動が中断している状態です。
234	データベースのサービスが停止しているか、または存在しません。
235	データベースが初期化されていません。
237	Hitachi Command Suite 製品またはデータベースの開始に失敗しました。
238	Hitachi Command Suite 製品またはデータベースの停止に失敗しました。
239	データベースの開始に失敗しました。
240	データベースの停止に失敗しました。
255	異常終了しました。

`import` オプション指定時のコマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	製品のバージョンの取得に失敗しました。
2	データベースが開始していません。
3	アーカイブファイルの展開に失敗しました。
4	作業用フォルダが空ではありません。
5	指定された製品がアーカイブファイルに存在しません。
6	指定された製品がインストールされていません。
7	インポートできないバージョンの製品があります。
8	作業用フォルダにインポート用のデータがありません。または、インポート用のデータの形式が不正です。
9	プライマリーサーバに、セカンダリーサーバのデータをインポートしようとして失敗しました。
10	セカンダリーサーバに、プライマリーサーバのデータをインポートしようとして失敗しました。
11	使用中のデータベースにデータをインポートしようとして失敗しました。
233	データベースの再起動が中断している状態です。
234	データベースのサービスが停止しているか、または存在しません。
235	データベースが初期化されていません。
237	Hitachi Command Suite 製品またはデータベースの開始に失敗しました。
238	Hitachi Command Suite 製品またはデータベースの停止に失敗しました。
239	データベースの開始に失敗しました。
240	データベースの停止に失敗しました。

戻り値	説明
255	異常終了しました。

## 使用例

次の場合のコマンドの使用例を示します。

- Windows で、JP1/AO のデータベースをバックアップする場合  
`hcnds64dbtrans /export /workpath "C:¥Users¥workfolder" /file "C:¥backup¥arcfile01" /auto`
- Windows で、JP1/AO のデータベースをリストアする場合  
`hcnds64dbtrans /import /type Automation /workpath "C:¥Users¥workfolder" /file "C:¥backup ¥arcfile01" /auto`

---

## 関連項目

- 1.3 コマンドの引数に指定できる文字
- 

## 1.7.5 hcnds64getlogs (ログ情報の採取)

### 機能

JP1/AO の運用中に出力されるログ情報を採取してアーカイブファイルに出力します。

### 形式

```
hcnds64getlogs
/dir 出力フォルダパス
[/types Automation]
[/arc アーカイブファイル名]
[/logtypes {log | db | csv}]
```

### 引数

`/dir 出力フォルダパス`

アーカイブファイルを出力するフォルダパスを指定します。指定できるのはローカルディスクのフォルダだけです。

出力フォルダパスには空のフォルダを絶対パスまたは相対パスで指定してください。指定したフォルダパスが存在しない場合はフォルダが作成されます。

指定できるパスの最大長は 100 文字です。なお、このオプションで指定したフォルダには書き込み許可が設定されます。

## /types Automation

ログ情報を収集する対象の製品名として「Automation」を指定します。JP1/AO サーバの OS が Windows の場合、大文字と小文字は区別しません。JP1/AO サーバの OS が Linux の場合、大文字と小文字を区別します。このオプションを省略した場合、JP1/AO およびインストールされているすべての Hitachi Command Suite 製品が対象となります。なお、この場合、ログの収集に時間がかかることがあります。

## /arc アーカイブファイル名

共通コンポーネントの資料採取ツールの実行結果として作成するアーカイブファイルの名称を指定します。このオプションを指定しない場合、アーカイブファイル名は「HiCommand\_log」になります。

アーカイブファイルは、dir オプションで指定したフォルダの下に出力されます。

アーカイブファイル名として指定できる文字は、ASCII 印字可能文字コード (0x20~0x7E) の中で、一部の特殊文字 (「¥」, 「/」, 「:」, 「,」, 「;」, 「\*」, 「?」, 「"」, 「<」, 「>」, 「|」, 「\$」, 「%」, 「&」, 「'」, 「」) を除いたものです。なお、拡張子の指定は不要です。

## /logtypes {log | db | csv}

取得したい共通コンポーネントのログファイルの種別を指定します。ログファイル種別と取得できるログファイルの対応を次の表に示します。

表 1-15 ログファイル種別と取得できるログファイル

ログファイル種別	取得できるログファイル (Windows の場合)	取得できるログファイル (Linux の場合)
log	<ul style="list-style-type: none"><li>arc オプションで指定したアーカイブファイル名.jar</li><li>arc オプションで指定したアーカイブファイル名.hdb.jar</li></ul>	<ul style="list-style-type: none"><li>/arc オプションで指定したアーカイブファイル名_64.jar</li><li>arc オプションで指定したアーカイブファイル名_64.hdb.jar</li></ul>
db	arc オプションで指定したアーカイブファイル名.db.jar	arc オプションで指定したアーカイブファイル名_64.db.jar
csv	arc オプションで指定したアーカイブファイル名.csv.jar	arc オプションで指定したアーカイブファイル名_64.csv.jar

このオプションを省略した場合は、共通コンポーネントのすべてのログファイルが取得されます。そのため、オプションを省略して、実行することをお勧めします。

複数のログファイル種別を指定する場合は、「logtypes log db csv」のように半角空白で区切ってください。なお、types オプションと logtypes オプションを同時に使用する場合は、logtypes オプションの値に必ず「log」を指定してください。

## 出力形式

コマンドを実行した結果、採取する資料の一覧を次に示します。

なお、ファイルの内容および出力形式については公開していません。



表 1-16 採取する資料一覧 (JP1/AO サーバの OS が Windows の場合)

アーカイブファイル	出力結果
/dir オプションで指定した出力先フォルダ¥Automation_1st_log.jar	<ul style="list-style-type: none"> <li>• JP1/AO のインストール先フォルダ¥logs 直下のすべてのファイル (サブフォルダは含まれません)</li> <li>• JP1/AO のインストール先フォルダ¥data¥task 配下のすべてのファイル</li> </ul>
/dir オプションで指定した出力先フォルダ¥Automation_log.jar	<ul style="list-style-type: none"> <li>• FILELIST.txt</li> <li>• JP1/AO のインストール先フォルダ¥conf 配下のすべてのファイル</li> <li>• JP1/AO のインストール先フォルダ¥data 配下のすべてのファイル</li> <li>• JP1/AO のインストール先フォルダ¥logs 配下のすべてのファイル</li> <li>• JP1/AO のインストール先フォルダ¥work 配下のすべてのファイル</li> <li>• Windows フォルダ**1¥Temp¥HITACHI_JP1_INST_LOG 配下のすべてのファイル</li> <li>• Windows フォルダ**1¥Temp¥jplcommon 配下のすべてのファイル</li> <li>• Program Files フォルダ**2¥InstallShield Installation Information¥{C4F6D00E-A9A2-4E57-A21A-B78B63FF1C54}¥setup.ini</li> <li>• Program Files フォルダ**2¥InstallShield Installation Information¥{C4F6D00E-A9A2-4E57-A21A-B78B63FF1C54}¥setup.ilg</li> <li>• REGDATA.DAT</li> </ul>
/dir オプションで指定した出力先フォルダ¥/arc オプションで指定したアーカイブファイル名.jar	共通コンポーネントの資料採取ツール (hcnds64get logs, hcnds64ras) の実行結果
/dir オプションで指定した出力先フォルダ¥/arc オプションで指定したアーカイブファイル名.hdb.jar	共通コンポーネントの資料採取ツール (hcnds64get logs) の実行結果
/dir オプションで指定した出力先フォルダ¥/arc オプションで指定したアーカイブファイル名.db.jar	共通コンポーネントの資料採取ツール (hcnds64get logs) の実行結果
/dir オプションで指定した出力先フォルダ¥/arc オプションで指定したアーカイブファイル名.csv.jar	共通コンポーネントの資料採取ツール (hcnds64get logs) の実行結果

注※1

Windows フォルダは、デフォルトの場合は C:¥WINDOWS となります。

注※2

Program Files フォルダは、デフォルトの場合は C:¥Program Files となります。

表 1-17 採取する資料一覧 (JP1/AO サーバの OS が Linux の場合)

アーカイブファイル	出力結果
dir オプションで指定した出力先フォルダ/arc オプションで指定したアーカイブファイル名_64.jar	<ul style="list-style-type: none"> <li>• FILELIST.txt</li> <li>• /opt/jplao/conf 配下のすべてのファイル</li> <li>• /var/opt/jplao/data 配下のすべてのファイル</li> </ul>

アーカイブファイル	出力結果
<i>dir</i> オプションで指定した出力先フォルダ/ <i>arc</i> オプションで指定したアーカイブファイル名_64.jar	<ul style="list-style-type: none"> <li>• /var/opt/jplao/logs 配下のすべてのファイル</li> <li>• /opt/jplao/tools 配下のすべてのファイル</li> <li>• /var/opt/jplao/work 配下のすべてのファイル</li> <li>• /tmp/ HITACHL_JP1_INST_LOG 配下のすべてのファイル</li> <li>• 共通コンポーネントの資料採取ツール (hcnds64getlogs, hcnds64ras) の実行結果</li> </ul>
<i>dir</i> オプションで指定した出力先フォルダ/ <i>arc</i> オプションで指定したアーカイブファイル名_64.hdb.jar	共通コンポーネントの資料採取ツール (hcnds64get logs) の実行結果
<i>dir</i> オプションで指定した出力先フォルダ/ <i>arc</i> オプションで指定したアーカイブファイル名_64.db.jar	共通コンポーネントの資料採取ツール (hcnds64get logs) の実行結果
<i>dir</i> オプションで指定した出力先フォルダ/ <i>arc</i> オプションで指定したアーカイブファイル名_64.csv.jar	共通コンポーネントの資料採取ツール (hcnds64get logs) の実行結果

## 格納先

Windows の場合

共通コンポーネントのインストール先フォルダ¥bin

Linux の場合

/opt/HiCommand/Base64/bin

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。

## 注意事項

- このコマンドは実行中に中断しないでください。
- このコマンドの実行が中断した場合、*dir* オプションで指定するフォルダに十分な空き容量がないため hcnds64get logs コマンドが途中で終了しています。*dir* オプションで指定するフォルダに十分な空き容量を確保したあとで、再度このコマンドを実行してください。
- このコマンドは同時に複数実行しないでください。
- JP1/AO をクラスタ構成で運用している場合は、実行系ホストと待機系ホストそれぞれでこのコマンドを実行してください。  
JP1/AO サーバが停止していてもこのコマンドは実行できます。そのため、クラスタ構成で障害が起きた場合も、系切り替えを行わないでログ情報を採取できます。ただし、データベースが停止している場合、データベースの情報は取得できません。
- 同じオプションを 2 回以上指定した場合、最初に指定したオプションが有効となります。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	異常終了しました。

## 使用例

Windows で、指定したフォルダ (C:¥Users¥folder01) にログ情報を収集する場合のコマンドの使用例を示します。

```
hcnds64getlogs /dir "C:¥Users¥folder01" /types Automation /arc A0_log
```

## 関連項目

- [1.3 コマンドの引数に指定できる文字](#)

## 1.7.6 restoresystem (JP1/AO システムのリストア)

### 機能

backupsystem コマンドで取得した JP1/AO の設定情報やデータベース情報のバックアップデータをリストアします。

コマンド実行時にリストアするデータを次に示します。

- タスク※
- デバッグタスク※
- タスクの履歴
- サービス
- デバッグサービス
- サービステンプレート
- ユーザー
- ユーザーグループ
- サービスグループ
- エージェントレス接続先定義

- サービス共有プロパティ
- 各種定義ファイル

#### 注※

リストア後のタスクおよびデバッグタスクの状態は次のように変更されます。

表 1-18 バックアップ時とリストア後のタスクおよびデバッグタスクの状態

バックアップ時のタスクおよびデバッグタスクの状態	リストア後のタスクおよびデバッグタスクの状態
待機中	キャンセル（タスクおよびデバッグタスクの終了日時にはリストアした日時が設定されます）
保留中	
実行中	失敗
応答待ち中	
異常検出	
停止中	
正常終了	正常終了
失敗	失敗
キャンセル	キャンセル

## 形式

```
restoresystem
  /dir バックアップデータのパス
  [/auto]
```

## 引数

### /dir バックアップデータのパス

backupsystem コマンドで指定したバックアップデータが格納されているバックアップフォルダのパスを絶対パスまたは相対パスで指定します。

### /auto

JP1/AO, JP1/OA および Hitachi Command Suite 製品のサービスとデータベースの停止・開始を自動で実行します。このオプションを省略した場合は、JP1/AO, JP1/OA および Hitachi Command Suite 製品のサービスとデータベースの停止・開始を自動で実行しません。

なお、クラスタ環境でこのオプションを指定する場合は、クラスタソフトに登録したサービスがオフラインである必要があります。

## 格納先

Windows の場合

`JP1/AO` のインストール先フォルダ¥bin

Linux の場合

`/opt/jplao/bin`

## 実行権限

Administrators 権限または root 権限を持つユーザーで実行してください。Administrators 権限または root 権限がない場合は、権限昇格を求めるメッセージが表示されます。

## 注意事項

- このコマンドは実行時に一時ファイルが作成されます。このため、バックアップファイルを格納するフォルダに十分な空き容量があることを確認してください。必要な空き容量は次のとおりです。  
バックアップ対象ファイルの容量+20MB
- このコマンドでは、次のファイルはリストアされません。必要に応じて手動で再設定してください。
  - https 接続用の SSL サーバ証明書ファイル
  - https 接続用の秘密鍵ファイル
  - 公開鍵認証用の秘密鍵ファイル
  - 作成したクラスタサービス制御コマンド (Linux でクラスタ構成の場合)

https 接続用のファイルは `user_httpsd.conf` ファイル、公開鍵認証用のファイルはユーザー設定プロパティファイル (`config_user.properties`) で定義されている、それぞれの格納先に配置してください。

- `auto` オプションを指定しない場合、JP1/AO のサービスが停止していることを確認してからこのコマンドを実行してください。

停止していない場合は、`hcmds64srv` コマンドに `stop` オプションを指定して実行し、サービスを停止してください。

- JP1/AO のインストールパスに半角スペースが含まれる場合、半角の直前までのパスと同名のパスのファイルやフォルダが存在すると、`restoresystem` コマンドの実行が戻り値 114 で失敗します。この現象が発生した場合は、該当ファイルまたはフォルダを別のパスに移動してください。その後、JP1/AO を再インストールし、`restoresystem` コマンドを再度実行してください。

(例) JP1/AO を "C:¥Program Files¥HITACHI¥JP1AO" にインストールしていて、"C:¥Program" のファイルまたはフォルダが存在した場合 `restoresystem` コマンドの実行に失敗します。

- リストアしたタスクおよびデバッグタスクは、次の操作を選択できません。
  - 失敗したステップからリトライ
  - 失敗した次のステップからリトライ
- リストアしたタスクおよびデバッグタスクは、ステップの進捗や状態が表示されません。

## 戻り値

コマンドの戻り値を次の表に示します。

戻り値	説明
0	正常に終了しました。
1	引数が不正です。
2	コマンドの実行を中断しました。
3	サービスの状態が不正です。
4	ほかのコマンドを実行中です。
7	パスの指定が不正です。
9	指定したパスが存在しません。
10	指定したパスにアクセスできません。
14	コマンドの実行権限がありません。
110	リストアの実行に失敗しました。
111	サービスの開始/停止に失敗しました。
113	バックアップファイルが不正です。
114	スケジューラーデータベースへのアクセスエラーが発生しました。
255	そのほかのエラーでコマンドの実行が中断しました。

## 使用例

Windows で、指定したバックアップフォルダ (C:¥Users¥Backup) のデータをリストアする場合のコマンドの使用例を示します。

```
restoresystem /dir C:¥Users¥Backup /auto
```

## 関連項目

- [1.7.1 backupsystem \(JP1/AO システムのバックアップ\)](#)
- [1.3 コマンドの引数に指定できる文字](#)
- マニュアル「JP1/AO 構築ガイド」－「ユーザー設定プロパティファイル (config\_user.properties)」のトピック

# 2

## API

JP1/AO が提供する API について説明します。API との通信には、http プロトコルまたは https プロトコルを使用できます。

## 2.1 API 一覧

JP1/AO で使用できる API の一覧を次の表に示します。

表 2-1 サービステンプレート機能の API 一覧

API 名	機能	参照先
サービステンプレート一覧の取得	JP1/AO に登録されているサービステンプレートの一覧を取得します。	2.4.1 サービステンプレート一覧の取得
サービステンプレートの情報取得	指定したサービステンプレートの情報を取得します。	2.4.2 サービステンプレートの情報取得
サービステンプレートの削除	指定したサービステンプレートを削除します。	2.4.3 サービステンプレートの削除
サービステンプレートに対する操作一覧の取得	指定したサービステンプレートに対して実行できる操作の一覧を取得します。	2.4.4 サービステンプレートに対する操作一覧の取得
サービステンプレートのインポートに必要な HTML ファイルの取得	指定したサービステンプレートのインポートに必要な HTML ファイルを取得します。	2.4.5 サービステンプレートのインポートに必要な HTML ファイルの取得
サービステンプレートのインポート	指定したサービステンプレートをインポートします。	2.4.6 サービステンプレートのインポート
サービステンプレートのエクスポートに必要な情報の取得	指定したサービステンプレートのエクスポートに必要な情報を取得します。	2.4.7 サービステンプレートのエクスポートに必要な情報の取得
サービステンプレートのエクスポート	指定したサービステンプレートをエクスポートします。	2.4.8 サービステンプレートのエクスポート
サービステンプレートの詳細説明を表示する URL の取得	指定したサービステンプレートの詳細説明を表示する URL を取得します。	2.4.9 サービステンプレートの詳細説明を表示する URL の取得
サービステンプレートを基にしたサービスの作成に必要な情報の取得	指定したサービステンプレートからサービスを作成する場合に必要な情報を取得します。	2.4.10 サービステンプレートを基にしたサービスの作成に必要な情報の取得
サービステンプレートを基にしたサービスの作成	指定したサービステンプレートからサービスを作成します。サービスの作成時に、プロパティを指定できます。	2.4.11 サービステンプレートを基にしたサービスの作成

表 2-2 サービス機能の API 一覧

API 名	機能	参照先
サービス一覧の取得	JP1/AO に登録されているサービスの一覧を取得します。	2.5.1 サービス一覧の取得
サービスの情報取得	指定したサービスの情報を取得します。	2.5.2 サービスの情報取得
サービスの編集	指定したサービスを編集します。	2.5.3 サービスの編集
サービスの削除	指定したサービスを削除します。	2.5.4 サービスの削除



API名	機能	参照先
サービスに対する操作一覧の取得	指定したサービスに対して実行できる操作の一覧を取得します。	2.5.5 サービスに対する操作一覧の取得
サービス実行に必要な情報の取得	指定したサービスの実行操作に必要な情報を取得します。	2.5.6 サービス実行に必要な情報の取得
サービスの実行	指定したサービスを実行します。	2.5.7 サービスの実行
サービスのカウンターのリセットに必要な情報の取得	指定したサービスのカウンターのリセット（統計情報の初期化）に必要な情報を取得します。	2.5.8 サービスのカウンターのリセットに必要な情報の取得
サービスのカウンターのリセット	指定したサービスのカウンターをリセット（統計情報を初期化）します。	2.5.9 サービスのカウンターのリセット
サービスの状態を「リリース」に変更する操作に必要な情報の取得	指定したサービスの状態を「リリース」に変更する操作に必要な情報を取得します。	2.5.10 サービスの状態を「リリース」に変更する操作に必要な情報の取得
サービスの状態を「リリース」に変更	指定したサービスの状態を「リリース」に変更します。	2.5.11 サービスの状態を「リリース」に変更
サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得	指定したサービスの状態を「メンテナンス」に変更する操作に必要な情報を取得します。	2.5.12 サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得
サービスの状態を「メンテナンス」に変更	指定したサービスの状態を「メンテナンス」に変更します。	2.5.13 サービスの状態を「メンテナンス」に変更
サービスの状態を「無効」に変更する操作に必要な情報の取得	指定したサービスの状態を「無効」に変更する操作に必要な情報を取得します。	2.5.14 サービスの状態を「無効」に変更する操作に必要な情報の取得
サービスの状態を「無効」に変更	指定したサービスの状態を「無効」に変更します。	2.5.15 サービスの状態を「無効」に変更
サービスの詳細説明の URL を取得	指定したサービスの詳細説明を表示する URL を取得します。	2.5.16 サービスの詳細説明の URL を取得
サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得	指定したサービスが使用するサービステンプレートのバージョン変更に必要な情報を取得します。	2.5.17 サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得
サービスが使用するサービステンプレートのバージョン変更	指定したサービスに、任意のバージョンのサービステンプレートを適用します。	2.5.18 サービスが使用するサービステンプレートのバージョン変更

表 2-3 スケジュール機能の API 一覧

API名	機能	参照先
スケジュール一覧の取得	タスクに設定されているスケジュール一覧を取得します。	2.6.1 スケジュール一覧の取得
スケジュールの情報取得	指定したスケジュールの情報を取得します。	2.6.2 スケジュールの情報取得

API名	機能	参照先
スケジュールに対する操作一覧の取得	指定したスケジュールに対して実行できる操作の一覧を取得します。	2.6.3 スケジュールに対する操作一覧の取得
スケジュールのキャンセルに必要な情報の取得	指定したスケジュールをキャンセルするために必要な情報を取得します。	2.6.4 スケジュールのキャンセルに必要な情報の取得
スケジュールのキャンセル	指定したスケジュールをキャンセルします。	2.6.5 スケジュールのキャンセル
スケジュールの一時停止に必要な情報の取得	指定したスケジュールを一時停止するために必要な情報を取得します。	2.6.6 スケジュールの一時停止に必要な情報の取得
スケジュールの一時停止	指定したスケジュールを一時停止します。	2.6.7 スケジュールの一時停止
スケジュールの再開に必要な情報の取得	指定したスケジュールを再開するために必要な情報を取得します。	2.6.8 スケジュールの再開に必要な情報の取得
スケジュールの再開	指定したスケジュールを再開します。	2.6.9 スケジュールの再開

表 2-4 タスク機能の API 一覧

API名	機能	参照先
タスク一覧の取得	タスクの一覧を取得します。	2.7.1 タスク一覧の取得
タスクの情報取得	指定したタスクの情報を取得します。	2.7.2 タスクの情報取得
タスクの編集	指定したタスクのノートおよび TODO を編集します。	2.7.3 タスクの編集
タスクの削除	指定したタスクを削除します。指定したタスクがデバッグタスク以外の場合は、アーカイブするための URL を取得します。	2.7.4 タスクの削除
タスク操作一覧の取得	指定したタスクに対して実行できる操作の一覧を取得します。	2.7.5 タスク操作一覧の取得
タスクの実行停止に必要な情報の取得	指定したタスクを実行停止するために必要な情報を取得します。	2.7.6 タスク実行停止に必要な情報の取得
タスクの実行停止	指定したタスクを実行停止します。	2.7.7 タスクの実行停止
タスクの強制停止に必要な情報の取得	指定したタスクを強制停止するために必要な情報を取得します。	2.7.8 タスクの強制停止に必要な情報の取得
タスクの強制停止	指定したタスクを強制停止します。	2.7.9 タスクの強制停止
タスクの再実行に必要な情報の取得	指定したタスクを再実行するために必要な情報を取得します。	2.7.10 タスクの再実行に必要な情報の取得
タスクの再実行	指定したタスクを再実行します。	2.7.11 タスクの再実行
応答待ち中のタスクに応答するために必要な情報の取得	応答待ち中のタスクに応答するために必要な情報を取得します。取得するのは、指定した ID のタスクの	2.7.12 応答待ち中のタスクに応答するために必要な情報の取得

API 名	機能	参照先
応答待ち中のタスクに回答するために必要な情報の取得	中で、応答待ち中になった日時が最も古いステップの情報です。	2.7.12 応答待ち中のタスクに回答するために必要な情報の取得
応答待ち中のタスクに回答	指定した ID のタスクの中で、応答待ち中になった日時が最も古いステップに回答入力します。	2.7.13 応答待ち中のタスクに回答
タスクのリトライに必要な情報の取得 (失敗したステップからリトライ)	タスクを指定し、失敗したステップからリトライするために必要な情報を取得します。	2.7.14 タスクのリトライに必要な情報の取得 (失敗したステップからリトライ)
失敗したステップからリトライ	タスクを指定し、失敗したステップからリトライします。	2.7.15 失敗したステップからリトライ
タスクのリトライに必要な情報の取得 (失敗した次のステップからリトライ)	タスクを指定し、失敗した次のステップからリトライするために必要な情報を取得します。	2.7.16 タスクのリトライに必要な情報の取得 (失敗した次のステップからリトライ)
失敗した次のステップからリトライ	タスクを指定し、失敗した次のステップからリトライします。	2.7.17 失敗した次のステップからリトライ
タスクのアーカイブに必要な情報の取得	指定したタスクをアーカイブするために必要な引数のテンプレートを取得します。	2.7.18 タスクのアーカイブに必要な情報の取得
タスクのアーカイブ	指定したタスクをアーカイブします。	2.7.19 タスクのアーカイブ
ステップ一覧の取得	指定したタスクに含まれるステップのうち、[タスク詳細] 画面に表示されるステップの一覧を取得します。	2.7.20 ステップ一覧の取得
タスクログの取得	指定したタスクのタスクログを取得します。	2.7.21 タスクログの取得

表 2-5 履歴関連の API 一覧

API 名	機能	参照先
履歴一覧の取得	履歴の一覧を取得します。	2.8.1 履歴一覧の取得
履歴の削除 (条件指定)	クエリパラメータの条件に従って、履歴を削除します。	2.8.2 履歴の削除 (条件指定)
履歴の取得	指定した ID の履歴を取得します。	2.8.3 履歴の取得
履歴の削除 (ID 指定)	指定した ID の履歴を削除します。	2.8.4 履歴の削除 (ID 指定)
履歴に対する操作一覧の取得	指定した ID の履歴に対して実行できる操作の一覧を取得します。	2.8.5 履歴に対する操作一覧の取得

表 2-6 プロパティ関連の API 一覧

API 名	機能	参照先
プロパティ定義一覧の取得	プロパティ定義の一覧を取得します。	2.9.1 プロパティ定義一覧の取得
プロパティ定義情報の取得	指定したプロパティ定義の情報を取得します。	2.9.2 プロパティ定義情報の取得

API名	機能	参照先
プロパティ定義に対する操作一覧の取得	指定したプロパティ定義に対して実行できる操作の一覧を取得します。	2.9.3 プロパティ定義に対する操作一覧の取得
プロパティ定義およびプロパティ値一覧の取得	プロパティ定義およびプロパティ値の情報の一覧を取得します。	2.9.4 プロパティ定義およびプロパティ値一覧の取得
プロパティ値一覧の取得	次に示すプロパティ値の一覧を取得します。 <ul style="list-style-type: none"> <li>サービス共有プロパティ</li> <li>特定のサービスに関連するプロパティ</li> <li>特定のスケジュールに関連するプロパティ</li> </ul>	2.9.5 プロパティ値一覧の取得
プロパティ値の一括更新	次に示すプロパティ値を一括更新します。 <ul style="list-style-type: none"> <li>特定のタスクに関連するプロパティ値</li> <li>特定のサービスに関連するプロパティ値</li> <li>サービス共有プロパティの値</li> <li>複数のサービスのプロパティ値</li> </ul>	2.9.6 プロパティ値の一括更新
プロパティ値の取得	指定したプロパティ値の情報を取得します。	2.9.7 プロパティ値の取得
プロパティ値の更新	指定した ID のプロパティ値を更新します。	2.9.8 プロパティ値の更新
プロパティ値に対する操作一覧の取得	指定したプロパティ値に対する操作の一覧を取得します。	2.9.9 プロパティ値に対する操作一覧の取得
プロパティグループ一覧の取得	サービスが保持するプロパティが所属しているプロパティグループの一覧を取得します。	2.9.10 プロパティグループ一覧の取得

表 2-7 サービスグループ関連の API 一覧

API名	機能	参照先
サービスグループ一覧の取得	サービスグループ一覧を取得します。	2.10.1 サービスグループ一覧の取得
サービスグループ情報の取得	指定したサービスグループの情報を取得します。	2.10.2 サービスグループ情報の取得
サービスグループに対する操作一覧の取得	指定したサービスグループに対して実行できる操作の一覧を取得します。	2.10.3 サービスグループに対する操作一覧の取得

表 2-8 タグ関連の API 一覧

API名	機能	参照先
タググループ一覧の取得	タググループの一覧を取得します。また、タググループに所属するタグ一覧を取得します。	2.11.1 タググループ一覧の取得
タグ一覧の取得	指定したリソースに設定されているタグの一覧を取得します。	2.11.2 タグ一覧の取得

表 2-9 情報管理の API 一覧

API 名	機能	参照先
ユーザー情報の取得	API を実行したユーザーの情報を取得します。	2.12.1 ユーザー情報の取得
バージョン情報の取得	JP1/AO と API のバージョンを取得します。	2.12.2 バージョン情報の取得

## 2.2 API 共通の仕様

---

各 API に共通する仕様を次に示します。なお、JP1/AO が提供している API は、REST (Representational State Transfer) アーキテクチャスタイルに従います。

ここでは JP1/AO に特有の仕様を記載しています。特に断り書きのない場合は HTTP1.1 に準拠します。

なお、この節での「API」は、「JP1/AO が提供する API」を指します。また、API を使用するポータルなどのユーザープログラムを総称して「API クライアント」と表記します。

### 2.2.1 通信方式

API が使用する通信プロトコルおよびポート番号について、次に示します。

- 通信プロトコル

API では、http プロトコルおよび https プロトコルをサポートしています。API は、JP1/AO が Web ブラウザーとの通信で使用するプロトコルを使用します。バージョンは、1.1 をサポートしています。通信プロトコルの詳細な仕様については、次の規定を参照してください。

- http プロトコルの場合  
RFC2616
- https プロトコルの場合  
RFC2818
- ポート番号

ポート番号のデフォルトの設定は、通信プロトコルおよび JP1/AO サーバの OS によって異なります。

- 通信プロトコルが http の場合  
22015
- 通信プロトコルが https の場合  
22016

なお、ポート番号を変更したい場合は、マニュアル「*JP1/AO 構築ガイド*」 - 「ポート番号を変更する手順」のトピックを参照してください。

### 2.2.2 セキュリティと認証

API のリクエストを発行してレスポンスを取得するには、ユーザー認証を受ける必要があります。JP1/AO の API は、Basic 認証 (Basic Access Authentication) または HSSO トークンによる認証を利用します。HSSO トークンは、シングルサインオンに必要なトークンです。HSSO トークンは、発行されて 1,000 秒でタイムアウトします。

ユーザー認証を受けるためには、使用する認証情報をリクエストヘッダーで指定してください。認証情報を指定する場合のリクエストヘッダーの指定例を次に示します。

## 指定例

Basic 認証の場合

```
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
```

HSSO トークンによる認証の場合

```
Authorization: HSSO 32bd25936120d68dceabcb49493079f8ef82a4_V0300
```

JP1/AO サーバは、権限のないリクエストが発行された場合、レスポンスとして、ステータスコード「401」を返却するとともに、ユーザー認証をリクエストします。

### ヒント

Basic 認証または HSSO トークンによる認証で JP1/AO サーバに接続した場合、レスポンスヘッダーとして「WWW-Authenticate: HSSO *hssso token*」が返却されます。再度、同じセッションで JP1/AO に接続して API を発行したい場合は、リクエストヘッダーを次のとおりに指定してください。

```
Authorization:HSSO hssso token
```

## 2.2.3 入出力形式

API のリクエストおよびレスポンスのデータ形式として、JSON 形式または XML 形式を利用できます。データ形式は、リクエストヘッダーで指定してください。指定を省略すると、JSON 形式が設定されます。入出力形式の文字コードは、UTF-8 を使用します。

入出力形式を XML 形式に指定する場合のリクエストヘッダーの指定例を次に示します。

## 指定例

```
Accept:application/xml  
Content-Type:application/xml
```

## 2.2.4 ネームスペース

API のリクエストおよびレスポンスのデータ形式が XML 形式の場合は、次のネームスペースを使用します。

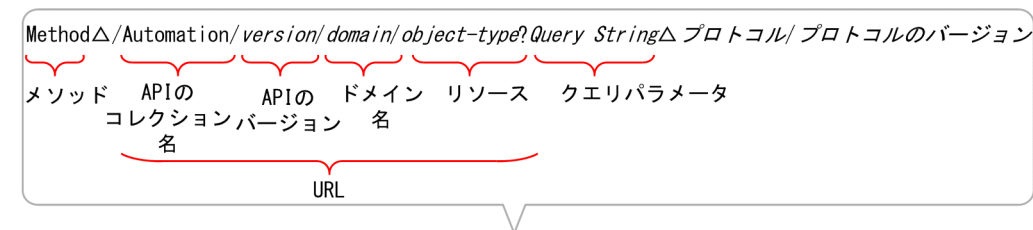
- <http://www.hitachi.com/products/it/software/xml/restfw/common/APIのバージョン>

- <http://www.hitachi.com/products/it/software/xml/automation/APIのバージョン>

## 2.2.5 リクエスト形式

API で JP1/AO が提供する機能を利用するためのリクエスト形式を次に示します。

図 2-1 リクエスト形式（ドメインが objects の場合の例）



Method△/Automation/version/domain/object-type?Query String△ プロトコル/ プロトコルのバージョン

Host: ホスト名またはIPアドレス: ポート番号  
 Accept: データ形式  
 Accept-Language: 言語コード (ja, zh, enなど)  
 User-Agent: APIクライアントのソフトウェア情報

} リクエストヘッダー

(凡例)

△ : 半角スペース

リクエスト形式の構成内容について、次の表で説明します。

表 2-10 リクエスト形式の構成内容

項目	説明	参照先	
メソッド	リソースに対する操作を指定します。	2.2.7 サポートするメソッド	
URL	API のコレクション名	API のコレクション*の名称です。固定値として「Automation」を指定します。	—
	API のバージョン	使用する API のバージョンを指定します。	各 API の「API のバージョン」を参照してください。
	ドメイン名	実行する API の操作対象となるリソースのドメイン名を指定します。なお、このリクエスト形式に該当するのは、ドメインが「objects」の場合です。	2.2.8 API で管理できるドメイン名とリソース
	リソース	JP1/AO が提供する機能を API のリソースとして提供しています。実行したい処理に応じてリソースを指定します。	
クエリパラメータ	リクエストに検索条件を付加することで、レスポンスの出力内容をフィルタリングしたり、ソートしたりすることができます。	2.2.9 クエリパラメータ	
プロトコル	API が使用する通信プロトコルとして「HTTP」を指定します。通信プロトコルに https を使用する場合も、「HTTP」と指定してください。	2.2.1 通信方式	



項目	説明	参照先	
プロトコルのバージョン	API が使用する通信プロトコルのバージョンとして「1.1」を指定します。	2.2.1 通信方式	
リクエストヘッダー	Host	ホストの情報を指定します。	2.2.10 リクエストヘッダー
	Accept	レスポンスのデータ形式を指定します。	
	Accept-Language	レスポンスの言語コードを指定します。	
	User-Agent	API クライアントのソフトウェア情報を指定します。	

(凡例)

—：該当しません。

注※

コレクションとは、処理の対象となるデータを指します。

リクエスト形式の構成内容や詳細は、API ごとに異なるため、各 API の「リクエスト形式」を参照してください。

## 2.2.6 レスポンス形式

レスポンス形式を次に示します。

### 図 2-2 レスポンス形式

プロトコル/プロトコルのバージョン ステータスコード メッセージ } ステータスライン  
 Content-Type: データ形式 } レスポンスヘッダー  
 レスポンスボディ

リクエストに対するレスポンス形式の構成内容について、次の表で説明します。

表 2-11 レスポンス形式の構成内容

項目	説明	参照先	
ステータスライン	プロトコル	API が使用する通信プロトコルが表示されます。	—
	プロトコルのバージョン	API が使用する通信プロトコルのバージョンが表示されます。	—
	ステータスコード	リクエストの実行結果がステータスコードとして返却されます。	<ul style="list-style-type: none"> <li>各 API の「ステータスコード」を参照してください。</li> <li>API を実行する前にエラーになった場合のステータスコードについては、「2.2.17 ステータスコード」のトピックを参照してください。</li> </ul>
	メッセージ	ステータスコードの内容が表示されます。	

項目		説明	参照先
レスポンスヘッダー	Content-Type	リクエストヘッダーで指定された、レスポンスのデータ形式が返却されます。	2.2.13 レスポンスヘッダー
レスポンスボディ		リクエストヘッダーで指定されたデータ形式のスキーマが返却されます。	各 API の「レスポンスのスキーマ」を参照してください。

(凡例)

－：該当しません。

## 2.2.7 サポートするメソッド

API では、リソースに対する操作をメソッドとして定義します。

API の処理に応じて、メソッドを指定してください。指定するメソッドについては、各 API の「リクエスト形式」を参照してください。

API がサポートするメソッドを次の表に示します。

表 2-12 サポートするメソッド

メソッド	説明
GET	リソースの一覧や情報を取得します。
POST	リソースに応じて JP1/AO の処理を実行します。
DELETE	リソースの情報を削除します。
PUT	リソースの情報を更新します。

## 2.2.8 API で管理できるドメイン名とリソース

実行する API の操作対象となるリソースのドメイン名、および JP1/AO がサポートするリソースを指定します。

なお、XML 形式のリクエストおよびレスポンスのデータでは、ドメイン名が objects のリソースの名称は単数形になります。

API で管理できるドメイン名とリソースの一覧を次の表に示します。

表 2-13 API で管理できるドメイン名とリソース

ドメイン名	リソース	リソースの説明
objects	ServiceTemplates	サービステンプレート
	Services	JP1/AO に登録されているサービス
	Schedules	タスクに設定されているスケジュール
	Tasks	サービスを実行することで生成されたタスク
	FlowSteps	タスクに含まれるステップ
	TaskLogs	タスクログ
	PropertyDefinitions	サービステンプレートのプロパティまたはサービス共有プロパティの定義
	PropertyInformations	プロパティ定義およびプロパティ値の情報
	PropertyValues	サービス、スケジュール、タスクのプロパティ、サービス共有プロパティの値
	PropertyGroups	プロパティグループ
	ServiceGroups	サービスグループ
	TagGroups	タググループ
	Tags	タグ
user	UserInfo*	JP1/AO が認証したユーザーの情報
configuration	VersionInfo*	JP1/AO および API のバージョン情報

注※

このリソースは、リクエストには指定しません。レスポンスに含まれるリソースの名称です。リクエストの指定方法は、各 API のリクエスト形式を参照してください。

## 2.2.9 クエリパラメータ

クエリパラメータでは、リクエストに検索条件を付加することで、レスポンスの出力内容をフィルタリングしたり、ソートしたりすることができます。

JP1/AO がサポートするクエリパラメータを示します。

### すべての API で指定できるクエリパラメータ

次の表に示すクエリパラメータは、すべての API に指定できます。

表 2-14 すべての API で指定できるクエリパラメータ

パラメータ	説明	指定できる値	デフォルト値
alt	リクエストの Content-Type ヘッダーや Accept ヘッダーと同様に、入出力のデータ形式を指定できます。	xml または json	—

(凡例)

—：該当しません。

## 一部の API で使用できるクエリパラメータ

次の表に示すクエリパラメータは、一部の API\*で指定できます。

注※

- サービスプレートの取得
- サービス一覧の取得
- スケジュール一覧の取得
- タスク一覧の取得
- ステップ一覧の取得
- タスクログの取得
- 履歴一覧の取得
- プロパティ定義一覧の取得
- プロパティ値一覧の取得
- プロパティ定義およびプロパティ値一覧の取得
- プロパティグループ一覧の取得
- サービスグループ一覧の取得
- タググループ一覧の取得
- タグ一覧の取得

表 2-15 一部の API で指定できるクエリパラメータ

パラメータ	説明	指定できる値※1	デフォルト値
HQL::filter	指定した条件でフィルタリングします。	[2.2.11 HQL 規格の使用]を参照してください。	—
HQL::fields	レスポンスに含めるメンバーを絞り込むときに指定します。[,] 区切りで複数指定できます。	メンバー名	—

パラメータ	説明	指定できる値 <sup>※1</sup>	デフォルト値
HQL::sortBy	指定されたメンバー名でソートします。	メンバー名 [{ASC  DESC}] (、メンバー名 [{ASC  DESC}]) <ul style="list-style-type: none"> <li>ASC：昇順</li> <li>DESC：降順</li> </ul>	ASC
HQL::offset <sup>※2</sup>	情報を取得する先頭オブジェクトの位置を指定します。レスポンスに含めるオブジェクトの最大数は HQL::count で指定します。HQL::offset よりも page が優先されます。	0～2147483647	0
HQL::count <sup>※2</sup>	HQL::offset に指定した先頭オブジェクトの位置を起点に、レスポンスに含めるオブジェクトの最大数を指定します。HQL::count よりも pageSize が優先されます。なお、HQL::count と HQL::offset の合計が 2,147,483,647 を超える場合、HQL::offset に指定した値から 2,147,483,647 番目までのオブジェクトを取得します。	1～2147483647	100
page <sup>※2</sup>	リソースを、ページ単位で分割する場合に、指定したページの情報を取得します。pageSize を同時に指定する必要があります。HQL::offset よりも page が優先されます。	1～2147483647	—
pageSize <sup>※2</sup>	1 ページに表示するオブジェクトの最大数を指定します。HQL::count よりも pageSize が優先されます。	1～2147483647	—

(凡例)

—：該当しません。

注※1

URL として表現できない文字列を値に指定したい場合は、UTF-8 形式で URL エンコードしてください。

注※2

「タスクログの取得」、「ステップ一覧の取得」および「プロパティグループ一覧の取得」の API では指定できません。

pageSize には 1 ページに表示するオブジェクトの最大数を指定し、page には分割されたページのうちの何ページ目を表示するかを指定します。これは、例えば JP1/AO の画面にある「サービス一覧」の「ページ」や「1 ページの行数」と概念は同じです。page および pageSize を指定すると、Pagination オブジェクトに全リソース数とページ数が返却されます。この値によって次のページがあるかどうかを判断できます。

なお、page および pageSize は次の式のとおり、HQL::offset の値に変換されて利用されます。

$$\text{HQL::offset} = \text{pageSize} * (\text{page} - 1)$$

そのため、page および pageSize が HQL::offset の範囲を超えた場合、ステータスコードは 400 の Bad Request が返却されます。

## 関連項目

- 2.2.11 HQL 規格の使用

## 2.2.10 リクエストヘッダー

リクエストヘッダーには、レスポンスのデータ形式およびレスポンスの言語コードなどを指定します。

表 2-16 リクエストヘッダー

ヘッダー	説明	指定できる値	デフォルト値	指定の要否
Host	ホストの情報として次の項目を指定します。 <ul style="list-style-type: none"><li>ホスト名または IP アドレス：JP1/AO サーバのホスト名または IP アドレスです。</li><li>ポート番号：API が JP1/AO に接続する際に、利用するポート番号です。 ポート番号を指定する際は、「2.2.1 通信方式」を参照してください。</li></ul>	ユーザーの環境を確認して指定します。	—	必須
Accept	レスポンスデータに指定したいデータ形式を指定します。	<ul style="list-style-type: none"><li>application/json：JSON 形式</li><li>application/xml：XML 形式</li><li>multipart/form-data：multipart 形式※1</li><li>text/html：HTML 形式※2</li></ul>	application/json	必須
Accept-Language	レスポンスデータに指定したい言語の言語コードを指定します。	<ul style="list-style-type: none"><li>ja または ja-JP：日本語</li><li>zh または zh-CN：中国語</li><li>en または en-US：英語</li><li>その他の地域のロケール：英語</li></ul>	en	必須
Content-Type	リクエストボディのデータ形式を指定します。	<ul style="list-style-type: none"><li>application/json：JSON 形式</li><li>application/xml：XML 形式</li></ul>	application/json	任意

ヘッダー	説明	指定できる値	デフォルト値	指定の要否
Content-Type	リクエストボディのデータ形式を指定します。	<ul style="list-style-type: none"> <li>application/octet-stream : octet-stream 形式<sup>※3</sup></li> </ul>	application/json	任意
Authorization	認証情報を指定します。	Basic 認証の場合 <i>ユーザー情報</i> HSSO トークンによる認証の場合 <i>hssso token</i>	—	任意

(凡例)

— : 該当しません。

注※1

「サービステンプレートのインポート」の API だけで有効です。

注※2

「サービステンプレートのインポートに必要な HTML ファイルの取得」の API だけで有効です。

注※3

「サービステンプレートのエクスポート」の API だけで有効です。

## 2.2.11 HQL 規格の使用

HQL::filter に HQL (Hitachi Query Language) を指定して、対象となるデータをフィルタリングできます。コレクションとは、フィルタリングの対象となるデータを指します。

URL として表現できない記号や文字は、UTF-8 形式で URL エンコードしてください。

### 形式

リソースリクエストのコレクションを定義するには、次に示す式を使用してください。

```

expression ::= "(" expression ")" | binary-expression | expression junction expression
junction ::= ( "and" | "or" )

binary-expression ::= (compare-expression | tuple-expression)

compare-expression ::= name-expression compare-operation value-expression

name-expression ::= property-name | "[" property-name "]"
compare-operation ::= ( "eq" | "=" | "ne" | "<" | "!=" | "gt" | ">" | "lt" | "<" | "ge"
| ">=" | "le" | "<=" | "starts" | "ends" )
value-expression ::= ( string-expression | number-expression | boolean-expression )

```

```

string-expression ::= "" ([^'] | [']{2})* ""
number-expression ::= ( "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" )+
boolean-expression ::= "true" | "false" | "TRUE" | "FALSE"

tuple-expression ::= name-expression tuple-operation tuple-value-expression
tuple-value-expression ::= "[" value-expression ("," value-expression)* "]"
tuple-operation ::= ( "in" | "not in" )

```

「*property-name*」は、リソースに定義されているメンバー名です。

「*value-expression*」は、その式が文字列表現、数値表現、真理値表現のどれであることを表示します。また、これはリソースに定義されているメンバーの実際のデータ型とは異なります。

データ型と表現形式との関係を次の表に示します。

表 2-17 データ型と表現形式との関係

データ型	表現形式
integer/long	number-expression
enum	string-expression
string	string-expression
ISO8601String	string-expression
URLString	string-expression
boolean	boolean-expression

HQL でサポートする演算子を、優先される順に次の表に示します。

表 2-18 HQL でサポートする演算子

演算子	説明	指定できるデータ型	優先順位
eq	等しい。	number-expression, string-expression (string, enum), boolean-expression	1
ne	等しくない。	number-expression, string-expression (string, enum), boolean-expression	1
gt	より大きい。	number-expression, string-expression (string, enum)	1
lt	より小さい。	number-expression, string-expression (string, enum)	1
ge	以上。	number-expression, string-expression (string, enum)	1
le	以下。	number-expression, string-expression (string, enum)	1



演算子	説明	指定できるデータ型	優先順位
starts*	開始値。	string-expression (ISO8601String を除く string)	1
ends*	終了値。	string-expression (ISO8601String を除く string)	1
in	含まれる。	number-expression, string-expression (string, enum), boolean-expression	1
not in	含まれない。	number-expression, string-expression (string, enum), boolean-expression	1
and	どちらも true である。	compare-expression, tuple-expression	2
or	どちらかが true である。	compare-expression, tuple-expression	3

注※

大文字と小文字は区別されません。

## 使用例

指定した列のフィルタリングをする例を次に示します。URL として表現できない文字列を値に指定したい場合は、UTF-8 形式で URL エンコードしてください。

URL エンコード前

```
...?HQL::filter=instanceID in [1000,1001,1002] and status = 'running'
```

URL エンコード後

```
...?HQL::filter=instanceID%20in%20%5b1000%2c1001%2c1002%5d%20and%20status%20%3d%20%27running%27
```

## 2.2.12 ドメインオブジェクトの形式

ドメインとは、JP1/AO がサポートするリソースが格納されている場所を指します。ドメインオブジェクトとは、リソースを指します。ここでは、リソースが持つメンバーのデータ形式について説明します。

### サポートするデータ型

JSON 形式および XML 形式でサポートするデータ型を次の表に示します。

表 2-19 サポートするデータ型

型名	説明
boolean	true または false

型名	説明
integer	32bit 符号付き整数
long	64bit 符号付き整数
string*	テキストデータ

#### 注※

ISO8601String, URLString および enum は, string 型の拡張表現です。

## 日付と時間

ドメインオブジェクトでの日付と時間の指定方法を説明します。

日付と時間は, ISO8601 フォーマットを使用して指定してください。フォーマットでは, 「年 (yyyy)」以外のすべての情報を省略できます。日付または時間が省略された場合, 指定可能な最小値が自動的に追加されます。タイムゾーンが省略された場合, デフォルトでは, JP1/AO サーバに設定されたタイムゾーンが設定されます。

ただし, HQL::filter を使用して日付と時間を指定する場合, 日付と時間は省略できません。また, JSON 形式で時間情報を取得する場合, タイムゾーン情報の「:」が表記されていない時間 (例: 2014-12-09T18:50:30.500+0900) が出力されます。JSON 形式で取得した時間情報を, そのまま API の入力に指定する場合には, タイムゾーンに「:」を追加してください (例: 2014-12-09T18:50:30.500+09:00)。「:」を追加しない場合, エラーになります。

なお, リソースのメンバーのデータ型が「ISO8601String」である場合, 年月日, 時刻, タイムゾーンは, 「yyyy-mmddThh:mm:ss.mmmTZD」の形式でレスポンスボディに表示されます。

表 2-20 年月日, 時刻, タイムゾーンの形式

フォーマット	指定例	JP1/AO で処理する時間
yyyy-mm-ddThh:mm:ss.mmmTZD	2014-12-09T18:50:30.500+09:00	指定例と同じ
yyyy-mm-ddThh:mm:ss.mmm	2014-12-09T18:50:30.500.000	2014-12-09T18:50:30.500.000[ホストサーバのタイムゾーン]
yyyy-mm-ddThh:mm:ssTZD	2014-12-09T18:50:30+09:00	2014-12-09T18:50:30.000+09:00
yyyy-mm-ddThh:mmTZD	2014-12-09T18:50+09:00	2014-12-09T18:50:00.000+09:00
yyyy-mm-ddThhTZD	2014-12-09T18+09:00	2014-12-09T18:00:00.000+09:00
yyyy-mm-dd	2014-12-09	2014-12-09T00:00:00.000[ホストサーバのタイムゾーン]
yyyy-mm	2014-12	2014-12-01T00:00:00.000[ホストサーバのタイムゾーン]
yyyy	2014	2014-01-01T00:00:00.000[ホストサーバのタイムゾーン]

## 2.2.13 レスポンスヘッダー

JP1/AO で制御しているレスポンスヘッダーを次の表に示します。

表 2-21 レスポンスヘッダー

ヘッダー	説明
Cache-Control	GET メソッドを指定した API のレスポンス情報をキャッシュ制御します。
Content-Type	レスポンスデータのデータ形式です。
Content-disposition	レスポンスデータが添付ファイルを示すときに付与されます。
Language	レスポンスデータの言語コードです。
Location	リクエストとは別の URL 情報で、レスポンスデータを取得する必要がある場合に、リダイレクション用の URL 情報が表示されます。
WWW-Authenticate	認証された HSSO トークンを出力します。
Warning	API の処理は成功しましたがサーバの状態に問題がある場合に、情報を表示します。

### 関連項目

- [2.2.10 リクエストヘッダー](#)

## 2.2.14 リソースのメンバー

JP1/AO が提供する機能をリソースとして分類します。レスポンスボディで、リソースの情報をメンバーとして取得できます。返却されるリソースのメンバー名、データ型、説明および HQL::filter と HQL::sortBy に適応するかどうかを、機能ごとの表に示します。

年月日の指定方法について特に断り書きがない場合は、「[表 2-20 年月日、時刻、タイムゾーンの形式](#)」を参照して指定してください。

表 2-22 「サービスプレート一覧の取得」で取得できるメンバー（リソース (ServiceTemplate)）

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
keyName	string	サービスプレート ID	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
displayName	string	サービステンプレートの表示名	○
iconURL	URLstring	サービステンプレートに設定されているアイコン画像の URL	×
vendorID	string	ベンダー ID	○
version	string	サービステンプレートのバージョン	○
vendorName	string	ベンダー名	○
tags	string	サービステンプレートに付与されたタグの一覧	×
createTime	ISO8601String	サービステンプレートが作成された年月日, 時刻, タイムゾーン	○
modifyTime	ISO8601String	サービステンプレートの更新年月日, 時刻, タイムゾーン	○
description	string	サービステンプレートの説明	○
releaseState	enum	サービステンプレートのリリース状態 <ul style="list-style-type: none"> <li>• debug : デバッグ</li> <li>• release : リリース</li> </ul>	○
latest	boolean	サービステンプレートが最新バージョンかどうか <ul style="list-style-type: none"> <li>• true : 最新バージョン</li> <li>• false : 最新バージョンではない</li> </ul>	○
imageUrl	URLString	サービス概要の画像の URL	×
supportedScheduleType	enum csv	サービステンプレートに適用できるスケジュールの種別 <ul style="list-style-type: none"> <li>• immediate : 即時実行</li> <li>• schedule : 指定日時実行</li> <li>• recurrence : 定期実行</li> </ul>	○
needVUP	boolean	古いバージョンのサービステンプレートを使っているサービスがあるかどうか <ul style="list-style-type: none"> <li>• true : ある</li> <li>• false : ない</li> </ul>	○
componentOutdated	boolean	サービステンプレートに古いバージョンのコンポーネントが含まれているかどうか <ul style="list-style-type: none"> <li>• true : 含まれている</li> <li>• false : 含まれていない</li> </ul>	○
usedServices	integer	サービステンプレートを使用しているサービスの数	×
usedTemplates	integer	サービス部品として使用しているサービステンプレートの数	×

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
disableFeatures	string	サービステンプレートに対する無効な操作	○
supportedActionType*	string	タスクに対して実行できる操作 <ul style="list-style-type: none"> <li>forciblyStop：強制停止</li> <li>retry：リトライ</li> </ul>	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

注※

supportedActionType が存在しない場合は、すべての操作が許可されます。

表 2-23 「サービス一覧の取得」で取得できるメンバー (リソース (Services))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	サービス名	○
description	string	サービスの説明	○
tags	string	タグの情報がタグ単位で「,」区切りで表示される	×
serviceTemplateName	string	サービスの元になるサービステンプレートの名称	○
createTime	ISO8601String	サービスの作成年月日, 時刻, タイムゾーン	○
modifyTime	ISO8601String	サービス情報の更新年月日, 時刻, タイムゾーン	○
serviceState	enum	サービスの種類 <ul style="list-style-type: none"> <li>debug：デバッグ</li> <li>test：テスト</li> </ul>	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
serviceState	enum	<ul style="list-style-type: none"> <li>• release：リリース</li> <li>• maintenance：メンテナンス</li> <li>• disabled：無効</li> </ul>	○
serviceGroupName	string	サービスが属するサービスグループの名称	○*
iconURL	URLString	サービステンプレートに設定されているアイコン画像の URL	×
vendorName	string	サービスの基になるサービステンプレートのベンダー名	○
version	string	サービスの基になるサービステンプレートのバージョン	○
lastSubmitTime	ISO8601String	ユーザーが最後にサービスを実行した年月日、時刻、タイムゾーン	○
favorite	boolean	サービスをお気に入りに登録しているかどうか <ul style="list-style-type: none"> <li>• true：登録している</li> <li>• false：登録していない</li> </ul>	○
failedCount	integer	サービスから生成されたタスクが失敗した回数	○
completedCount	integer	サービスから生成されたタスクが正常終了した回数	○
lastFailedTime	ISO8601String	サービスから生成されたタスクが最後に失敗した年月日、時刻、タイムゾーン	○
resetTime	ISO8601String	サービスのカウンターがリセットされた年月日、時刻、タイムゾーン	○
executedCount	integer	サービスから生成されたタスクの実行回数	○
latest	boolean	サービスが利用しているサービステンプレートが最新バージョンかどうか <ul style="list-style-type: none"> <li>• true：最新バージョン</li> <li>• false：最新バージョンではない</li> </ul>	○
imageUrl	URLString	サービス概要の画像の URL	×
supportedScheduleType	enum csv	サービスのスケジュール種別 <ul style="list-style-type: none"> <li>• immediate：即時実行</li> <li>• schedule：指定日時実行</li> <li>• recurrence：定期実行</li> </ul>	○
submitCount	integer	サービスを実行した回数	○
serviceTemplateID	long	サービスの基になるサービステンプレートの ID	○
serviceGroupID	long	サービスグループの ID	○
supportedActionType	string	タスクに対して実行できる操作 <ul style="list-style-type: none"> <li>• forciblyStop：強制停止</li> </ul>	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
supportedActionType	string	• retry：リトライ	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

注※

「All Resources」を指定した場合は、「DefaultServiceGroup」を指定したとみなされます。

表 2-24 「スケジュール一覧の取得」で取得できるメンバー (リソース (Schedules))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	サービスの実行時に指定したタスク名	○
submitter	string	実行ユーザー名	○
status	enum	定期実行のスケジュールの状態 <ul style="list-style-type: none"> <li>complete：定期実行が完了したスケジュール。実行されない状態である。</li> <li>running：定期実行中のスケジュール。定期的に行われる状態である。</li> </ul>	○
scheduleType	enum	スケジュールの種別 <ul style="list-style-type: none"> <li>immediate：即時実行</li> <li>schedule：指定日時実行</li> <li>recurrence：定期実行</li> </ul>	○
createTime	ISO8601String	サービスを実行して、スケジュールが作成された年月日、時刻、タイムゾーン	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
modifyTime	ISO8601String	タスク情報の更新年月日, 時刻, タイムゾーン	○
description	string	タスクの説明	○
scheduledStartTime	ISO8601String	指定日時実行タスクの開始予定年月日, 時刻, タイムゾーン	○*
recurrenceInterval	enum	定期実行周期 <ul style="list-style-type: none"> <li>• daily: 毎日</li> <li>• weekly: 毎週</li> <li>• monthly: 毎月</li> </ul>	○
recurrenceMinutes	integer	定期実行周期が毎日の場合に, サービスを実行する間隔 (分) <ul style="list-style-type: none"> <li>• 60</li> <li>• 120</li> <li>• 180</li> <li>• 240</li> <li>• 360</li> <li>• 480</li> <li>• 720</li> <li>• 1440</li> </ul>	×
recurrenceDayOfWeek	string	定期実行周期が毎週の場合に, サービスを実行する曜日 (1:日曜~7:土曜)	×
recurrenceDayOfMonth	string	定期実行周期が毎月の場合に, サービスを実行する日 (1~31 日)	×
recurrenceLastDayOfMonth	boolean	月の最終日に実行するかどうか <ul style="list-style-type: none"> <li>• true: 実行する</li> <li>• false: 実行しない</li> </ul>	○
recurrenceStartDate	string	定期実行タスクの実行開始日 (yyyy-mm-dd)	○
recurrenceTime	string	定期実行タスクの実行時刻 (hh:mm:ss)	○
serviceState	enum	サービスの種類 <ul style="list-style-type: none"> <li>• debug: デバッグ</li> <li>• test: テスト</li> <li>• release: リリース</li> <li>• maintenance: メンテナンス</li> </ul>	○
serviceID	long	スケジュールの生成元となるサービスの ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×



メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

注※

HQL::sortBy は適応しません。

表 2-25 「タスク一覧の取得」で取得できるメンバー (リソース (Tasks))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	タスク名	○
status	enum	タスクの状態 <ul style="list-style-type: none"> <li>failed：失敗</li> <li>completed：正常終了</li> <li>canceled：キャンセル</li> <li>inProgressTerminating：停止中</li> <li>inProgressWithError：異常検出</li> <li>waitingForInput：応答待ち中</li> <li>inProgress：実行中</li> <li>suspended：保留中</li> <li>waiting：待機中</li> <li>longRunning：長期実行中</li> </ul>	○
startTime	ISO8601String	タスクの開始年月日, 時刻, タイムゾーン	○
completionTime	ISO8601String	タスクの終了年月日, 時刻, タイムゾーン	○
scheduledStartTime	ISO8601String	指定日時実行タスクの開始予定年月日, 時刻, タイムゾーン	○
submitter	string	実行ユーザー名	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
submitTime	ISO8601String	サービス実行によってタスクが生成された年月日、時刻、タイムゾーン	○
modifyTime	ISO8601String	タスク情報の更新年月日、時刻、タイムゾーン	○
serviceState	enum	タスクの種類 <ul style="list-style-type: none"> <li>• debug：デバッグ</li> <li>• test：テスト</li> <li>• release：リリース</li> <li>• maintenance：メンテナンス</li> <li>• buildDebug：デバッガから実行</li> </ul>	○
scheduleType	enum	スケジュールの種類別 <ul style="list-style-type: none"> <li>• immediate：即時実行</li> <li>• schedule：指定日時実行</li> <li>• recurrence：定期実行</li> </ul>	○
description	string	タスクの説明	○
serviceName	string	タスクの生成元のサービス名	○
tags	string	タスクに付与されたタグの一覧	○
recurrenceInterval	enum	定期実行タスクの実行間隔 <ul style="list-style-type: none"> <li>• daily：毎日</li> <li>• weekly：毎週</li> <li>• monthly：毎月</li> </ul>	○
recurrenceTime	string	定期実行タスクの実行時刻 (hh:mm:ss)	○
recurrenceStartDate	ISO8601String	定期実行タスクの実行開始年月日 (yyyy-mm-dd)	○
serviceGroupName	string	タスクの生成元のサービスが所属するサービスグループの名称	○
todo	boolean	タスクに TODO が設定されているかどうか <ul style="list-style-type: none"> <li>• true：設定されている</li> <li>• false：設定されていない</li> </ul>	○
notes	string	タスクに記述したノート	○
stepStartTime	long	タスクに含まれるステップが最初に実行された年月日、時刻、タイムゾーン	○
serviceTemplateID	long	タスクが基づくサービステンプレートの ID	○
scheduleID	long	タスクが基づくスケジュールの ID	○
serviceGroupID	long	タスクの生成元のサービスが所属するサービスグループの ID	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
serviceID	long	タスクの生成元となるサービスの ID	○
supportedActionType	string	タスクに対して実行できる操作 <ul style="list-style-type: none"> <li>forciblyStop：強制停止</li> <li>retry：リトライ</li> </ul>	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-26 「応答待ち中のタスクに応答」で取得できるメンバー (リソース (ResponseInput))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
instanceID	string	インスタンス ID	×
dialogText	string	[応答入力] 画面に表示される文字列	×
labelButton0	string	選択肢 0	×
labelButton1	string	選択肢 1	×
labelButton2	string	選択肢 2	×
labelButton3	string	選択肢 3	×
labelButton4	string	選択肢 4	×
labelButton5	string	選択肢 5	×
labelButton6	string	選択肢 6	×
labelButton7	string	選択肢 7	×
labelButton8	string	選択肢 8	×
labelButton9	string	選択肢 9	×

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
screenURL	string	[応答入力] 画面を表示する URL	×
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

×：適応しません。

表 2-27 「ステップ一覧の取得」で取得できるメンバー (リソース (FlowSteps))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	string	インスタンス ID	×
name	string	ステップ名	×
startTime	string	ステップの開始年月日と時刻 (yyyy-MM-dd hh:mm:ss)	×
completionTime	string	ステップの終了年月日と時刻 (yyyy-MM-dd hh:mm:ss)	×
jobStatus	enum	ステップの状態 <ul style="list-style-type: none"> <li>• noplan：未計画</li> <li>• warning：警告</li> <li>• waiting：待機中</li> <li>• holding：保留中</li> <li>• break：中断中</li> <li>• break_after：中断中 (実行後)</li> <li>• running：実行中</li> <li>• waiting_for_response：応答待ち中</li> <li>• abnormal_continue：異常検出</li> <li>• complete：正常終了</li> <li>• error：失敗</li> </ul>	×

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
jobStatus	enum	<ul style="list-style-type: none"> <li>• abnormal：警告終了</li> <li>• un_exec：未実行終了</li> <li>• bypass：分岐未実行</li> <li>• terminate：停止</li> <li>• waiting_for_foreach：繰り返し実行待ち</li> </ul>	×
comment	string	ステップのコメント	×
stepStatus	enum	ステップの状態 (JP1/AO) <ul style="list-style-type: none"> <li>• normal：正常</li> <li>• warning：警告</li> <li>• waiting：待機中</li> <li>• holding：保留中</li> <li>• break：中断中</li> <li>• break_after：中断中（実行後）</li> <li>• running：実行中</li> <li>• waiting_for_response：応答待ち中</li> <li>• abnormal_continue：異常検出</li> <li>• complete：正常終了</li> <li>• error：失敗</li> <li>• abnormal：警告終了</li> <li>• un_exec：未実行終了</li> <li>• bypass：分岐未実行</li> <li>• terminate：停止</li> <li>• waiting_for_foreach：繰り返し実行待ち</li> </ul>	×
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

×：適応しません。

表 2-28 「タスクログの取得」で取得できるメンバー（リソース（Tasklogs））

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	×
text	string	タスクログの本文	×
totalSize	long	タスクログの総ファイルサイズ（単位：バイト）	×
readSize	long	取得したタスクログのサイズ（単位：バイト）	×
lineCount	long	取得したタスクログの行数	×
offset	long	取得時に指定したオフセット（単位：バイト）	×
reverse	boolean	タスクログをオフセットから逆方向に取得したか <ul style="list-style-type: none"> <li>• true：逆方向に取得した</li> <li>• false：正方向に取得した</li> </ul>	×
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page（ページ番号）	×
pageSize	integer	リクエストで指定した pageSize（ページに含まれるオブジェクトの最大数）	×
numPages	integer	page（ページ番号）の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数（0～n）	×

（凡例）

×：適応しません。

表 2-29 「履歴一覧の取得」で取得できるメンバー（リソース（TaskHistory））

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	タスク名	○
submitter	string	実行ユーザー名	○
serviceName	string	タスクの生成元のサービス名	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
tags	string	タグの情報 (CSV 形式)	×
scheduleType	enum	スケジュールの種別 <ul style="list-style-type: none"> <li>• immediate : 即時実行</li> <li>• schedule : 指定日時実行</li> <li>• recurrence : 定期実行</li> </ul>	○
scheduledStartTime	ISO8601String	指定日時実行タスクの開始年月日, 時刻, タイムゾーン	○
startTime	ISO8601String	タスクの開始年月日, 時刻, タイムゾーン	○
completionTime	ISO8601String	タスクの実行完了年月日, 時刻, タイムゾーン	○
stepStartTime	ISO8601String	長期実行中のタスクの開始年月日, 時刻, タイムゾーン	○
recurrenceInterval	enum	定期実行タスクの実行間隔 <ul style="list-style-type: none"> <li>• daily : 毎日</li> <li>• weekly : 毎週</li> <li>• monthly : 毎月</li> </ul>	○
recurrenceMinutes	integer	定期実行周期が毎日の場合に, サービスを実行する間隔 (分) <ul style="list-style-type: none"> <li>• 60</li> <li>• 120</li> <li>• 180</li> <li>• 240</li> <li>• 360</li> <li>• 480</li> <li>• 720</li> <li>• 1440</li> </ul>	×
recurrenceDayOfWeek	string	定期実行周期が毎週の場合に, サービスを実行する曜日 (1:日曜~7:土曜)	×
recurrenceDayOfMonth	string	定期実行周期が毎月の場合に, サービスを実行する日 (1~31 日)	×
recurrenceLastDayOfMonth	boolean	月の最終日に実行するかどうか <ul style="list-style-type: none"> <li>• true : 実行する</li> <li>• false : 実行しない</li> </ul>	○
recurrenceTime	string	定期実行タスクの実行時刻 (hh:mm:ss)	○
archiveTime	ISO8601String	タスクがアーカイブされた年月日, 時刻, タイムゾーン	○
taskID	long	タスク ID	○
submitTime	ISO8601String	タスクが実行された年月日, 時刻, タイムゾーン	○
recurrenceStartDate	ISO8601String	定期実行タスクの実行開始日 (yyyy-mm-dd)	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
status	enum	タスクの状態 <ul style="list-style-type: none"> <li>failed：失敗</li> <li>completed：正常終了</li> <li>canceled：キャンセル</li> <li>inProgressTerminating：停止中</li> <li>inProgressWithError：異常検出</li> <li>waitingForInput：応答待ち中</li> <li>inProgress：実行中</li> <li>suspended：保留中</li> <li>waiting：待機中</li> <li>longRunning：長期実行中</li> </ul>	○
description	string	タスクの説明	○
serviceState	enum	サービスのリリース状態 <ul style="list-style-type: none"> <li>debug：デバッグ</li> <li>test：テスト</li> <li>release：リリース</li> <li>maintenance：メンテナンス</li> </ul>	○
todo	boolean	タスクに TODO が設定されているかどうか <ul style="list-style-type: none"> <li>true：設定されている</li> <li>false：設定されていない</li> </ul>	○
notes	string	タスクに記述したノート	○
serviceGroupName	string	履歴の生成元のサービスが所属するサービスグループ名	○
serviceGroupID	long	履歴の生成元のサービスが所属するサービスグループ ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。



表 2-30 「プロパティ定義一覧の取得」で取得できるメンバー（リソース（PropertyDefinitions））

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
keyName	string	プロパティキー名	○
displayName	string	プロパティの表示名	×
defaultValue	string	プロパティのデフォルト値	○
type	enum	プロパティのデータ型 <ul style="list-style-type: none"> <li>• boolean</li> <li>• integer</li> <li>• string</li> <li>• double</li> <li>• timestamp(date)</li> <li>• password</li> <li>• list</li> <li>• file</li> </ul>	○
visibility	enum	プロパティの可視性 <ul style="list-style-type: none"> <li>• config</li> <li>• exec</li> </ul>	○
scope	enum	プロパティの有効範囲 <ul style="list-style-type: none"> <li>• share：サービス共有プロパティ</li> <li>• local：そのサービスだけに有効なプロパティ</li> </ul>	○
description	string	プロパティの説明	×
mode	enum	プロパティの入出力種別 <ul style="list-style-type: none"> <li>• in：入力プロパティ</li> <li>• out：出力プロパティ</li> </ul>	○
required	boolean	実行時にプロパティが指定必須かどうか <ul style="list-style-type: none"> <li>• true：指定必須</li> <li>• false：省略できる</li> </ul>	○
maxLength	integer	プロパティに入力できる文字列の最大長	○
minLength	integer	プロパティに入力できる文字列の最小長	○
minValue	string	プロパティに入力できる数値の最小値	○
maxValue	string	プロパティに入力できる数値の最大値	○
pattern	string	string または password のプロパティに指定できる文字列の正規表現パターン	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
valueList	string	プロパティのデータ型が list の場合に、プロパティ値の候補を「,」区切りで表現したもの	○
propertyGroupName	string	プロパティグループ名	○
validationScript	string	プロパティのバリデーション処理を JavaScript で記述したもの	○
readOnly	boolean	プロパティ値の変更を抑制するかどうか <ul style="list-style-type: none"> <li>• true：抑制する</li> <li>• false：抑制しない</li> </ul>	○
hidden	boolean	プロパティの画面表示を抑制するかどうか <ul style="list-style-type: none"> <li>• true：抑制する</li> <li>• false：抑制しない</li> </ul>	○
reference	boolean	プロパティ値が、ほかのプロパティ値を参照しているかどうか <ul style="list-style-type: none"> <li>• true：参照している</li> <li>• false：参照していない</li> </ul>	○
serviceTemplateID	long	サービステンプレートの ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-31 「プロパティ定義およびプロパティ値一覧の取得」で取得できるメンバー (リソース (PropertyInformation))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
keyName	string	プロパティのキー名	○
displayName	string	プロパティの表示名	×
defaultValue	string	プロパティのデフォルト値	○
value	string	プロパティの現在の値	○
type	enum	プロパティのデータ型 <ul style="list-style-type: none"> <li>• boolean</li> <li>• integer</li> <li>• string</li> <li>• double</li> <li>• timestamp</li> <li>• password</li> <li>• list</li> <li>• file</li> </ul>	○
visibility	enum	プロパティの可視性 <ul style="list-style-type: none"> <li>• config：[サービス設定] 画面の入力項目として表示される</li> <li>• exec：[サービス設定] 画面および [サービス実行] 画面の入力項目として表示される</li> </ul>	○
scope	enum	プロパティの有効範囲 <ul style="list-style-type: none"> <li>• share：サービス共有プロパティ</li> <li>• local：そのサービスだけに有効なプロパティ</li> </ul>	○
description	string	プロパティの説明	×
mode	enum	プロパティの入出力種別 <ul style="list-style-type: none"> <li>• in：入力プロパティ</li> <li>• out：出力プロパティ</li> </ul>	○
required	boolean	実行時にプロパティが指定必須かどうか <ul style="list-style-type: none"> <li>• true：指定必須</li> <li>• false：省略できる</li> </ul>	○
maxLength	integer	プロパティに入力できる文字列の最大長	○
minLength	integer	プロパティに入力できる文字列の最小長	○
minValue	string	プロパティに入力できる数値の最小値	○
maxValue	string	プロパティに入力できる数値の最大値	○
pattern	string	string または password のプロパティに指定できる文字列の正規表現パターン	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
valueList	string	プロパティのデータ型が list の場合に、プロパティ値の候補を「,」区切りで表現したもの	○
propertyGroupName	string	プロパティグループ名	○
validationScript	string	プロパティのバリデーション処理を JavaScript で記述したもの	×
readOnly	boolean	プロパティ値の変更を抑制するかどうか <ul style="list-style-type: none"> <li>• true：抑制する</li> <li>• false：抑制しない</li> </ul>	○
hidden	boolean	プロパティの画面表示および変更を抑制するかどうか <ul style="list-style-type: none"> <li>• true：抑制する</li> <li>• false：抑制しない</li> </ul>	○
reference	boolean	プロパティ値が、ほかのプロパティ値を参照しているかどうか <ul style="list-style-type: none"> <li>• true：参照している</li> <li>• false：参照していない</li> </ul>	×
serviceTemplateID	long	リソースのサービステンプレート ID	○
serviceID	long	リソースのサービス ID	○
taskID	long	リソースのタスク ID	○
scheduleID	long	リソースのスケジュール ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

#### (凡例)

○：適応します。 ×：適応しません。

表 2-32 「プロパティ値一覧の取得」で取得できるメンバー（リソース (PropertyValues)）

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
type	enum	プロパティのデータ型 <ul style="list-style-type: none"> <li>• boolean</li> <li>• integer</li> <li>• string</li> <li>• double</li> <li>• timestamp(date)</li> <li>• password</li> <li>• list</li> <li>• file</li> </ul>	○
keyName	string	プロパティキー名	○
value	string	プロパティの値	○
serviceID	long	リソースのサービス ID	○
scheduleID	long	リソースのスケジュール ID	○
taskID	long	リソースのタスク ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-33 「プロパティグループ一覧の取得」で取得できるメンバー（リソース (PropertyGroup)）

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
keyName	string	プロパティグループの ID	○

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
displayName	string	プロパティグループの表示名	×
description	string	プロパティグループの説明	×
ordinal	integer	プロパティグループの表示順序	×
validationScript	string	プロパティグループ内のプロパティ間のバリデーション処理を JavaScript で記述したもの	×
display	enum	プロパティグループを表示するかどうか <ul style="list-style-type: none"> <li>• submit : [サービス実行] 画面に表示</li> <li>• config : [サービス編集] 画面に表示</li> <li>• taskDetail : [タスク詳細] 画面に表示</li> </ul>	×
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○ : 適応します。 × : 適応しません。

表 2-34 「サービスグループ一覧の取得」で取得できるメンバー (リソース (ServiceGroup))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
objectID	string	リソース [Resource Group] のインスタンス ID	○
name	string	サービスグループの名称	○
description	string	サービスグループの説明	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-35 「タググループ一覧の取得」で取得できるメンバー (リソース (TagGroup))

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	タググループの名称	○
tag	string	所属するタグのリスト (CSV 形式)	×
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-36 「タグ一覧の取得」で取得できるメンバー（リソース (Tag)）（detail クエリパラメータを指定しない場合）

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	タグの名称	○
tagGroupID	long	タグが所属するタググループの ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-37 「タグ一覧の取得」で取得できるメンバー（リソース (Tag)）（detail クエリパラメータを指定する場合）

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
data	Object	リソースの一覧	×
instanceID	long	インスタンス ID	○
name	string	タグの名称	○
usedTemplates	integer	タグが使われているリリースサービステンプレートおよび開発サービステンプレートの数	○
usedServices	integer	タグが使われているサービスの数	○
usedTasks	integer	タグが使われているタスクの数	○
usedHistories	integer	タグが使われている履歴の数	○
usedPlugins	integer	タグが使われているリリース部品の数	○



メンバー名	データ型	説明	HQL::filterまたはHQL::sortByの適応
usedDevelopPlugins	integer	タグが使われている開発部品の数	○
usedDevelopTemplates	integer	タグが使われている開発サービステンプレートの数	○
tagGroupID	long	タグが所属するタググループの ID	○
pagination	Object	リソースをページ単位で分割する場合の情報	×
page	integer	リクエストで指定した page (ページ番号)	×
pageSize	integer	リクエストで指定した pageSize (ページに含まれるオブジェクトの最大数)	×
numPages	integer	page (ページ番号) の総数	×
totalCount	integer	返却されるリソースの総数	×
count	integer	クエリパラメータの条件に合致したデータの個数 (0~n)	×

(凡例)

○：適応します。 ×：適応しません。

表 2-38 情報管理機能のリソース (UserInfo) のメンバー

メンバー名	データ型	説明	HQL::filterまたはHQL::sortByの適応
userName	string	ユーザー名	×
accessPermission	string	ユーザーに与えられたアクセス権限	×
fullName	string	ユーザーのフルネーム	×
description	string	ユーザーの説明	×
email	string	ユーザーのメールアドレス	×
resourceGroup	ResourceGroup	ユーザーに与えられた、リソース「Resource Group」別のアクセス権限	×

(凡例)

×：適応しません。

表 2-39 情報管理機能のリソース (ResourceGroup) のメンバー

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
instanceID	string	インスタンス ID	×
name	string	リソース「Resource Group」の名称	×
description	string	リソース「Resource Group」の説明	×
accessPermission	string[]	ユーザーに与えられた、リソース「Resource Group」別のアクセス権限	×

(凡例)

×：適応しません。

表 2-40 情報管理機能のリソース (VersionInfo) のメンバー

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
productName	string	製品名 (JP1/Automatic Operation)	×
productVersion	string	製品のバージョン	×
apiVersionI	string	API のバージョン	×

(凡例)

×：適応しません。

表 2-41 情報管理機能のリソース (Information) のメンバー

メンバー名	データ型	説明	HQL::filter または HQL::sortBy の適応
message	string	メッセージ	×
messageID	string	メッセージ ID	×

(凡例)

×：適応しません。

## 2.2.15 JP1/AO を操作する API で返却されるメンバー

JP1/AO が提供する API のうち、JP1/AO の操作を実行するものがあります。該当する API と、返却されるメンバーを次に示します。なお、リクエストについては、各 API のリクエスト形式を参照してください。

### JP1/AO を操作する API

- サービステンプレートのインポート
- サービステンプレートのエクスポート
- サービステンプレートを基にしたサービスの作成
- サービスの実行
- サービスのカウンターのリセット
- サービスの状態を「リリース」に変更
- サービスの状態を「メンテナンス」に変更
- サービスの状態を「無効」に変更
- サービスが使用するサービステンプレートのバージョン変更
- スケジュールのキャンセル
- スケジュールの一時停止
- スケジュールの再開
- タスクの実行停止
- タスクの強制停止
- タスクの再実行
- 応答待ち中のタスクに応答
- 失敗したステップからリトライ
- 失敗した次のステップからリトライ
- タスクのアーカイブ

表 2-42 JP1/AO を操作する API で返却されるメンバー

メンバー名	データ型	説明
instanceID	string	インスタンス ID を示します。
created	string	オブジェクトの生成日時を示します。ISO8601 形式です。
updated	string	非同期処理の場合に、このオブジェクトが更新された時刻を示します。同期処理の場合は「created」と同時刻を示します。ISO8601 形式です。
completed	string	非同期処理の場合に、処理が完了した時刻を示します。同期処理の場合は「created」と同時刻を示します。ISO8601 形式です。

メンバー名	データ型	説明
state	string	<ul style="list-style-type: none"> <li>• queued：操作が処理を開始していません。この状態では、停止処理だけを受け付けます。</li> <li>• running：操作が実行中です。この状態では、停止処理だけを受け付けます。</li> <li>• failed：操作が失敗しました。</li> <li>• success：操作が正常終了しました。</li> <li>• stopping：操作が停止の処理中です。</li> <li>• stopped：操作が完了する前に停止されました。</li> </ul>
affectedResource	string	操作の結果、作成または更新された API のリソースの URL を示します。

## 2.2.16 実行できる操作を取得する API で返却されるメンバー

JP1/AO が提供する API のうち、リソースに対して実行できる操作、およびその実行先 URL を取得するものがあります。該当する API と、返却されるメンバーを次に示します。なお、リクエストについては、各 API のリクエスト形式を参照してください。

### 実行できる操作を取得する API

- サービステンプレートに対する操作一覧の取得
- サービステンプレートのインポートに必要な HTML ファイルの取得
- サービステンプレートのエクスポートに必要な情報の取得
- サービステンプレートの詳細説明の URL を取得
- サービステンプレートを基にしたサービスの作成に必要な情報の取得
- サービスに対する操作一覧の取得
- サービス実行に必要な情報の取得
- サービスのカウンターのリセットに必要な情報の取得
- サービスの状態を「リリース」に変更する操作に必要な情報の取得
- サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得
- サービスの状態を「無効」に変更する操作に必要な情報の取得
- サービスの詳細説明の URL を取得
- サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得
- スケジュールに対する操作一覧の取得
- スケジュールのキャンセルに必要な情報の取得
- スケジュールの一時停止に必要な情報の取得
- スケジュールの再開に必要な情報の取得
- タスクの削除

- タスク操作一覧の取得
- タスク実行停止に必要な情報の取得
- タスクの強制停止に必要な情報の取得
- タスクの再実行に必要な情報の取得
- 応答待ち中のタスクに応答するために必要な情報の取得
- タスクのリトライに必要な情報の取得（失敗したステップからリトライ）
- タスクのリトライに必要な情報の取得（失敗した次のステップからリトライ）
- タスクのアーカイブに必要な情報の取得
- 履歴に対する操作一覧の取得
- プロパティ定義に対する操作一覧の取得
- プロパティ値に対する操作一覧の取得
- サービスグループに対する操作一覧の取得

表 2-43 実行できる操作を取得する API で返却されるメンバー

メンバー名	データ型	説明
name	string	操作名です。
href	string	操作の実行先 URL です。
method	string	メソッド名です。 <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> </ul>
parameters	Object	操作実行時に必要なパラメータです。

## 2.2.17 ステータスコード

API の実行時に返却されるステータスコードの一覧を示します。なお、返却されるステータスコードは、API ごとに異なるため、詳細は各 API を参照してください。

表 2-44 ステータスコード

ステータスコード	メッセージ	説明
200	OK	リクエストの処理に成功しました。
201	Created	リソースの作成に成功した場合、ステータス 200 の代わりにステータス 201 が返却されます。
400	Bad Request	リクエスト内容が不正です。

ステータスコード	メッセージ	説明
401	Unauthorized	認証に失敗しています。認証情報または許可情報が不正です。受理された認証方法は、レスポンスヘッダー「WWW-Authenticate」で通知します。受理された認証方法を、リクエストヘッダー「Authorization」に指定してください。または、サービスグループに対する権限が User Management 権限がありません。
403	Forbidden	リクエストの実行権限がありません。
404	Not found	リクエストされたリソースまたはリソースに対する操作がありません。または、クエリパラメータの指定が不正です。
405	Method not allowed	リクエストされたメソッドは、このリソースにはありません。
406	Not acceptable	指定されたレスポンスのフォーマット形式はサポートされていません。
409	Conflict	サーバ上にすでに存在しているデータと競合しているため、または、リクエストを受理できない状態のため、リクエストを完了できません。
412	Precondition failed	前提条件を満たしていないため、リクエストを受け付けられません。
415	Unsupported media type	指定されたリクエストのフォーマット形式はサポートされていません。
500	Server-side error	サーバ処理エラーが発生しました。

## 関連項目

- 2.2.18 エラー情報

## 2.2.18 エラー情報

API のリクエストがエラーになった場合について説明します。API のリクエストがエラーになった場合、レスポンス情報として、次の表に示すスキーマが返却されます。なお、次の表に示すスキーマ以外のエラー情報については、マニュアル [API/AO メッセージ] を参照してください。

エラー情報のスキーマを、次の表に示します。

表 2-45 エラー情報のスキーマ

メンバー名	データ型	説明
errorSource	string	エラーが発生した API です。
message	string	エラーのメッセージ内容です。
messageID	string	メッセージ ID です。リクエストの際に XML の記載を誤るなどの理由でエラーが発生した場合は、メッセージ ID が "generic error" になります。

メンバー名	データ型	説明
application	string	エラーが発生した API を保持するアプリケーション情報 (Automation) です。
messageData	string	エラーの詳細情報です。

## 出力例

エラー情報の messageID として、KNAE02102-E が出力される場合の例を次に示します。

```
{
  "errorSource" : "http://10.196.184.238:22015/Automation/v1/objects/Tasks/555",
  "message" : "The specified resource does not exist or you do not have access. After reviewing the content of the following, please re-run.¥n- The presence or absence of resources¥n- Access rights to the resource",
  "messageID" : "KNAE02102-E",
  "application" : "Automation"
}
```

## 2.3 API の記述形式

---

各 API で説明する項目を次に示します。ただし、API によっては説明のない項目もあります。

### 機能

API の機能について説明します。

### 実行権限

API を実行できる権限およびロールを示します。

### API のバージョン

API のバージョンを示します。

### リクエスト形式

API を使用する際のリクエスト形式について説明しています。

### ステータスコード

http プロトコルまたは https プロトコルを使用して API を実行したあとのステータスコードについて説明します。API を実行する前にエラーになった場合のステータスコードについては、「[2.2.17 ステータスコード](#)」のトピックを参照してください。

### レスポンスのスキーマ

API が正常終了したときに返却するレスポンスのスキーマ情報について説明します。

### 使用例

API を使用するためのリクエスト例およびレスポンス例について説明します。

ただし、ここでは、http プロトコルを使用している場合の例を記載します。https プロトコルを使用する場合は、「HTTP」を「HTTPS」に読み替えてください。



## 2.4 サービステンプレート関連の API

### 2.4.1 サービステンプレート一覧の取得

#### 機能

JP1/AO に登録されているサービステンプレートの一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceTemplates
```

API の実行ユーザーが権限を持つ、すべてのサービステンプレートを対象として一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-46 API 「サービステンプレート一覧の取得」で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
tags	値をすべて含んでいるか。[,] 区切りで複数指定できる。
q	次に示す検索対象のスキーマに対して、値を含んでいるか全文検索する。 <ul style="list-style-type: none"><li>• keyName</li><li>• displayName</li><li>• vendorID</li><li>• vendorName</li><li>• tags</li><li>• description</li></ul> 半角スペース区切りで複数指定した場合、値をすべて含んでいるか全文検索する。大文字・小文字を区別しない。
usingServiceTemplateID	指定した値を含むサービス部品。
vendorID	指定した値に等しい。大文字・小文字を区別しない。
keyName	

クエリパラメータ	フィルタ条件
version	指定した値に等しい。大文字・小文字を区別しない。

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービステンプレート取得の権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"サービステンプレート機能のリソース (ServiceTemplates) のメンバー" :
値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのサービステンプレートを対象として、サービステンプレート一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceTemplates HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
```

Date: Thu, 30 Jul 2015 00:34:32 GMT  
Server: Cosminexus HTTP Server  
Access-Control-Expose-Headers: WWW-Authenticate  
WWW-Authenticate: HSSO  
998ebb201be1cf76e7491a1380c4c54d5a59b7\_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=\_V0810  
Access-Control-Allow-Origin: \*  
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS  
Access-Control-Allow-Credentials: true  
Cache-Control: no-cache  
Transfer-Encoding: chunked  
Content-Type: application/json

レスポンスボディ:

```
{
  "data": [ {
    "instanceID": 560,
    "keyName": "remoteCommandExe",
    "displayName": "Execute Remote Command",
    "iconURL": "http://10.196.184.182:22015/Automation/icon/services/
com.hitachi.software.dna.cts/remoteCommandExe/01.20.00",
    "vendorID": "Hitachi,Ltd.",
    "version": "01.20.00",
    "vendorName": "Hitachi,Ltd.",
    "tags": "Execute Script,Linux,Windows",
    "createTime": "2015-07-29T15:27:02.000+09:00",
    "modifyTime": "2015-07-29T15:27:02.000+09:00",
    "description": "Executes a command on the remote execution target server.",
    "releaseState": "release",
    "latest": true,
    "supportedScheduleType": "immediate,schedule,recurrence",
    "needVUP": false,
    "componentOutdated": false,
    "usedServices": 0,
    "usedTemplates": 0,
    "supportedActionType": "forciblyStop,retry"
  }, {
    "instanceID": 1116,
    "keyName": "SP_GenericApplication",
    "displayName": "Allocate Volumes for Generic Application",
    "iconURL": "http://10.196.184.182:22015/Automation/icon/services/
com.hitachi.software.dna.cts/SP_GenericApplication/01.20.00",
    "vendorID": "Hitachi,Ltd.",
    "version": "01.20.00",
    "vendorName": "Hitachi, Ltd.",
    "tags": "Add New Storage",
    "createTime": "2015-07-29T16:48:25.000+09:00",
    "modifyTime": "2015-07-29T16:48:25.000+09:00",
    "description": "Intelligent allocation service that uses sets of volumes from the
associated infrastructure group to be consumed by server(s) running a generic application",
    "releaseState": "release",
    "latest": true,
    "imageURL": "http://10.196.184.182:22015/Automation/services/custom/000000000001116/
SP_GenericApplication_overview.png",
    "supportedScheduleType": "immediate,schedule",
    "needVUP": false,
    "componentOutdated": false,
    "usedServices": 0,
  }
]
```

```
    "usedTemplates" : 0,
    "supportedActionType" : "forciblyStop,retry"
  } ],
  "count" : 2
}
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.4.2 サービステンプレートの情報取得

### 機能

指定したサービステンプレートの情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceTemplates/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービステンプレート取得の権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```

{
  "instanceID" : instance-id,
  "keyName" : "key-name",
  "displayName" : "display-name",
  "iconURL" : "icon-URL",
  "vendorID" : "vendor-ID",
  "version" : "version",
  "vendorName" : "vendor-name",
  "tags" : "tag",
  "createTime" : "created-date-and-time",
  "modifyTime" : "updated-date-and-time",
  "description" : "description",
  "releaseState" : "release-state",
  "latest" : {true|false},
  "imageURL" : "imageURL",
  "supportedScheduleType" : "supported-schedule-type",
  "needVUP" : {true|false},
  "componentOutdated" : {true|false},
  "usedServices" : used-services,
  "usedTemplates" : used-Templates,
  "disableFeatures" : "disable-features",
  "supportedActionType" : "supported-action-type"
}

```

## 使用例

instanceID=1116 のサービステンプレートに関する情報を取得する場合の使用例を示します。

リクエストヘッダー:

```

GET /Automation/v1/objects/ServiceTemplates/1116 HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 00:36:51 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
a9a6768131e2eff3ecbd5e4457f49e82e0506c_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{

```

```
  "instanceID" : 1116,
  "keyName" : "SP_GenericApplication",
  "displayName" : "Allocate Volumes for Generic Application",
  "iconURL" : "http://10.196.184.182:22015/Automation/icon/services/
com.hitachi.software.dna.cts/SP_GenericApplication/01.20.00",
  "vendorID" : "com.hitachi.software.dna.cts",
  "version" : "01.20.00",
  "vendorName" : "Hitachi, Ltd.",
  "tags" : "Add New Storage",
  "createTime" : "2015-07-29T16:48:25.000+09:00",
  "modifyTime" : "2015-07-29T16:48:25.000+09:00",
  "description" : "Intelligent allocation service that uses sets of volumes from the
associated infrastructure group to be consumed by server(s) running a generic application",
  "releaseState" : "release",
  "latest" : true,
  "imageURL" : "http://10.196.184.182:22015/Automation/services/custom/000000000001116/
SP_GenericApplication_overview.png",
  "supportedScheduleType" : "immediate,schedule",
  "needVUP" : false,
  "componentOutdated" : false,
  "usedServices" : 0,
  "usedTemplates" : 0,
  "supportedActionType" : "forciblyStop,retry"
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.4.3 サービステンプレートの削除

### 機能

指定したサービステンプレートを削除します。

### 実行権限

Admin ロール, Develop ロール

### API のバージョン

v1

### リクエスト形式

```
DELETE http://host:port/Automation/version/objects/ServiceTemplates/id
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
204	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービステンプレート削除の権限がありません。
409	Conflict	指定したサービステンプレートから生成されたサービスがあります。または、指定したサービステンプレートを、サービス部品として使用しているサービステンプレートがあります。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## 使用例

instanceID=1116 のサービステンプレートを削除する場合の使用例を示します。

リクエストヘッダー:

```
DELETE /Automation/v1/objects/ServiceTemplates/1116 HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 204 No Content
Date: Thu, 30 Jul 2015 00:39:20 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
7cfe7ffcd3e5603af8b08e3d2abdfafc5da41e3_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.4.4 サービステンプレートに対する操作一覧の取得

### 機能

指定したサービステンプレートに対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceTemplates/id/actions
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {
    "name" : "delete",
    "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id",
    "method" : "DELETE",
    "parameters" : []
  }, {
    "name" : "export",
    "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/export/invoke",
    "method" : "POST",
    "parameters" : []
  }, {
```



```

    "name" : "detailhelp",
    "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/
detailhelp",
    "method" : "GET",
    "parameters" : []
  }, {
    "name" : "bind",
    "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/bind/
invoke",
    "method" : "POST",
    "parameters" : []
  } ],
  "count" : count
}

```

## 使用例

instanceID=1116 のサービステンプレートに対する操作一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```

GET /Automation/v1/objects/ServiceTemplates/1116/actions HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 00:39:20 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
7cfe7ffcd3e5603af8b08e3d2abdfafc5da41e3_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{
  "data" : [ {
    "name" : "delete",
    "href" : "http://10.196.184.182:22015/Automation/v1/objects/ServiceTemplates/1116",
    "method" : "DELETE",
    "parameters" : [ ]
  }, {
    "name" : "export",
    "href" : "http://10.196.184.182:22015/Automation/v1/objects/ServiceTemplates/1116/
actions/export/invoke",

```

```
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "detailhelp",
    "href" : "http://10.196.184.182:22015/Automation/v1/objects/ServiceTemplates/1116/
actions/detailhelp",
    "method" : "GET",
    "parameters" : [ ]
  }, {
    "name" : "bind",
    "href" : "http://10.196.184.182:22015/Automation/v1/objects/ServiceTemplates/1116/
actions/bind/invoke",
    "method" : "POST",
    "parameters" : [ ]
  } ],
  "count" : 4
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.4.5 サービステンプレートのインポートに必要な HTML ファイルの取得

### 機能

サービステンプレートのインポートに必要な HTML ファイルを取得します。なお、HTML ファイルには認証情報が付与されていません。実行時には、あらかじめ JP1/AO にログインしてセッションを確保してください。

### 実行権限

Admin ロール, Develop ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/services/ServiceTemplates/actions/import
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	インポートの権限がありません。
406	Not acceptable	Accept ヘッダーの指定が不正です。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
<html>
<body>
<form method="POST" action="http://host:port/Automation/version/services/ServiceTemplates/
actions/import/invoke"
  enctype="multipart/form-data">
  <input name="file" type="file"></input>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

## 使用例

サービステンプレートのインポートに必要な HTML ファイルを取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/services/ServiceTemplates/actions/import HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: text/html
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 00:40:59 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
ea15867727ce4f2cd07d5a48a3dedf919a34577_Vlo8Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
```

```
Transfer-Encoding: chunked
Content-Type: text/html
```

レスポンスボディ:

```
<html>
<body>
<form method="POST" action="http://10.196.184.182:22015/Automation/v1/services/
ServiceTemplates/actions/import/invoke" enctype="multipart/form-data">
  <input name="file" type="file"></input>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.4.6 サービステンプレートのインポート

### 機能

指定したサービステンプレートをインポートします。

### 実行権限

Admin ロール, Develop ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/services/ServiceTemplates/actions/import/invoke
```

リクエストボディには、サービステンプレート (.st または .zip) を指定してください。

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。

ステータスコード	メッセージ	説明
400	Bad Request	.st ファイルまたは.zip ファイル以外が指定されました。 または、指定された.st ファイルまたは.zip ファイルが、壊れているか誤っています。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	インポートの権限がありません。
412	Precondition failed	サーバが利用できません。
415	Unsupported media type	Content-Type ヘッダーの指定が不正です。
500	Server-side error	一時保存フォルダへの保存に失敗しました。または、サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

## 使用例

サービステンプレート (SP\_GenericApplication\_01.20.00.st) をインポートする場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/services/ServiceTemplates/actions/import/invoke HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Length: 2106265
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----5564f06622f7727e
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Date: Wed, 29 Jul 2015 07:48:21 GMT
```

```
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
9c9f012d1d34b9ede86d68728604c884b85e8_Vlo8Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "f4c5065a-ff42-45df-bca9-e2d79b4b5bb7",
  "created" : "2015-07-29T16:48:26.528+09:00",
  "updated" : "2015-07-29T16:48:26.528+09:00",
  "completed" : "2015-07-29T16:48:26.528+09:00",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/
ServiceTemplates/1116" ],
  "result" : [ {
    "message" : "サービステンプレートのインポートに成功しました (サービステンプレートファイ
ル名: SP_GenericApplication_01.20.00.st).",
    "messageID" : "KNAE03111-I"
  } ]
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.4.7 サービステンプレートのエクスポートに必要な情報の取得

### 機能

指定したサービステンプレートのエクスポートに必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceTemplates/id/actions/export
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービステンプレートの取得権限がありません。 または、サービステンプレートがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "export",
  "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/export/invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=1116 のサービステンプレートのエクスポートに必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceTemplates/1116/actions/export HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 00:42:05 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
c21cd879a4c62f90d8f7c5775ec1194e88a92b_Vlo8Y30JdDBUB3ljJSVpRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
```

```
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "export",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/ServiceTemplates/1116/actions/export/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.4.8 サービステンプレートのエクスポート

### 機能

指定したサービステンプレートをエクスポートします。

### 実行権限

Admin ロール, Develop ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/ServiceTemplates/id/actions/export/invoke
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。



ステータスコード	メッセージ	説明
404	Not found	権限が不正です。または、リソースがありません。
406	Not acceptable	Accept ヘッダーの指定が不正です。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## 使用例

instanceID=1116 のサービステンプレートをエクスポートする場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/ServiceTemplates/1116/actions/export/invoke HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/octet-stream
Content-Type: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 01:58:34 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
866ad68aa7c23e457456b5b08479fb62250fdf_Vl08Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Content-disposition: attachment;
filename="com.hitachi.software.dna.cts_SP_GenericApplication_01.20.00.st"
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/octet-stream
```

レスポンスボディ:

"com.hitachi.software.dna.cts\_SP\_GenericApplication\_01.20.00.st"のバイナリ形式。

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.4.9 サービステンプレートの詳細説明を表示する URL の取得

### 機能

指定したサービステンプレートの詳細説明を表示する URL を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceTemplates/id/actions/detailhelp
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービステンプレートの取得権限がありません。 または、サービステンプレートがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "export",
  "href" : "Link-to-the-detail-help",
  "method" : "POST",
  "parameters" : []
}
```

### 使用例

instanceID=1116 のサービステンプレートの詳細説明を表示する URL を取得する場合の使用例を示します。

#### リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceTemplates/1116/actions/detailhelp HTTP/1.1
Authorization: Basic c3lzdGVtOmlhbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

#### レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 02:04:35 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
11baaddb4ff5c120d1cca95c75fab1417d2c921_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

#### レスポンスボディ:

```
{
  "name" : "detailhelp",
  "href" : "http://10.196.184.182:22015/Automation/services/custom/00000000001116/
r_all_vol_details.html",
  "method" : "GET",
  "parameters" : [ ]
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)

---

## 2.4.10 サービステンプレートを基にしたサービスの作成に必要な情報の取得

### 機能

指定したサービステンプレートからサービスを作成する場合に必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceTemplates/id/actions/bind
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービステンプレートの取得権限がありません。 または、サービステンプレートがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "bind",
  "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/bind/
invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=560 のサービステンプレートを基にしたサービスの作成に必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceTemplates/560/actions/bind HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

#### レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 02:08:29 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
77efd47709df8b7f65468cb4778e804db1e6c_Vlo8Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

#### レスポンスボディ:

```
{
  "name" : "bind",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/ServiceTemplates/560/actions/
bind/invoke",
  "method" : "POST",
  "parameters" : [ {
    "name" : "Execute Remote Command",
    "description" : "Executes a command on the remote execution target server.",
    "tags" : "Execute Script,Linux,Windows",
    "serviceTemplateName" : "remoteCommandExe",
    "serviceState" : "test",
    "serviceGroupName" : "DefaultServiceGroup",
    "supportedScheduleType" : "immediate,schedule,recurrence",
    "serviceTemplateID" : 560
  }, {
    "type" : "string",
    "keyName" : "common.targetHost",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "type" : "string",
    "keyName" : "common.remoteCommand",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "type" : "string",
    "keyName" : "common.remoteCommandParameter",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  } ]
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.4.11 サービステンプレートを基にしたサービスの作成

### 機能

指定したサービステンプレートからサービスを作成します。サービスの作成時に、プロパティの値を指定できます。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/ServiceTemplates/id/actions/bind/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "bind",
  "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/bind/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

リクエストのスキーマで「parameters」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-47 「parameters」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
サービス	Services	1	作成するサービス
プロパティ値	PropertyValues	0	サービスの入力プロパティ

これらのオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
Services	name	1
	description	
	tags	
	supportedScheduleType	

リソース名称	メンバー名	数
Services	serviceState	1
	serviceGroupName	
PropertyValues	value	0~n

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	次のうちのどれかが原因です。 <ul style="list-style-type: none"> <li>引数が不正です。</li> <li>サービスグループに割り当てられた権限が不正です。</li> <li>すでに存在しているサービス名を指定しています。</li> <li>サービス数、タグ数が上限に達しています。</li> </ul>
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービスを作成する権限がありません。
404	Not found	サービステンプレートの取得権限がありません。または、サービステンプレートが存在しません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "bind",
  "href" : "http://host:port/Automation/version/objects/ServiceTemplates/id/actions/bind/
invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=560 のサービステンプレートを基にしてサービスを作成する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/ServiceTemplates/560/actions/bind/invoke HTTP/1.1
```

```
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 1001
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 02:30:37 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
fdef80b1cbd2d625cdbda39c16fda15f68a3d8c_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "55e8c5b7-b0ab-4016-ba62-f334b67c20c4",
  "created" : "2015-07-30T11:30:39.042+09:00",
  "updated" : "2015-07-30T11:30:39.042+09:00",
  "completed" : "2015-07-30T11:30:39.042+09:00",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Services/
2004" ],
  "result" : [ ]
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-



## 2.5 サービス関連の API

サービスリソースを管理するための操作を説明します。

### 2.5.1 サービス一覧の取得

#### 機能

JP1/AO に登録されているサービスの一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/Services
```

API の実行ユーザーが権限を持つ、すべてのサービスを対象として一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-48 API 「サービス一覧の取得」 で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
serviceGroupID	指定した値に等しい。
serviceTemplateID	
favorite	
propertyKey	指定した値を含む「PropertyValues」リソースの keyName
tags	値をすべて含んでいるか。[,] 区切りで複数指定できる。
q	次に示す検索対象のスキーマに対して、値を含んでいるか全文検索する。 <ul style="list-style-type: none"><li>• name</li><li>• description</li><li>• tags</li><li>• serviceTemplateName</li><li>• vendorName</li></ul>

クエリパラメータ	フィルタ条件
q	半角スペース区切りで複数指定した場合、値をすべて含んでいるか全文検索する。大文字・小文字を区別しない。

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"サービス機能のリソース (Services) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのサービスを対象として、サービス一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Host: 10.196.184.182:22015
User-Agent: curl/7.36.0
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 02:30:37 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
```

```
WWW-Authenticate: HSSO
fdef80b1cbd2d625cdbda39c16fda15f68a3d8c_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 5137,
    "name" : "Execute Remote Command",
    "description" : "Executes a command on the remote execution target server.",
    "tags" : "Windows, Linux, Execute Script",
    "serviceTemplateName" : "Execute Remote Command",
    "createTime" : "2015-08-07T14:44:07.000+09:00",
    "modifyTime" : "2015-08-07T14:44:07.000+09:00",
    "serviceState" : "test",
    "serviceGroupName" : "DefaultServiceGroup",
    "iconURL" : "http://10.196.184.182:22015/Automation/icon/services/
com.hitachi.software.dna.cts/remoteCommandExe/01.20.00",
    "vendorName" : "Hitachi, Ltd.",
    "version" : "01.20.00",
    "favorite" : false,
    "failedCount" : 0,
    "completedCount" : 0,
    "executedCount" : 0,
    "latest" : true,
    "supportedScheduleType" : "immediate, schedule, recurrence",
    "submitCount" : 0,
    "serviceTemplateID" : 5106,
    "serviceGroupID" : 3,
    "supportedActionType" : "forciblyStop, retry"
  } ],
  "count" : 1
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.5.2 サービスの情報取得

### 機能

指定したサービスに関する情報を取得します。

## 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "name" : "service-display-name",
  "description" : "description-text",
  "tags" : "tags"
  "serviceTemplateName" : "service-template-name"
  "createTime" : "created-date-and-time",
  "modifyTime" : "updated-date-and-time",
  "serviceState" : "service-state"
  "serviceGroupName" : "service-group-name",
  "iconURL" : "icon-URL",
  "vendorName" : "vendor-name",
  "version" : "version"
  "lastSubmitTime" : "last-submit-time",
  "favorite" : {true|false},
  "failedCount" : failed-count,
  "completedCount" : completed-count,
  "lastFailedTime" : last-failed-time,
  "resetTime" : reset-time,
  "executedCount" : executed-count,
  "latest" : {true|false},
  "imageURL" : "image-URL",
```

```
"supportedScheduleType" : "supported-schedule-type",
"submitCount" : submit-count,
"serviceTemplateID" : service-template-id,
"serviceGroupID" : service-group-id,
"supportedActionType" : supported-action-type
}
```

## 使用例

instanceID=2015 のサービスに関する情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2015 HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 11:40:06 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
1aa95d66e62d885b5583da3620bd166fd3a3_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : 2015,
  "name" : "testService1",
  "description" : "description",
  "tags" : "",
  "serviceTemplateName" : "testService",
  "createTime" : "2014-07-14T01:16:11.000-0700",
  "modifyTime" : "2014-07-14T04:36:30.000-0700",
  "serviceState" : "release",
  "serviceGroupName" : "DefaultServiceGroup",
  "iconURL" : "http://10.196.184.238:22015/Automation/icon/services/com.hitachi.software/remoteCommandExe/01.20.00",
  "vendorName" : "Hitachi,Ltd.",
  "version" : "01.20.00",
  "lastSubmitTime" : "2014-07-14T01:16:11.000-0700",
  "favorite" : false,
  "failedCount" : 0,
  "completedCount" : 0,
}
```

```

"executedCount" : 0,
"latest" : true,
"supportedScheduleType" : "immediate,schedule,recurrence",
"submitCount" : 0,
"serviceTemplateID" : 5106,
"serviceGroupID" : 3,
"supportedActionType" : "forciblyStop,retry"
}

```

## 2.5.3 サービスの編集

### 機能

指定したサービスを編集します。

この API では、サービスのプロパティ値は変更できません。プロパティ値を更新したい場合は「[2.9.6 プロパティ値の一括更新](#)」または「[2.9.8 プロパティ値の更新](#)」のトピックを参照してください。

Submit ロールを持つユーザーは、プロパティ「favorite」だけを更新できます。Admin ロール、Develop ロール、または Modify ロールの権限を持つユーザーは、すべてのプロパティを更新できます。

### 実行権限

Admin ロール、Develop ロール、Modify ロール、Submit ロール

### API のバージョン

v1

### リクエスト形式

```
PUT http://host:port/Automation/version/objects/Services/id
```

リクエストのスキーマは、API「サービスの情報取得」のレスポンスボディと同じ形式です。「Services」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-49 「Services」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
サービス	Services	1	指定した ID の Services リソース

このオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
Services	name	1

リソース名称	メンバー名	数
Services	description	1
	tags	
	favorite	
	serviceState	
	supportedScheduleType	

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数不正です。または、すでに存在しているサービス名を指定しています。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "name" : "service-display-name",
  "description" : "description-text",
  "tags" : "tags"
  "serviceTemplateName" : "service-template-name"
  "createTime" : "created-date-and-time",
  "modifyTime" : "updated-date-and-time",
  "serviceState" : "service-state"
  "serviceGroupName" : "service-group-name",
  "iconURL" : "icon-URL",
  "vendorName" : "vendor-name",
  "version" : "version"
  "lastSubmitTime" : "last-submit-time",
  "favorite" : {true|false},
  "failedCount" : failed-count,
  "completedCount" : completed-count,
  "executedCount" : executed-count,
  "latest" : {true|false},
  "imageURL" : "image-URL",
  "supportedScheduleType" : "supported-schedule-type",
  "submitCount" : submit-count,
```

```
"serviceTemplateID" : service-template-id,
"serviceGroupID" : service-group-id,
"supportedActionType" : supported-action-type
}
```

## 使用例

instanceID=2015 のサービスを編集する場合の使用例を示します。

リクエストヘッダー:

```
PUT /Automation/v1/objects/Services/2015 HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 11:40:10 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
5929972368348e976584903133f5f8ce93ce2aec_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : 2015,
  "name" : "testService1",
  "description" : "description",
  "tags" : "",
  "serviceTemplateName" : "testService",
  "createTime" : "2014-07-14T01:16:11.000-0700",
  "modifyTime" : "2014-07-14T04:36:30.000-0700",
  "serviceState" : "release",
  "serviceGroupName" : "DefaultServiceGroup",
  "iconURL" : "http://10.196.184.238:22015/Automation/icon/services/com.hitachi.software/remoteCommandExe/01.20.00",
  "vendorName" : "Hitachi,Ltd.",
  "version" : "01.20.00",
  "lastSubmitTime" : "2014-07-14T01:16:11.000-0700",
  "favorite" : false,
  "failedCount" : 0,
  "completedCount" : 0,
  "executedCount" : 0,
  "latest" : true,
```



```
"supportedScheduleType" : "immediate,schedule,recurrence",
"submitCount" : 0,
"serviceTemplateID" : 5106,
"serviceGroupID" : 3,
"supportedActionType" : "forciblyStop,retry"
}
```

## 2.5.4 サービスの削除

### 機能

指定したサービスを削除します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
DELETE http://host:port/Automation/version/objects/Services/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
204	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービスを削除する権限がありません。
409	Conflict	該当のサービスから生成されたタスクがあります。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### 使用例

instanceID=2015 のサービスを削除する場合の使用例を示します。

リクエストヘッダー:

```
DELETE /Automation/v1/objects/Services/2015 HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Content-Type: application/json
Content-Length: 918
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

```
レスポンスヘッダー:
HTTP/1.1 204 No Content
Date: Fri, 07 Aug 2015 09:48:51 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
82b94e7adbdb8cebcb060b12f8c32ee2660a34b_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Content-Length: 0
Content-Type: application/json
```

## 2.5.5 サービスに対する操作一覧の取得

### 機能

指定したサービスに対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。

ステータスコード	メッセージ	説明
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {
    "name" : "update",
    "href" : "http://host:port/Automation/version/objects/Services/id",
    "method" : "PUT",
    "parameters" : []
  }, {
    "name" : "submit",
    "href" : " http://host:port/Automation/version/objects/Services/id/actions/submit/
invoke",
    "method" : "POST",
    "parameters" : []
  }, {
    "name" : "detailhelp",
    "href" : " http://host:port/Automation/version/objects/Services/id/actions/detailhelp",
    "method" : "GET",
    "parameters" : []
  }, {
    "name" : "delete",
    "href" : "http://host:port/Automation/version/objects/Services/id",
    "method" : "DELETE",
    "parameters" : []
  }, {
    "name" : "reset",
    "href" : " http://host:port/Automation/version/objects/Services/id/actions/reset/
invoke",
    "method" : "POST",
    "parameters" : []
  }, {
    "name" : "release",
    "href" : " http://host:port/Automation/version/objects/Services/id/actions/release/
invoke",
    "method" : "POST",
    "parameters" : []
  }, {
    "name" : "maintenance",
    "href" : " http://host:port/Automation/version/objects/Services/id/actions/maintenance/
invoke",
    "method" : "POST",
    "parameters" : []
  }, {
    "name" : "disable",
    "href" : " http://host:port/Automation/version/objects/Services/id/actions/disable/
```

```

    invoke",
      "method" : "POST",
      "parameters" : []
    }, {
      "name" : "applyTemplate",
      "href" : " http://host:port/Automation/version/objects/Services/id/actions
applyTemplate/invoke",
      "method" : "POST",
      "parameters" : []
    } ],
    "count" : 9
  }
}

```

## 使用例

instanceID=2004 のサービスに対して、実行できる操作の一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```

GET /Automation/v1/objects/Services/2004/actions HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 04:40:59 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
72fe74c462e2a50793542df0c0589289ce3f3_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{
  "data" : [ {
    "name" : "update",
    "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004",
    "method" : "PUT",
    "parameters" : [ ]
  }, {
    "name" : "submit",
    "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
submit/invoke",
    "method" : "POST",
    "parameters" : [ ]
  }
]
}

```

```

    }, {
      "name" : "detailhelp",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
detailhelp",
      "method" : "GET",
      "parameters" : [ ]
    }, {
      "name" : "delete",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004",
      "method" : "DELETE",
      "parameters" : [ ]
    }, {
      "name" : "reset",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/reset/
invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "release",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
release/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "maintenance",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
maintenance/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "disable",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
disable/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "applyTemplate",
      "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
applyTemplate/invoke",
      "method" : "POST",
      "parameters" : [ ]
    } ],
    "count" : 9
  }
}

```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.5.6 サービス実行に必要な情報の取得

### 機能

指定したサービスの実行操作に必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/submit
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービスを取得する権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "submit",
  "href" : "http://host:port/Automation/version/objects/Services/id/actions/submit/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

「parameters」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-50 「parameters」のメンバーとして出力されるオブジェクト（サービス実行に必要な情報の取得）

機能	リソース名称	数	説明
スケジュール	Schedule	1	サービスの実行予定
プロパティ値の一覧	PropertyValue	0~n	サービスの入力プロパティ

## 使用例

instanceID=2015 のサービスを実行する準備として、必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2015/actions/submit HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 04:40:59 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
75cdef77cf941edbf5b2934f6afe1e8e18fdb8a_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "submit",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2015/actions/submit/
invoke",
  "method" : "POST",
  "parameters" : [ {
    "name" : "Execute Remote Command",
    "submitter" : "",
    "scheduleType" : "immediate",
    "description" : "",
    "scheduledStartTime" : "2015-07-30T14:51:23.342+09:00",
    "recurrenceInterval" : "daily",
    "recurrenceDayOfWeek" : "",
    "recurrenceDayOfMonth" : "",
    "recurrenceLastDayOfMonth" : false,
    "recurrenceStartDate" : "2015-07-30",
```

```
    "recurrenceTime" : "00:00:00",
    "serviceID" : 5137
  }, {
    "instanceID" : 5112,
    "type" : "string",
    "keyName" : "common.targetHost",
    "value" : "",
    "readOnly" : false,
    "hidden" : false,
    "serviceID" : 5137
  }, {
    "instanceID" : 5135,
    "type" : "string",
    "keyName" : "common.remoteCommand",
    "value" : "",
    "readOnly" : false,
    "hidden" : false,
    "serviceID" : 5137
  }, {
    "instanceID" : 5128,
    "type" : "string",
    "keyName" : "common.remoteCommandParameter",
    "value" : "",
    "readOnly" : false,
    "hidden" : false,
    "serviceID" : 5137
  } ]
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
  - [2.5.7 サービスの実行](#)
- 

## 2.5.7 サービスの実行

### 機能

指定したサービスを実行します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1



## リクエスト形式

```
POST http://host:port/Automation/version/objects/Services/id/actions/submit/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "submit",
  "href" : "http://host:port/Automation/version/objects/Services/id/actions/submit/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

リクエストのスキーマで「parameters」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-51 「parameters」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
スケジュール	Schedule	1	サービスの実行予定
プロパティ値の一覧	PropertyValue	0~n	サービスの入力プロパティ

これらのオブジェクトに指定必須のプロパティについて、次に説明します。どのタイミング（即時実行、指定日時実行または定期実行）でサービスを実行する場合でも、プロパティに次のメンバーを指定できます。

リソース名称	メンバー名	数
Schedule	name	1
	description	
	scheduleType	
PropertyValue	keyName	0~n
	value	

指定日時実行または定期実行の場合は、プロパティに次のメンバーを指定できます。

リソース名称	メンバー名	数	プロパティの指定可否
Schedule	scheduledStartTime	1	指定日時実行の場合に指定できます。
	recurrenceInterval		定期実行の場合に指定できます。
	recurrenceMinutes		
	recurrenceDayOfWeek		
	recurrenceDayOfMonth		

リソース名称	メンバー名	数	プロパティの指定可否
Schedule	recurrenceLastDayOfMonth	1	定期実行の場合に指定できます。
	recurrenceStartDate		
	recurrenceTime		

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
201	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービス実行権限がありません。
404	Not found	サービス取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-52 「affectedResources」のメンバーとして出力されるオブジェクト（サービスの実行）

出力	リソース名称	数	説明
作成されたスケジュールへのリンク	String	1	作成されたスケジュール機能のリソース (Schedules) へのリンク
作成されたタスクへのリンク	String		作成されたタスク機能のリソース (Tasks) へのリンク

## 使用例

instanceID=2015 のサービスを実行する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Services/2015/actions/submit/invoke HTTP/1.1
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 811
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 11:45:34 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
72fe74c462e2a50793542df0c0589289ce3f3_Vlo8Y30JdDBUB3lJJSVParTjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "3d9069ca-444f-4757-b0c5-a57ddd7d44cf",
  "created" : "2014-07-14T04:45:35.293-0700",
  "updated" : "2014-07-14T04:45:35.293-0700",
  "completed" : "2014-07-14T04:45:35.293-0700",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Schedules/2025", "http://10.196.184.182:22015/Automation/v1/objects/Tasks/2026" ]
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.5.8 サービスのカウンターのリセットに必要な情報の取得

### 機能

指定したサービスのカウンターのリセット（統計情報の初期化）に必要な情報を取得します。

## 実行権限

Admin ロール, Develop ロール, Modify ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/reset/
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "reset",
  "href" : " http://host:port/Automation/version/objects/Services/id/actions/reset/invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=2004 のサービスのカウンターをリセットするために必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2004/actions/reset HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
```

Accept-Language: ja

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 04:44:34 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
afc7e78858ad7ff3a8e53c84ac519a7e663b97b4_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "reset",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/reset/
invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 2.5.9 サービスのカウンターのリセット

### 機能

指定したサービスのカウンターをリセット（統計情報を初期化）します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Services/id/actions/reset/invoke
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	カウンターをリセットする権限がありません。
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "reset",
  "href" : " http://host:port/Automation/version/objects/Services/id/actions/reset/invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=2004 のサービスのカウンターをリセットする場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2004/actions/reset HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 04:44:34 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
afc7e78858ad7ff3a8e53c84ac519a7e663b97b4_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "reset",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/reset/
invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 2.5.10 サービスの状態を「リリース」に変更する操作に必要な情報の取得

### 機能

指定したサービスの状態を「リリース」に変更する操作に必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/release
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "release",
  "href" : " http://host:port/Automation/version/objects/Services/id/actions/release/
invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=2004 のサービスの状態を「リリース」に変更するために必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2004/actions/release HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 04:53:56 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
da3af9677bd825b8186bb9d6f0a67f4dbc78d7_VLo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "release",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/release/
invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```



## 2.5.11 サービスの状態を「リリース」に変更

### 機能

指定したサービスの状態を「リリース」に変更します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Services/id/actions/release/invoke
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	サービスを「リリース」に変更できる状態ではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	変更できないサービスです。または、サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

## 使用例

instanceID=2004 のサービスの状態を「リリース」に変更する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Services/2004/actions/release/invoke HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 175
Accept-Language: ja
```

リクエストヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 04:55:39 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
2a40239379d63c60ba2537f856c1673efd23746b_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
Accept-Language: ja
```

レスポンスボディ:

```
{
  "instanceID" : "4c63e655-1ec2-4c70-912f-c1d80be59066",
  "created" : "2015-07-30T13:55:39.457+09:00",
  "updated" : "2015-07-30T13:55:39.457+09:00",
  "completed" : "2015-07-30T13:55:39.457+09:00",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Services/2004" ],
  "result" : [ ]
}
```

### 2.5.12 サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得

#### 機能

指定したサービスの状態を「メンテナンス」に変更する操作に必要な情報を取得します。

## 実行権限

Admin ロール, Develop ロール, Modify ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/maintenance
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "maintenance",
  "href" : " http://host:port/Automation/version/objects/Services/id/actions/maintenance/
invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=2004 のサービスの状態を「メンテナンス」に変更するために必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2004/actions/maintenance HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
```

```
Accept: application/json
```

```
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
```

```
Date: Thu, 30 Jul 2015 05:02:47 GMT
```

```
Server: Cosminexus HTTP Server
```

```
Access-Control-Expose-Headers: WWW-Authenticate
```

```
WWW-Authenticate: HSSO
```

```
2370bb888129f799683dc8289b0484da547fceb_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
```

```
Access-Control-Allow-Origin: *
```

```
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
```

```
Access-Control-Allow-Credentials: true
```

```
Cache-Control: no-cache
```

```
Transfer-Encoding: chunked
```

```
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "maintenance",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/
maintenance/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 2.5.13 サービスの状態を「メンテナンス」に変更

### 機能

指定したサービスの状態を「メンテナンス」に変更します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Services/id/actions/maintenance/invoke
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	サービスを「メンテナンス」に変更できる状態ではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	変更できないサービスです。または、サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

## 使用例

instanceID=2004 のサービスの状態を「メンテナンス」に変更する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Services/2004/actions/maintenance/invoke HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 183
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 05:04:40 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
```

```
WWW-Authenticate: HSSO
23916dfb9e33860332c7e7995f78c2f2507dbf_Vlo8Y30JdDBUB3LjJSVParTjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "36a53982-ac92-45aa-acea-21ede67b7df2",
  "created" : "2015-07-30T14:04:41.028+09:00",
  "updated" : "2015-07-30T14:04:41.028+09:00",
  "completed" : "2015-07-30T14:04:41.028+09:00",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Services/
2004" ],
  "result" : [ ]
}
```

## 2.5.14 サービスの状態を「無効」に変更する操作に必要な情報の取得

### 機能

指定したサービスの状態を「無効」に変更する操作に必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/disable
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。

ステータスコード	メッセージ	説明
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "disable",
  "href" : " http://host:port/Automation/version/objects/Services/id/actions/disable/
invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=2004 のサービスの状態を「無効」に変更するために必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2004/actions/disable HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 05:05:53 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
4ceed74c19dfb6a6c289e561e1c23f5a9088f58_Vl08Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "disable",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2004/actions/disable/
```

```
invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 2.5.15 サービスの状態を「無効」に変更

### 機能

指定したサービスの状態を「無効」に変更します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Services/id/actions/disable/invoke
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	サービスを「無効」に変更できる状態ではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	変更できないサービスです。または、サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
}
```



```
"created" : "created-date-and-time",
"updated" : "updated-date-and-time",
"completed" : "completed-date-and-time",
"state" : "state",
"affectedResources" : [ {...} ],
"result" : [ {...} ],
"resultType" : "result-type"
}
```

## 使用例

instanceID=2004 のサービスの状態を「無効」に変更する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Services/2004/actions/disable/invoke HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 175
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 05:07:57 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
3bae2e194f9f7417a578e3d18492e9ccf94388_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "ff785246-c3c9-425c-87a5-109336e8b387",
  "created" : "2015-07-30T14:07:58.053+09:00",
  "updated" : "2015-07-30T14:07:58.053+09:00",
  "completed" : "2015-07-30T14:07:58.053+09:00",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Services/2004" ],
  "result" : [ ]
}
```

## 2.5.16 サービスの詳細説明の URL を取得

### 機能

指定したサービスの詳細説明を表示する URL を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/detailhelp
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	サービスの取得権限がありません。または、サービスがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "export",
  "href" : "Link-to-the-detail-help",
  "method" : "POST",
  "parameters" : []
}
```

### 使用例

instanceID=2004 のサービスの詳細説明を表示する URL を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2004/actions/detailhelp HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 05:08:56 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
8e609f55fd6858f17ddc4527cd6f890b79153e2_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "detailhelp",
  "href" : "http://10.196.184.182:22015/Automation/services/custom/000000000000560/
remoteCommandExe.html",
  "method" : "GET",
  "parameters" : [ ]
}
```

## 2.5.17 サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得

### 機能

指定したサービスが使用するサービステンプレートのバージョンを変更する操作に必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/Services/id/actions/applyTemplate
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	サービステンプレートの取得権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "applyTemplate",
  "href" : "http://host:port/Automation/version/objects/Services/id/actions/applyTemplate/
invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=2188 のサービスが使用するサービステンプレートのバージョン変更をするために、必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Services/2188/actions/applyTemplate HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 05:21:04 GMT
```

```
Server Cosminexus HTTP Server is not blacklisted
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
95fa1a17b658d5f34912ec64299aadb522e0d6f5_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "applyTemplate",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Services/2188/actions/
applyTemplate/invoke",
  "method" : "POST",
  "parameters" : [ {
    "instanceID" : 2111,
    "keyName" : "SP_GenericApplication",
    "displayName" : "Allocate Volumes for Generic Application",
    "iconURL" : "http://10.196.184.182:22015/Automation/icon/services/
com.hitachi.software.dna.cts/SP_GenericApplication/01.14.00",
    "vendorID" : "com.hitachi.software.dna.cts",
    "version" : "01.14.00",
    "vendorName" : "Hitachi, Ltd.",
    "tags" : "Add New Storage",
    "createTime" : "2015-07-30T14:14:29.000+09:00",
    "modifyTime" : "2015-07-30T14:14:29.000+09:00",
    "description" : "Intelligent allocation service that uses sets of volumes from the
associated infrastructure group to be consumed by server(s) running a generic application",
    "releaseState" : "release",
    "latest" : false,
    "imageUrl" : "http://10.196.184.182:22015/Automation/services/custom/000000000002111/
SP_GenericApplication_overview.png",
    "supportedScheduleType" : "immediate,schedule",
    "needVUP" : false,
    "componentOutdated" : true,
    "usedServices" : 1,
    "usedTemplates" : 0
  } ]
}
```

## 2.5.18 サービスが使用するサービステンプレートのバージョン変更

### 機能

指定したサービスに、任意のバージョンのサービステンプレートを適用します。

## 実行権限

Admin ロール, Develop ロール, Modify ロール

## API のバージョン

v1

## リクエスト形式

```
POST http://host:port/Automation/version/objects/Services/id/actions/applyTemplate/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "applyTemplate",
  "href" : "http://host:port/Automation/version/objects/Services/id/actions/applyTemplate/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

リクエストのスキーマで「parameters」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-53 「parameters」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
サービステンプレート	ServiceTemplate	1	バージョンアップするサービステンプレート

このオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
ServiceTemplate	instanceID	1

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。

ステータスコード	メッセージ	説明
409	Conflict	変更先のサービステンプレートの状態が不正です。
412	Precondition failed	サーバが利用できません。または、テンプレート変更の前提条件を満たしていません。
500	Server-side error	変更できないサービスです。または、サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

## 使用例

instanceID=2188 のサービスが使用するサービステンプレートのバージョン変更をする場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Services/2188/actions/applyTemplate/invoke HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 1199
Expect: 100-continue
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 05:23:38 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
456eb72dda7029ba9cbdf3dd57233a25247d2717_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
```

Cache-Control: no-cache  
Transfer-Encoding: chunked  
Content-Type: application/json

レスポンスボディ:

```
{  
  "instanceID" : "10920fed-ad4f-4be1-9015-bd2066e5312c",  
  "created" : "2015-07-30T14:23:38.683+09:00",  
  "updated" : "2015-07-30T14:23:38.683+09:00",  
  "completed" : "2015-07-30T14:23:38.683+09:00",  
  "state" : "success",  
  "affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Services/  
2188" ],  
  "result" : [ ]  
}
```



## 2.6 スケジュール関連の API

タスクに設定されているスケジュール機能を管理するための操作を説明します。

### 2.6.1 スケジュール一覧の取得

#### 機能

指定したタスクに設定されているスケジュール一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules
```

API の実行ユーザーが権限を持つ、すべてのスケジュールを対象として一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-54 API 「スケジュール一覧の取得」 で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
serviceID	指定した値と等しい
serviceGroupID	
serviceTemplateID	
scheduleStatus*	未実行のタスクのスケジュール情報

#### 注※

未実行のタスクのスケジュール情報を取得したい場合は、scheduleStatus に「running」を指定してください。

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## 指定例

クエリパラメータとして、serviceID に 2015 を指定する例を次に示します。

```
?serviceID=2015
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"スケジュール機能のリソース (Schedules) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのスケジュールを対象として、スケジュール一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Schedules HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:25:42 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
1aa95d66e62d885b5583da3620bd166fd3a3_Vlo8Y30JBWokHUYTEXAMXx5iHgQ=_V0810
```

Access-Control-Allow-Origin: \*  
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS  
Access-Control-Allow-Credentials: true  
Cache-Control: no-cache  
Transfer-Encoding: chunked  
Content-Type: application/json

レスポンスボディ:

```
{
  "data": [ {
    "instanceID": 2060,
    "name": "testService1_20140714044426_Resubmit",
    "submitter": "System",
    "scheduleType": "immediate",
    "createTime": "2014-07-14T05:19:39.000-0700",
    "modifyTime": "2014-07-14T05:19:39.000-0700",
    "description": "",
    "serviceState": "release",
    "serviceID": 2015
  }, {
    "instanceID": 2029,
    "name": "testService1_20140714045613",
    "submitter": "System",
    "scheduleType": "immediate",
    "createTime": "2014-07-14T04:56:15.000-0700",
    "modifyTime": "2014-07-14T04:56:15.000-0700",
    "description": "",
    "serviceState": "release",
    "serviceID": 2015
  }, {
    "instanceID": 2025,
    "name": "testService1_20140714044426",
    "submitter": "System",
    "scheduleType": "immediate",
    "createTime": "2014-07-14T04:45:34.000-0700",
    "modifyTime": "2014-07-14T04:45:34.000-0700",
    "description": "",
    "serviceState": "release",
    "serviceID": 2015
  }, {
    "instanceID": 2056,
    "name": "Execute remote command_20140714045708",
    "submitter": "System",
    "scheduleType": "immediate",
    "createTime": "2014-07-14T04:57:09.000-0700",
    "modifyTime": "2014-07-14T04:57:09.000-0700",
    "description": "",
    "serviceState": "test",
    "serviceID": 2040
  }, {
    "instanceID": 2134,
    "name": "stop_20140714052330",
    "submitter": "System",
    "scheduleType": "immediate",
    "createTime": "2014-07-14T05:23:32.000-0700",
    "modifyTime": "2014-07-14T05:23:32.000-0700",
    "description": ""
  }
]
```

```
    "serviceState" : "test",
    "serviceID" : 2092
  } ],
  "count" : 5
}
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.6.2 スケジュールの情報取得

### 機能

指定したスケジュールに関する情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```

{
  "instanceID" : instance-id,
  "name" : "registered-service-name",
  "submitter" : "submit-user-name",
  "status" : "status-of-schedule",
  "scheduleType" : "type-of-schedule",
  "createTime" : "created-date-and-time",
  "modifyTime" : "updated-date-and-time",
  "description" : "description-text",
  "scheduledStartTime" : "scheduled-start-time",
  "recurrenceInterval" : "interval-type",
  "recurrenceDayOfWeek" : "interval-of-weekly-job",
  "recurrenceDayOfMonth" : "interval-of-monthly-job",
  "recurrenceLastDayOfMonth" : {true|false},
  "recurrenceStartDate" : "recurrence-start-date",
  "recurrenceTime" : "exec-time-of-day",
  "serviceState" : "service-state",
  "serviceID" : service-id
}

```

## 使用例

instanceID=2060 のスケジュールに関する情報を取得する場合の使用例を示します。

リクエストヘッダー:

```

GET /Automation/v1/objects/Schedules/2060 HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:26:19 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
55fb30b1218f2ceec1b52d59d1b77b267895821_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{
  "instanceID" : 2060,
  "name" : "testService1_20140714044426_Resubmit",
  "submitter" : "System",
  "scheduleType" : "immediate",

```

```
"createTime" : "2014-07-14T05:19:39.000-0700",
"modifyTime" : "2014-07-14T05:19:39.000-0700",
"description" : "",
"serviceState" : "release",
"serviceID" : 2015
}
```

## 2.6.3 スケジュールに対する操作一覧の取得

### 機能

指定したスケジュールに対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules/id/actions
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {
    "name" : "cancel",
    "href" : "http://host:/Automation/version/objects/Schedules/id/actions/cancel/invoke",
```

```

    "method" : "POST",
    "parameters" : []
  }, {
    "name" : "suspend",
    "href" : " http://host:port/Automation/version/objects/Schedules/id/actions/suspend/
invoke",
    "method" : "POST",
    "parameters" : []
  }, {
    "name" : "resume",
    "href" : " http://host:port/Automation/version/objects/Schedules/id/actions/resume/
invoke",
    "method" : "POST",
    "parameters" : []
  } ],
  "count" : 3
}

```

## 使用例

instanceID=2193 のスケジュールに対して実行できる操作の一覧を、取得する場合の使用例を示します。

リクエストヘッダー:

```

GET /Automation/v1/objects/Schedules/2193/actions HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:29:28 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{
  "data" : [ {
    "name" : "cancel",
    "href" : "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193/actions/
cancel/invoke",
    "method" : "POST",
    "parameters" : [ ]
  }
]
}

```

```
    }, {
      "name" : "suspend",
      "href" : "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193/actions/
suspend/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "resume",
      "href" : "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193/actions/
resume/invoke",
      "method" : "POST",
      "parameters" : [ ]
    } ],
    "count" : 3
  }
}
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.6.4 スケジュールのキャンセルに必要な情報の取得

### 機能

指定したスケジュールをキャンセルするために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules/id/actions/cancel
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。



ステータスコード	メッセージ	説明
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "cancel",
  "href" : "http://host:port/Automation/version/objects/Schedules/id/actions/cancel/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=2193 のスケジュールをキャンセルするために必要になる情報を、取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Schedules/2193/actions/cancel HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:34:39 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "cancel",
  "href" : "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193/actions/cancel/invoke",
  "method" : "POST",
```

```
} "parameters" : [ ]
```

## 関連項目

- 2.6.5 スケジュールのキャンセル

## 2.6.5 スケジュールのキャンセル

### 機能

指定したスケジュールをキャンセルします。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Schedules/id/actions/cancel/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "cancel",
  "href" : "http://host:port/Automation/version/objects/Schedules/id/actions/cancel/invoke",
  "method" : "POST",
  "parameters" : null
}
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。

ステータスコード	メッセージ	説明
409	Conflict	タスクの状態が待機中または保留中ではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-55 「affectedResources」のメンバーとして出力されるオブジェクト（スケジュールのキャンセル）

出力	リソース名称	数	説明
影響を受けたスケジュールへのリンク	String	1	影響を受けたスケジュール機能のリソース (Schedules) へのリンク

## 使用例

instanceID=2193 のスケジュールをキャンセルする場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Schedules/2193/actions/cancel/invoke HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Content-Type: application/json
Content-Length: 172
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:35:22 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
```

```
WWW-Authenticate: HSSO
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3LjJSVParTjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "7a2924f8-1d5b-4f94-aef0-babccb2eb525",
  "created" : "2014-07-14T05:35:23.113-0700",
  "updated" : "2014-07-14T05:35:23.113-0700",
  "completed" : "2014-07-14T05:35:23.113-0700",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.238:22015/Automation/v1/objects/Schedules/
2193" ]
}
```

## 2.6.6 スケジュールの一時停止に必要な情報の取得

### 機能

指定したスケジュールを一時停止するために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules/id/actions/suspend
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。

ステータスコード	メッセージ	説明
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "suspend",
  "href" : "http://host:port/Automation/version/objects/Schedules/id/actions/suspend/
invoke",
  "method" : "POST",
  "parameters" : null
}
```

## 使用例

instanceID=2193 のスケジュールを一時停止するために必要になる情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Schedules/2193/actions/suspend HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:31:38 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "suspend",
  "href" : "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193/actions/
suspend/invoke",
```

```
"method" : "POST",
"parameters" : [ ]
}
```

## 関連項目

- 2.6.7 スケジュールの一時停止

## 2.6.7 スケジュールの一時停止

### 機能

指定したスケジュールを一時停止します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Schedules/id/actions/suspend/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "suspend",
  "href" : "http://host:port/Automation/version/objects/Schedules/id/actions/suspend/invoke",
  "method" : "POST",
  "parameters" : null
}
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。

ステータスコード	メッセージ	説明
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が待機中ではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-56 「affectedResources」のメンバーとして出力されるオブジェクト（スケジュールの一時停止）

出力	リソース名称	数	説明
影響を受けたスケジュールへのリンク	String	1	影響を受けた Schedules リソースへのリンク

## 使用例

instanceID=2193 のスケジュールを一時停止する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Schedules/2193/actions/suspend/invoke HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Content-Type: application/json
Content-Length: 174
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:32:16 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
```

```
WWW-Authenticate: HSSO
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "3a6ac368-e49c-49ec-ac5b-380370800551",
  "created" : "2014-07-14T05:32:16.519-0700",
  "updated" : "2014-07-14T05:32:16.519-0700",
  "completed" : "2014-07-14T05:32:16.519-0700",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193" ]
}
```

## 2.6.8 スケジュールの再開に必要な情報の取得

### 機能

指定したスケジュールを再開するために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules/id/actions/resume
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	リソースの検索または削除に成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。



ステータスコード	メッセージ	説明
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "resume",
  "href" : "http://host:port/Automation/version/objects/Schedules/id/actions/resume/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=2193 のスケジュールを再開するために必要となる情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Schedules/2193/actions/resume HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:33:15 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "resume",
  "href" : "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2193/actions/resume/invoke",
  "method" : "POST",
```

```
} "parameters" : [ ]
```

## 関連項目

- [2.6.9 スケジュールの再開](#)

## 2.6.9 スケジュールの再開

### 機能

指定したスケジュールを再開します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Schedules/id/actions/resume/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "resume",
  "href" : "http://host:port/Automation/version/objects/Schedules/id/actions/resume/invoke",
  "method" : "POST",
  "parameters" : null
}
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。

ステータスコード	メッセージ	説明
409	Conflict	タスクの状態が保留中ではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-57 「affectedResources」のメンバーとして出力されるオブジェクト（スケジュールの再開）

出力	リソース名称	数	説明
影響を受けたスケジュールへのリンク	String	1	影響を受けたスケジュール機能のリソース (Schedules) へのリンク

## 使用例

instanceID=2193 のスケジュールを再開する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Schedules/2193/actions/resume/invoke HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Content-Type: application/json
Content-Length: 172
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:33:56 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3lJJSVPaRtjBSA=_V0810
```

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "a109b95d-e7ef-4982-ab24-2d062b38e088",
  "created" : "2014-07-14T05:33:56.925-0700",
  "updated" : "2014-07-14T05:33:56.925-0700",
  "completed" : "2014-07-14T05:33:56.925-0700",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.238:22015/Automation/v1/objects/Schedules/
2193" ]
}
```

## 2.7 タスク関連の API

タスクリソースを管理するための操作を説明します。

### 2.7.1 タスク一覧の取得

#### 機能

タスクの一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks
```

API の実行ユーザーが権限を持つ、すべてのタスクを対象として一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-58 API 「タスク一覧の取得」 で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
serviceID	指定した値と等しい。
scheduleID	
serviceGroupID	
serviceTemplateID	
tags	値をすべて含んでいるか。[,] 区切りで複数指定できる。
q	次に示す検索対象のスキーマに対して、値を含んでいるか全文検索する。 <ul style="list-style-type: none"><li>• name</li><li>• submitter</li><li>• description</li><li>• serviceName</li><li>• tags</li></ul>

クエリパラメータ	フィルタ条件
q	<ul style="list-style-type: none"> <li>notes</li> </ul> 半角スペース区切りで複数指定した場合、値をすべて含んでいるか全文検索する。大文字・小文字を区別しない。

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

#### 指定例

クエリパラメータとして、serviceID に 2015 を指定する例を次に示します。

```
?serviceID=2015
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"タスク機能のリソース (Tasks) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのタスクを対象として、タスク一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 02:00:18 GMT
Server Cosminexus HTTP Server is not blacklisted
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
baa98567d9a18be55be1594ea9677ab1da826a3_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 3042,
    "name" : "Execute Remote Command_20150731105831",
    "status" : "waiting",
    "scheduledStartTime" : "2015-07-31T11:30:00.000+09:00",
    "submitter" : "System",
    "submitTime" : "2015-07-31T11:00:06.000+09:00",
    "modifyTime" : "2015-07-31T11:00:06.000+09:00",
    "serviceState" : "release",
    "scheduleType" : "schedule",
    "description" : "",
    "serviceName" : "Execute Remote Command",
    "tags" : "Windows, Linux, Execute Script",
    "serviceGroupName" : "DefaultServiceGroup",
    "toDo" : false,
    "notes" : "",
    "serviceTemplateID" : 560,
    "scheduleID" : 3020,
    "serviceGroupID" : 3,
    "serviceID" : 2004,
    "supportedActionType" : "forciblyStop, retry"
  } ],
  "count" : 1
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.7.2 タスクの情報取得

### 機能

指定したタスクに関する情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "name" : task-name,
  "status" : task-status,
  "startTime" : start-date-and-time,
  "completionTime" : completion-time,
  "scheduledStartTime" : schedule-start-date-and-time,
  "submitter" : submit-user-name,
  "submitTime" : created-date-and-time,
  "modifyTime" : updated-date-and-time,
  "serviceState" : service-state,
  "scheduleType" : schedule-type,
  "description" : description,
  "serviceName" : service-name,
```



```

"tags" : "tags",
"recurrenceInterval" : "recurrenceInterval",
"recurrenceTime" : "recurrenceTime",
"recurrenceStartDate" : "recurrenceStartDate",
"serviceGroupName" : "serviceGroupName",
"todo" : {true|false},
"notes" : "notes",
"stepTime" : "step-time",
"serviceTemplateID" : service-template-id,
"scheduleID" : schedule-id,
"serviceGroupID" : service-group-id,
"serviceID" : service-id,
"supportedActionType" : supported-action-type
}

```

## 使用例

instanceID=3042 のタスクに関する情報を取得する場合の使用例を示します。

リクエストヘッダー:

```

GET /Automation/v1/objects/Tasks/3042 HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 02:02:09 GMT
Server: Cosminexus HTTP Server is not blacklisted
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
bb3f961e88fd1fe908176cbea77a395fcd56_Vlo8Y30JdDBUB3LjJSVpArTjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{
  "instanceID" : 3042,
  "name" : "Execute Remote Command_20150731105831",
  "status" : "waiting",
  "scheduledStartTime" : "2015-07-31T11:30:00.000+09:00",
  "submitter" : "System",
  "submitTime" : "2015-07-31T11:00:06.000+09:00",
  "modifyTime" : "2015-07-31T11:00:06.000+09:00",
  "serviceState" : "release",

```

```

"scheduleType" : "schedule",
"description" : "",
"serviceName" : "Execute Remote Command",
"tags" : "Windows,Linux,Execute Script",
"serviceGroupName" : "DefaultServiceGroup",
"todo" : false,
"notes" : "",
"serviceTemplateID" : 560,
"scheduleID" : 3020,
"serviceGroupID" : 3,
"serviceID" : 2004,
"supportedActionType" : "forciblyStop,retry"
}

```

## 2.7.3 タスクの編集

### 機能

指定したタスクのノートおよび TODO を編集します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
PUT http://host:port/Automation/version/objects/Tasks/id
```

リクエストのスキーマは、API「タスクの情報取得」のレスポンスボディと同じ形式です。「Task」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-59 「Task」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
タスク	Task	1	指定した ID の Task リソース

このオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
Task	notes	1
	todo	

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	タスクを編集する権限がありません。
404	Not found	タスクを取得する権限がありません。または、タスクがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "name" : "task-name",
  "status" : "task-status",
  "startTime" : "start-date-and-time",
  "completionTime" : "completion-time",
  "scheduledStartTime" : "schedule-start-date-and-time",
  "submitter" : "submit-user-name",
  "submitTime" : "created-date-and-time",
  "modifyTime" : "updated-date-and-time",
  "serviceState" : "service-state",
  "scheduleType" : "schedule-type",
  "description" : "description",
  "serviceName" : "service-name",
  "tags" : "tags",
  "recurrenceInterval" : "recurrenceInterval",
  "recurrenceTime" : "recurrenceTime",
  "recurrenceStartDate" : "recurrenceStartDate",
  "serviceGroupName" : "serviceGroupName",
  "todo" : {true|false},
  "notes" : "notes",
  "stepTime" : "step-time",
  "serviceTemplateID" : service-template-id,
  "scheduleID" : schedule-id,
  "serviceGroupID" : service-group-id,
  "serviceID" : service-id,
  "supportedActionType" : supported-action-type
}
```

## 使用例

instanceID=3042 のタスクのノートおよび TODO を編集する場合の使用例を示します。

### リクエストヘッダー:

```
PUT /Automation/v1/objects/Tasks/3042 HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 666
Accept-Language: ja
```

### レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 03:37:03 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
91351d8e544375a67473e7c7494d1aa7c67b24_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

### レスポンスボディ:

```
{
  "instanceID" : 3042,
  "name" : "Execute Remote Command_20150731105831",
  "status" : "completed",
  "startTime" : "2015-07-31T11:30:00.000+09:00",
  "completionTime" : "2015-07-31T11:30:33.000+09:00",
  "scheduledStartTime" : "2015-07-31T11:30:00.000+09:00",
  "submitter" : "System",
  "submitTime" : "2015-07-31T11:00:06.000+09:00",
  "modifyTime" : "2015-07-31T12:37:03.000+09:00",
  "serviceState" : "release",
  "scheduleType" : "schedule",
  "description" : "",
  "serviceName" : "Execute Remote Command",
  "tags" : "Windows,Linux,Execute Script",
  "serviceGroupName" : "DefaultServiceGroup",
  "toDo" : true,
  "notes" : "Notes Test",
  "serviceTemplateID" : 560,
  "scheduleID" : 3020,
  "serviceGroupID" : 3,
  "serviceID" : 2004,
  "supportedActionType" : "forciblyStop,retry"
}* Connection #0 to host 10.196.184.182 left intact
```

## 2.7.4 タスクの削除

### 機能

指定したタスクを削除します。指定したタスクがデバッグタスク以外の場合は、アーカイブするための URL を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
DELETE http://host:port/Automation/version/objects/Tasks/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
204	OK	タスクの削除に成功しました。
303	See Other	タスクを削除するための URL のレスポンスに成功しました。レスポンスヘッダーの「Location」に表示された URL で、アーカイブを実行してください。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	タスクを削除する権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### 使用例

instanceID=5169 のタスクを削除する場合の使用例を示します。

リクエストヘッダー:

```
DELETE /Automation/v1/objects/Tasks/5169 HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Host: 192.168.146.132:22015
Accept: application/json
```

User-Agent:useragent1

Accept-Language: ja

レスポンスヘッダー:

HTTP/1.1 303 See Other

Date: Fri, 07 Aug 2015 07:38:26 GMT

Server: Cosminexus HTTP Server

Access-Control-Expose-Headers: WWW-Authenticate

WWW-Authenticate: HSSO

10a7b94b76e0747b63ee8e0828c186a5d95f699\_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=\_V0810

Access-Control-Allow-Origin: \*

Location: http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/archive

Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS

Access-Control-Allow-Credentials: true

Cache-Control: no-cache

Transfer-Encoding: chunked

Content-Type: text/html;charset=utf-8

レスポンスボディ:

```
<html><head><title>303 See Other</title></head><body><h1>303 See Other</h1></body></html>
```

## 2.7.5 タスク操作一覧の取得

### 機能

指定したタスクに対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。

ステータスコード	メッセージ	説明
404	Not found	権限がありません。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {
    "name" : "update",
    "href" : "http://host:port/Automation/version/objects/Tasks/id",
    "method" : "PUT",
    "parameters" : [ ]
  }, {
    "name" : "delete",
    "href" : "http://host:port/Automation/version/objects/Tasks/id",
    "method" : "DELETE",
    "parameters" : [ ]
  }, {
    "name" : "stop",
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/stop/invoke",
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "forceStop",
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/forceStop/
invoke",
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "resubmit",
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/resubmit/invoke",
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "archive",
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/archive/invoke",
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "response",
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/response/invoke",
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "rerunStart",
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/rerunStart/
invoke",
    "method" : "POST",
    "parameters" : [ ]
  }, {
    "name" : "rerunStepStart",
```

```
    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/rerunStepStart/
invoke",
    "method" : "POST",
    "parameters" : [ ]
  } ],
  "count" : 9
}
```

## 使用例

instanceID=5169 のタスクに対して実行できる操作の一覧を、取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/5169/actions HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 07:32:08 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
10fad7e4bd5eb0e56b4740f5efc08e6dc750d972_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "name" : "update",
    "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169",
    "method" : "PUT",
    "parameters" : [ ]
  }, {
    "name" : "delete",
    "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169",
    "method" : "DELETE",
    "parameters" : [ ]
  }, {
    "name" : "stop",
    "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/stop/
invoke",
    "method" : "POST",
    "parameters" : [ ]
  } ]
}
```



```

    }, {
      "name" : "forceStop",
      "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/forceStop/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "resubmit",
      "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/resubmit/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "archive",
      "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/archive/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "response",
      "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/response/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "rerunStart",
      "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/rerunStart/invoke",
      "method" : "POST",
      "parameters" : [ ]
    }, {
      "name" : "rerunStepStart",
      "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/rerunStepStart/invoke",
      "method" : "POST",
      "parameters" : [ ]
    } ],
    "count" : 9
  }
}

```

## 2.7.6 タスク実行停止に必要な情報の取得

### 機能

指定したタスクを実行停止するために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/stop
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	リソースの検索または削除に成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "stop",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/stop/invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=2026 のタスクを実行停止するために必要となる情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/2026/actions/stop HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:21:37 GMT
```

```
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
f214b39fba479af17375f1ad0e052124041ea60_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "stop",
  "href" : "http://10.196.184.238:22015/Automation/v1/objects/Tasks/2026/actions/stop/
invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

---

## 関連項目

- [2.7.7 タスクの実行停止](#)

## 2.7.7 タスクの実行停止

### 機能

指定したタスクを実行停止します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/stop/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "stop",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/stop/invoke",
```

```
"method" : "POST",
"parameters" : null
}
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が実行中、応答待ち中、異常検出のどれでもありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-60 「affectedResources」のメンバーとして出力されるオブジェクト（タスクの実行停止）

出力	リソース名称	数	説明
影響を受けたタスクへのリンク	String	1	更新されたタスク機能のリソース (Tasks) へのリンク

## 使用例

instanceID=2026 のタスクを実行停止する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Tasks/2026/actions/stop/invoke HTTP/1.1
Host: 10.196.184.238:22015
```

```
Accept: application/json
Content-Type: application/json
Content-Length: 164
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:23:58 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 d3b775e19041295c9834a332f7936467d94358e_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "f550ef02-b4f8-4332-95da-3b685f2cedf8",
  "created" : "2014-07-14T05:23:59.222-0700",
  "updated" : "2014-07-14T05:23:59.222-0700",
  "completed" : "2014-07-14T05:23:59.222-0700",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.238:22015/Automation/v1/objects/Tasks/2026" ]
}
```

## 2.7.8 タスクの強制停止に必要な情報の取得

### 機能

指定したタスクを強制停止するために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/forceStop
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "forceStop",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/forceStop/invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=5283 のタスクを強制停止するために必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/5283/actions/forceStop HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 09:57:14 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
9bf53394a45188743ac8b2522efcc67284cd_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "forceStop",
  "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5283/actions/forceStop/
invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 2.7.9 タスクの強制停止

### 機能

指定したタスクを強制停止します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/forceStop/invoke
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が実行中、応答待ち中、異常検出のどれでもありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ ]
}
```

## 使用例

instanceID=5381 のタスクを強制停止する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Tasks/5381/actions/forceStop/invoke HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Content-Type: application/json
Content-Length: 175
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
```

```
{
  "name" : "forceStop",
  "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5381/actions/forceStop/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

Accept-Language: ja

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 10:00:39 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
a2e8ab6f7a9c35323fb7d9331735a9419235ebad_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:



```
{
  "instanceID" : "68451399-53c2-4f6b-bbdd-be025a61ed02",
  "created" : "2015-08-07T19:00:40.025+09:00",
  "updated" : "2015-08-07T19:00:40.025+09:00",
  "completed" : "2015-08-07T19:00:40.025+09:00",
  "state" : "success",
  "affectedResource" : [ "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5381" ],
  "result" : [ ]
}
```

## 2.7.10 タスクの再実行に必要な情報の取得

### 機能

指定したタスクを再実行するために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/resubmit
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	リソースの検索または削除に成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "resubmit",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/resubmit/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

「parameters」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-61 「parameters」のメンバーとして出力されるオブジェクト（タスクの再実行に必要な情報の取得）

機能	リソース名称	数	説明
スケジュール	Schedule	1	サービスの実行予定
プロパティ値の一覧	PropertyValue	0~n	サービスの入力プロパティ

## 使用例

instanceID=2026 のタスクを再実行するために必要となる情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/2026/actions/resubmit HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:03:20 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 c733364e62b52913e477addabfbf8c55f9de831_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "resubmit",
  "href" : "http://10.196.184.238:22015/Automation/v1/objects/Tasks/2026/actions/resubmit/invoke",
  "method" : "POST",
  "parameters" : [ {
    "name" : "testService1_20140714044426_Resubmit",
```

```
    "submitter" : "System",
    "scheduleType" : "immediate",
    "description" : "",
    "scheduledStartTime" : "2014-07-14T05:03:20.441-07:00",
    "recurrenceInterval" : "daily",
    "recurrenceDayOfWeek" : "",
    "recurrenceDayOfMonth" : "",
    "recurrenceLastDayOfMonth" : false,
    "recurrenceStartDate" : "2014-07-14",
    "recurrenceTime" : "00:00:00",
    "serviceID" : 2015
  }, {
    "instanceID" : 2012,
    "type" : "string",
    "keyName" : "testProp",
    "value" : "defaultValue",
    "serviceID" : 2015
  } ]
}
```

---

## 関連項目

- 2.2.14 リソースのメンバー
  - 2.7.11 タスクの再実行
- 

## 2.7.11 タスクの再実行

### 機能

指定したタスクを再実行します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/resubmit/
invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "resubmit",
```

```

    "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/resubmit/invoke",
    "method" : "POST",
    "parameters" : [ {...} ]
  }

```

リクエストのスキーマで「parameters」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-62 「parameters」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
スケジュール	Schedule	1	サービスの実行予定
プロパティ値の一覧	PropertyValue	0~n	サービスの入力プロパティ

これらのオブジェクトに指定必須のプロパティについて、次に説明します。

共通の設定の場合

表 2-63 共通の設定の場合

リソース名称	メンバー名	数
Schedule	name	1
Schedule	description	
Schedule	scheduleType	
PropertyValue	keyName	0~n
PropertyValue	value	

[即時実行] の場合

プロパティの指定は不要です。

[指定日時実行] の場合

表 2-64 指定日時実行の場合

リソース名称	メンバー名	数
Schedule	scheduledStartTime	1

[定期実行] の場合

表 2-65 定期実行の場合

リソース名称	メンバー名	数
Schedule	recurrenceInterval	1
Schedule	recurrenceMinutes	
Schedule	recurrenceDayOfWeek	

リソース名称	メンバー名	数
Schedule	recurrenceDayOfMonth	1
Schedule	recurrenceLastDayOfMonth	
Schedule	recurrenceStartDate	
Schedule	recurrenceTime	

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が正常終了またはキャンセルではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-66 「affectedResources」のメンバーとして出力されるオブジェクト（タスクの再実行）

出力	リソース名称	数	説明
作成されたスケジュールへのリンク	String	1	作成されたスケジュール機能のリソース (Schedules) へのリンク
作成されたタスクへのリンク	String		作成されたタスク機能のリソース (Tasks) へのリンク

## 使用例

instanceID=2026 のタスクを再実行する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Tasks/2026/actions/resubmit/invoke HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Content-Type: application/json
Content-Length: 821
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:19:39 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 5011884058b535482bf6bac7390956be5fc2122_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "d2a2284f-9a94-4be0-8813-e5f991762740",
  "created" : "2014-07-14T05:19:40.089-0700",
  "updated" : "2014-07-14T05:19:40.089-0700",
  "completed" : "2014-07-14T05:19:40.089-0700",
  "state" : "success",
  "affectedResource" : [ "http://10.196.184.238:22015/Automation/v1/objects/Schedules/2060", "http://10.196.184.238:22015/Automation/v1/objects/Tasks/2063" ],
  "result" : [ ]
}
```

### 2.7.12 応答待ち中のタスクに応答するために必要な情報の取得

#### 機能

応答待ち中のタスクに応答するために必要な情報を取得します。取得するのは、指定した ID のタスクの中で、応答待ち中になった日時が最も古いステップの情報です。

## 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/response
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が応答待ちではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "response",
  "href" : " http://host:port/Automation/version/objects/Tasks/id/actions/response/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=3179 のタスクに応答するために、必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/3179/actions/response HTTP/1.1
Authorization: Basic c3lzdGVtOmlhbmFnZXI=
```

```
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 04:36:56 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
13691d353edd835f6f83942ec70f4ae1411a3f_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "response",
  "href" : "http://10.196.184.182:22015/Automation/v1/objects/Tasks/3179/actions/response/
invoke",
  "method" : "POST",
  "parameters" : [ {
    "instanceID" : 3239,
    "dialogText" : "",
    "labelButton0" : "OK",
    "screenURL" : "services/default/index.jsp",
    "taskID" : 3179
  } ]
}
```

## 2.7.13 応答待ち中のタスクに応答

### 機能

指定した ID のタスクの中で、応答待ち中になった日時が最も古いステップに応答入力します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1



## リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/response/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "enter Response",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/response/invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

リクエストのスキーマで「parameters」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-67 「parameters」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
タスク	ResponseInput	1	応答入力

このオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
ResponseInput	instanceId	1
	labelbuttonX**	
	taskId	

注※

Xには任意の番号が入ります。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が応答待ち中ではありません。
412	Precondition failed	サーバが利用できません。

ステータスコード	メッセージ	説明
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

## 使用例

instanceID=3179 のタスクに応答する場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Tasks/3179/actions/response/invoke HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Content-Type: application/json
Content-Length: 329
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 04:42:14 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
5d4cd25fd30d1b8d6b67f2d7b4cc5479a16364f_Vl08Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "4fb38028-81d7-4573-851a-672e7524a4fc",
  "created" : "2015-07-31T13:42:15.030+09:00",
```

```
"updated" : "2015-07-31T13:42:15.030+09:00",
"completed" : "2015-07-31T13:42:15.030+09:00",
"state" : "success",
"affectedResource" : [ "http://10.196.184.182:22015/Automation/v1/objects/Tasks/3179" ],
"result" : [ ]
}
```

## 2.7.14 タスクのリトライに必要な情報の取得（失敗したステップからリトライ）

### 機能

タスクを指定し、失敗したステップからリトライするために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/rerunStart
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "rerunStart",
```

```
"href" : "http://host:port/Automation/version/objects/Tasks/id/actions/rerunStart/
invoke",
"method" : "POST",
"parameters" : [ {...} ]
}
```

## 使用例

instanceID=5381 のタスクを、失敗したステップからリトライするために必要となる情報を、取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/5381/actions/rerunStart HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: useragent1
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 10:16:10 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
f261dfd5d7e3befa74903ab7318a59455a86df3_Vl08Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "rerunStart",
  "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5381/actions/
rerunStart/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

---

## 関連項目

- [2.7.15 失敗したステップからリトライ](#)
-

## 2.7.15 失敗したステップからリトライ

### 機能

タスクを指定し、失敗したステップからリトライします。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/rerunStart/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "rerunStart",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/rerunStart/invoke",
  "method" : "POST",
  "parameters" : null
}
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が失敗ではありません。
412	Precondition failed	サーバが起動していません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```

{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
  "result" : [ {...} ],
  "resultType" : "result-type"
}

```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-68 「affectedResources」のメンバーとして出力されるオブジェクト（失敗したステップからリトライ）

出力	リソース名称	数	説明
影響を受けたタスクへのリンク	String	1	更新されたタスク機能のリソース (Tasks) へのリンク

## 使用例

instanceID=5381 のタスクを、失敗したステップからリトライする場合の使用例を示します。

リクエストヘッダー:

```

POST /Automation/v1/objects/Tasks/5381/actions/rerunStart/invoke HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Content-Type: application/json
Content-Length: 177
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja

```

レスポンスヘッダー:

```

HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 10:19:44 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
dfd342179388629104cd0bb13d288884bed541b_Vl08Y30JBW0KHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

```

レスポンスボディ:

```

{
  "instanceID" : "17356cf0-f709-4561-a56b-17a6fbc321e3",

```

```
"created" : "2015-08-07T19:19:44.552+09:00",
"updated" : "2015-08-07T19:19:44.552+09:00",
"completed" : "2015-08-07T19:19:44.552+09:00",
"state" : "success",
"affectedResource" : [ "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5381" ],
"result" : [ ]
}
```

## 2.7.16 タスクのリトライに必要な情報の取得（失敗した次のステップからリトライ）

### 機能

タスクを指定し、失敗した次のステップからリトライするために必要な情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/rerunStepStart
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "rerunStart",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/rerunStepStart/
invoke",
  "method" : "POST",
  "parameters" : [ {...} ]
}
```

## 使用例

instanceID=5381 のタスクを、失敗した次のステップからリトライするために、必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/5381/actions/rerunStepStart HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Content-Type: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 10:24:44 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
7abadbb2b4c4d9c1cf18e5465654ef786a9851_Vl08Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "rerunStepStart",
  "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5381/actions/
rerunStepStart/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

---

## 関連項目

- [2.7.17 失敗した次のステップからリトライ](#)
-



## 2.7.17 失敗した次のステップからリトライ

### 機能

タスクを指定し、失敗した次のステップからリトライします。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/rerunStepStart/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "rerunStepStart",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/rerunStepStart/invoke",
  "method" : "POST",
  "parameters" : null
}
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が失敗ではありません。
412	Precondition failed	サーバが起動していません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造は、次のようになっています。

```
{
  "instanceId" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ]
  "result" : [ {...} ],
  "resultType" : "result-type"
}
```

「affectedResources」のメンバーとして出力されるオブジェクトを、次の表で説明します。

表 2-69 「affectedResources」のメンバーとして出力されるオブジェクト（失敗した次のステップからリトライ）

出力	リソース名称	数	説明
影響を受けたタスクへのリンク	String	1	更新されたタスク機能のリソース (Tasks) へのリンク

## 使用例

instanceID=5381 のタスクを失敗した次のステップからリトライする場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Tasks/5381/actions/rerunStepStart/invoke HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Content-Type: application/json
Content-Length: 185
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 10:29:33 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
c19a775746fbd61d1efd3658d2b4eacadcf435_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "852af753-989f-4797-a7dc-50faaf07b896",
  "created" : "2015-08-07T19:29:33.562+09:00",
  "updated" : "2015-08-07T19:29:33.562+09:00",
  "completed" : "2015-08-07T19:29:33.562+09:00",
  "state" : "success",
  "affectedResource" : [ "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5381" ],
  "result" : [ ]
}
```

## 2.7.18 タスクのアーカイブに必要な情報の取得

### 機能

指定したタスクをアーカイブするために必要な引数のテンプレートを取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tasks/id/actions/archive
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "name" : "archive",
  "href" : " http://host:port/Automation/version/objects/Tasks/id/actions/archive/invoke",
  "method" : "POST",
  "parameters" : []
}
```

## 使用例

instanceID=5169 のタスクをアーカイブするために、必要な情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tasks/5169/actions/archive HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 07:28:17 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
58791edf45552caa5592c652b533c730df4b708_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "name" : "archive",
  "href" : "http://192.168.146.132:22015/Automation/v1/objects/Tasks/5169/actions/archive/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## 2.7.19 タスクのアーカイブ

### 機能

指定したタスクをアーカイブします。

## 実行権限

Admin ロール, Develop ロール, Modify ロール

## API のバージョン

v1

## リクエスト形式

```
POST http://host:port/Automation/version/objects/Tasks/id/actions/archive/invoke
```

リクエストボディの構造は、次のようになっています。

```
{
  "name" : "archive",
  "href" : "http://host:port/Automation/version/objects/Tasks/id/actions/archive/invoke",
  "method" : "POST",
  "parameters" : [ ]
}
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
409	Conflict	タスクの状態が正常終了、失敗、またはキャンセルではありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : "instance-id",
  "created" : "created-date-and-time",
  "updated" : "updated-date-and-time",
  "completed" : "completed-date-and-time",
  "state" : "state",
  "affectedResources" : [ {...} ],
}
```

```
"result" : [ {...} ],
"resultType" : "result-type"
}
```

## 使用例

instanceID=5209 のタスクをアーカイブする場合の使用例を示します。

リクエストヘッダー:

```
POST /Automation/v1/objects/Tasks/5209/actions/archive/invoke HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Content-Type: application/json
Content-Length: 171
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 08:15:46 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
aec4a069aea32fe6d59c8325bfae96af27dde14_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : "0fea3bf2-9747-4d29-a4bf-faaddf22076d",
  "created" : "2015-08-07T17:15:46.474+09:00",
  "updated" : "2015-08-07T17:15:46.474+09:00",
  "completed" : "2015-08-07T17:15:46.474+09:00",
  "state" : "success",
  "affectedResource" : [ "http://192.168.146.132:22015/Automation/v1/objects/TaskHistories/5237" ],
  "result" : [ ]
}
```

## 2.7.20 ステップ一覧の取得

### 機能

指定したタスクに含まれるステップのうち、[タスク詳細] 画面に表示されるステップの一覧を取得します。これは、V11 より前の JP1/AO のための API です。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/FlowSteps
```

クエリパラメータとして taskID を必ず指定してください。

taskID を指定することで対象のタスクを絞り込みます。そのタスクに含まれるステップのうち、[タスク詳細] 画面に表示されるステップの一覧を取得できます。クエリパラメータの指定がない場合、または複数のクエリパラメータを指定した場合はエラーになります。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-70 API「ステップ一覧の取得」に指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
taskID	指定した値と等しい

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

### 指定例

クエリパラメータとして taskID に 512 を指定する例を次に示します。

```
?taskID=512
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが起動していません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"ステップ機能のリソース (FlowSteps) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

タスク ID 「5381」 のタスクが実行するステップの一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/FlowSteps?taskID=5381 HTTP/1.1
Host: 10.196.184.238:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVtOmhbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:51:18 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
f1bd56cdd5e340caa0d6f2419205ba81b3317ef_VLo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
```



```

    "data" : [ {
      "instanceID" : "remoteHostCommandExe_2052",
      "name" : "remoteHostCommandExe",
      "startTime" : "2014-07-14 04:57:10",
      "completionTime" : "2014-07-14 04:57:34",
      "jobStatus" : "normal",
      "comment" : "Executes a command on the remote execution target server and displays the
results.",
      "stepStatus" : "complete"
    } ],
    "count" : 1
  }

```

## 2.7.21 タスクログの取得

### 機能

指定したタスクのタスクログを取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/TaskLogs
```

API の実行ユーザーが権限を持つすべてのタスクを対象に、指定した taskID のタスクログを取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。

指定した taskID のタスクログを、offset に指定した値を起点にし、最大で readSize 分取得します。reverse を指定した場合は、offset から逆方向に readSize 分取得します。クエリパラメータは、次の形式で指定してください。

なお、ログは行単位で取得されます。指定したログサイズに達した場合は、取得中の行は取得されません。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-71 API 「タスクログの取得」 に指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
taskID	指定した値と等しい

クエリパラメータ	フィルタ条件
readSize	指定した値と等しい (ただし、タスクログの最後まで)
offset	指定した値と等しい
reverse*	offset からの取得方向を逆にする

#### 注※

reverse には、値を指定しないでください。

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

#### 指定例

taskID=512 のタスクログを、読み出し位置 3,000 バイトから逆方向に 1,000 バイト分、取得する場合の指定例を次に示します。

```
?taskID=512&offset=3000&readSize=1000&reverse
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	Created	成功しました。
400	Bad Request	引数が不正です。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが起動していません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"タスクログ機能のリソース (Tasklogs) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

taskID=2052 のタスクログを、最初 (読み出し位置 0 バイト) から 5,000,000 バイト分、取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/TaskLogs?taskID=2052&offset=0&readSize=5000000 HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOmlhbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:42:47 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 a6daa83f23a873c39c7a65069728796ce882153_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 2052,
    "text" : "**** Windows Server 2008 R2 6.1
TZ=America/Los Angeles
2014/07/14 04:57:31.730¥r¥n
yyyy/mm/dd hh:mm:ss.sss                pid      tid      message-id
message(LANG=ja)¥r¥n
8947 2014/07/14 04:57:31.746      Automation      27A4F43E 12252248 KNAE08001-I      部
品実行を開始しました (タスク名: Execute remote command_20140714045708, タスクID: 2052, ス
テップID: /remoteHostCommandExe, 実行ID: @A102)。¥r¥n
9101 2014/07/14 04:57:31.824      Automation      27A4F43E 12252248 KNAE08129-I      汎
用コマンド実行を開始しました (コマンド: a)。¥r¥n
9103 2014/07/14 04:57:31.824      Automation      27A4F43E 12252248 KNAE08070-I
root権限に昇格する設定 (SSH接続用) が有効になっています。¥r¥n
9140 2014/07/14 04:57:34.433      Automation      27A4F43E 12252248 KNAE08131-E      ER 汎
用コマンド実行に失敗しました (コマンド: a, 部品戻り値: 77)。¥r¥n
9148 2014/07/14 04:57:34.449      Automation      27A4F43E 12252248 KNAE08002-I      部
品実行を終了しました (タスク名: Execute remote command_20140714045708, タスクID: 2052, ス
テップID: /remoteHostCommandExe, 実行ID: @A102, 部品戻り値: 77)。¥r¥n
9152 2014/07/14 04:57:34.449      Automation      27A4F43E 12252248 KNAE08004-I
schema_version=1.1¥r¥n9154 2014/07/14 04:57:34.449      Automation      27A4F43E 12252248
KNAE08004-I      vendor=com.hitachi.software.dna¥r¥n
9156 2014/07/14 04:57:34.449      Automation      27A4F43E 12252248 KNAE08004-I
name=ExecuteCommandPlugin¥r¥n9158 2014/07/14 04:57:34.449      Automation      27A4F43E
12252248 KNAE08004-I      version=01.10.00¥r¥n
9160 2014/07/14 04:57:34.449      Automation      27A4F43E 12252248 KNAE08004-I
plugin_type=javaClass¥r¥n9162 2014/07/14 04:57:34.449      Automation      27A4F43E
12252248 KNAE08004-I      vendor_display_name=(株)日立製作所¥r¥n
9164 2014/07/14 04:57:34.449      Automation      27A4F43E 12252248 KNAE08004-I
display_name=汎用コマンド実行部品¥r¥n
```

```

9166 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
short_description=コマンドラインを対象ホストで実行する部品です。¥r¥n
9168 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
tags=Basic,Hitachi¥r¥n9170 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248
KNAE08004-I enable_SSH_charset_detection=true¥r¥n
9172 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.bcc, value=¥r¥n
9174 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.cc, value=¥r¥n
9177 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.from, value=*****¥r¥n
9179 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.notify, value=false¥r¥n
9182 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.smtp.password, value=*****¥r¥n
9184 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.smtp.port, value=25¥r¥n
9186 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.smtp.server, value=¥r¥n
9188 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.smtp.userid, value=¥r¥n
9190 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.mail.to, value=¥r¥n
9192 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=com.hitachi.software.dna.sys.task.log.level, value=10¥r¥n
9194 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=common.remoteCommand, value=a¥r¥n
9196 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=common.remoteCommandParameter, value=a¥r¥n
9198 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=common.stdoutProperty, value=¥r¥n
9200 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=common.targetHost, value=a¥r¥n
9202 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=foreach.max_value, value=3¥r¥n
9204 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=mail.plugin.retry.interval, value=10¥r¥n
9206 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=mail.plugin.retry.times, value=3¥r¥n
9208 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=plugin.private.builtin.baseUrl, value=http://WIN-JLTV0PQLK2A:22015/Automation/¥r¥n
9210 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.service.category, value=0S_Operations/Basic¥r¥n
9212 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.service.name, value=Execute remote command¥r¥n
9214 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.service.resourceGroupName, value=DefaultServiceGroup¥r¥n
9216 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.step.path, value=/remoteHostCommandExe¥r¥n
9218 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.step.prevReturnCode, value=0¥r¥n
9220 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.task.description, value=¥r¥n
9222 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.task.dir, value=C:¥¥Program Files¥¥HiCommand¥¥Automation¥¥data¥¥task¥
¥2052¥r¥n
9224 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.task.id, value=2052¥r¥n

```

```

9226 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.task.name, value=Execute remote command 20140714045708¥r¥n
9228 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.task.submitter, value=System¥r¥n
9230 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=reserved.task.url, value=http://WIN-JLTV0PQLK2A:22015/Automation/launcher/
TaskDetails?task_id=2052¥r¥n
9232 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=account, value=¥r¥n
9234 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=commandLine, value=a¥r¥n
9236 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=commandLineParameter, value=a¥r¥n
9238 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=credentialType, value=destination¥r¥n
9240 2014/07/14 04:57:34.449 Automation 27A4F43E 12252248 KNAE08004-I
property=destinationHost, value=a¥r¥n
9243 2014/07/14 04:57:34.457 Automation 27A4F43E 12252248 KNAE08004-I
property=password, value=*****¥r¥n
9245 2014/07/14 04:57:34.457 Automation 27A4F43E 12252248 KNAE08004-I
property=stdoutPattern1, value=((?s).*)¥r¥n
9247 2014/07/14 04:57:34.457 Automation 27A4F43E 12252248 KNAE08004-I
property=stdoutPattern2, value=¥r¥n
9249 2014/07/14 04:57:34.457 Automation 27A4F43E 12252248 KNAE08004-I
property=stdoutPattern3, value=¥r¥n
9252 2014/07/14 04:57:34.457 Automation 27A4F43E 12252248 KNAE08004-I
property=suPassword, value=*****¥r¥n
9254 2014/07/14 04:57:34.457 Automation 27A4F43E 12252248 KNAE08009-I 部
品の標準出力はありません。 ¥r¥n
9258 2014/07/14 04:57:34.699 Automation 27A4F43E 12252248 KNAE08014-E ER 指
定された接続先ホスト名の名前解決に失敗しました (詳細情報: a [errno=11004,
syscall=getaddrinfo])。
指定された接続先ホスト名の名前解決ができません。接続先ホスト名の指定値、ネットワークおよび
DNSの設定を確認して、再度サービスを実行してください。 ¥r¥n
9260 2014/07/14 04:57:34.699 Automation 27A4F43E 12252248 KNAE08016-E ER 部
品実行で障害が発生しました (タスク名: Execute remote command 20140714045708, タスクID:
2052, ステップID: /remoteHostCommandExe, 実行ID: @A102, 部品戻り値: 77)。 ¥r¥n
次の要因が考えられます。 ¥r¥n・部品実行で何らかの障害が発生した。 ¥r¥n・タスクの強制停止操作が
行われた。 ¥r¥n
・本製品の停止操作が行われた。 ¥r¥n
タスクログおよび公開ログ(Server[n]. log)の発生時刻前後のメッセージを確認して対処してくださ
い。タスクの強制停止や本製品の停止操作の形跡がなく、
発生時刻前後にエラーメッセージも出力されていない場合は、資料採取ツールで資料を採取し、システ
ム管理者に連絡してください。 ¥r¥n
**** Windows Server 2008 R2 6.1
TZ=America/Los_Angeles 2014/07/14 04:58:17.793¥r¥n
yyyy/mm/dd hh:mm:ss.sss pid tid message-id
message(LANG=ja)¥r¥n
9913 2014/07/14 04:58:17.793 Automation 27A4F43E 1A87FBF6 KNAE08904-I 失
敗した次のステップからリトライしました (ユーザーID: System)。 ¥r¥n",
"totalSize" : 9477,
"readSize" : 9477,
"lineCount" : 66,
"offset" : 0,
"reverse" : false
} ],

```

```
"count" : 1  
}
```

## 2.8 履歴関連の API 一覧

### 2.8.1 履歴一覧の取得

#### 機能

履歴の一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/TaskHistories
```

API の実行ユーザーが権限を持つ、すべての履歴を対象として一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-72 API 「履歴一覧の取得」 で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
start	指定した値と「startTime」が同じか、指定した値が「startTime」以降。
end	指定した値と「completionTime」が同じか、指定した値が「completionTime」以前。
serviceGroupID	指定した値と等しい。
tags	値をすべて含んでいるか。[,] 区切りで複数指定できる。
q	次に示す検索対象のスキーマに対して、値を含んでいるか全文検索する。 <ul style="list-style-type: none"><li>• name</li><li>• submitter</li><li>• serviceName</li><li>• tags</li><li>• description</li><li>• notes</li></ul> 半角スペース区切りで複数指定した場合、値をすべて含んでいるか全文検索する。大文字・小文字を区別しない。

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"履歴機能のリソース (TaskHistories) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべての履歴を対象として一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/TaskHistories HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 06:22:25 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
24f87c98d12f4f434cf398edcbe582939cee4d6_Vlo8Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
```



Transfer-Encoding: chunked  
Content-Type: application/json

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 4006,
    "name" : "Execute Remote Command_20150731105831",
    "submitter" : "System",
    "serviceName" : "Execute Remote Command",
    "tags" : "Windows,Linux,Execute Script",
    "scheduleType" : "schedule",
    "scheduledStartTime" : "2015-07-31T11:30:00.000+09:00",
    "startTime" : "2015-07-31T11:30:00.000+09:00",
    "completionTime" : "2015-07-31T11:30:33.000+09:00",
    "archiveTime" : "2015-07-31T15:22:21.000+09:00",
    "taskID" : 3042,
    "submitTime" : "2015-07-31T11:00:06.000+09:00",
    "status" : "completed",
    "description" : "",
    "serviceState" : "release",
    "todo" : true,
    "notes" : "Notes Test",
    "serviceGroupName" : "DefaultServiceGroup",
    "serviceGroupID" : 3
  } ],
  "count" : 1
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.8.2 履歴の削除 (条件指定)

### 機能

クエリパラメータの条件に従って、履歴を削除します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

## リクエスト形式

```
DELETE http://host:port/Automation/version/objects/TaskHistories
```

クエリパラメータを指定して、削除する履歴を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-73 API「履歴の削除（条件指定）」で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
start	指定した値と「startTime」が同じか、指定した値が「startTime」以降
end	指定した値と「completionTime」が同じか、指定した値が「completionTime」以前
serviceGroupID	指定した値と等しい

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
204	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	履歴を削除する権限がありません。
412	Precondition failed	サーバが起動していません。
500	Server-side error	サーバ処理エラーが発生しました。

## 使用例

2015年7月31日から2015年8月31日までの期間の serviceGroupID=1451 の履歴を削除する場合の使用例を示します。

リクエストヘッダー:

```
DELETE /Automation/v1/objects/TaskHistories?  
serviceGroupID=1451&start=2015-07-31T11:30:00.000+09:00&end=2015-08-31T11:30:00.000+09:00  
HTTP/1.1  
Host: 192.168.146.132:22015  
User-Agent: curl/7.36.0
```

```
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 204 No Content
Date: Fri, 07 Aug 2015 11:17:40 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
e949c7e079a0bc9a137cd1bf3515c72685a506a_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Content-Length: 0
Content-Type: application/json
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.8.3 履歴の取得

### 機能

指定した ID の履歴を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/TaskHistories/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。

ステータスコード	メッセージ	説明
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "name" : "task-name",
  "submitter" : "submit-user-name",
  "serviceName" : "service-name",
  "tags" : "tags",
  "scheduleType" : "type-of-schedule",
  "scheduledStartTime" : "schedule-start-date-and-time",
  "startTime" : "start-date-and-time",
  "completionTime" : "completion-date-and-time",
  "stepStartTime" : "step-start-time",
  "recurrenceInterval" : "interval-type",
  "recurrenceDayOfWeek" : "interval-of-weekly-job",
  "recurrenceDayOfMonth" : "interval-of-monthly-job",
  "recurrenceLastDayOfMonth" : {true|false},
  "recurrenceTime" : "exec-time-of-day",
  "archiveTime" : "removed-date-and-time",
  "taskID" : task-id,
  "submitTime" : "submit-date-and-time",
  "recurrenceStartDate" : "recurrence-start-date-and-time",
  "status" : "task-status",
  "description" : "description",
  "serviceState" : "service-state",
  "toDo" : {true|false},
  "notes" : "notes",
  "serviceGroupName" : "service-group-name",
  "serviceGroupID" : service-group-id
}
```

## 使用例

instanceID=4006 の履歴を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/TaskHistories/4006 HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
```

Accept: application/json  
Accept-Language: ja

レスポンスヘッダー:

HTTP/1.1 200 OK  
Date: Fri, 31 Jul 2015 06:24:06 GMT  
Server Cosminexus HTTP Server is not blacklisted  
Server: Cosminexus HTTP Server  
Access-Control-Expose-Headers: WWW-Authenticate  
WWW-Authenticate: HSS0  
2615a636c3da92888fe355da9ca7d223e6e214\_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=\_V0810  
Access-Control-Allow-Origin: \*  
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS  
Access-Control-Allow-Credentials: true  
Cache-Control: no-cache  
Transfer-Encoding: chunked  
Content-Type: application/json

レスポンスボディ:

```
{
  "instanceID" : 4006,
  "name" : "Execute Remote Command_20150731105831",
  "submitter" : "System",
  "serviceName" : "Execute Remote Command",
  "tags" : "Windows, Linux, Execute Script",
  "scheduleType" : "schedule",
  "scheduledStartTime" : "2015-07-31T11:30:00.000+09:00",
  "startTime" : "2015-07-31T11:30:00.000+09:00",
  "completionTime" : "2015-07-31T11:30:33.000+09:00",
  "archiveTime" : "2015-07-31T15:22:21.000+09:00",
  "taskID" : 3042,
  "submitTime" : "2015-07-31T11:00:06.000+09:00",
  "status" : "completed",
  "description" : "",
  "serviceState" : "release",
  "todo" : true,
  "notes" : "Notes Test",
  "serviceGroupName" : "DefaultServiceGroup",
  "serviceGroupID" : 3
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)

---

## 2.8.4 履歴の削除 (ID 指定)

### 機能

タスク ID を指定して履歴を削除します。

## 実行権限

Admin ロール, Develop ロール, Modify ロール

## API のバージョン

v1

## リクエスト形式

```
DELETE http://host:port/Automation/version/objects/TaskHistories/id
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
204	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	履歴を削除する権限がありません。
412	Precondition failed	サーバが起動していません。
500	Server-side error	サーバ処理エラーが発生しました。

## 使用例

instanceID=5237 のタスクの履歴を削除する場合の使用例を示します。

リクエストヘッダー:

```
DELETE /Automation/v1/objects/TaskHistories/5237 HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 204 No Content
Date: Fri, 07 Aug 2015 11:14:12 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
3b6cddc1eaffe8cd8c2bbcc88ce991e8419472cc_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
```

```
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Content-Length: 0
Content-Type: application/json
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.8.5 履歴に対する操作一覧の取得

### 機能

指定した ID の履歴に対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/TaskHistories/id/actions
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。またはリソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {
    "name" : "delete",
    "href" : "http://host:port/Automation/version/objects/TaskHistories/id",
    "method" : "DELETE",
    "parameters" : []
  } ],
  "count" : 1
}
```

## 使用例

instanceID=5237 の履歴に対して実行できる操作の一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/TaskHistories/5237/actions HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 11:12:20 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
a754baf585ff2447abf34a09fb93ea3b953cfe_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "name" : "delete",
    "href" : "http://192.168.146.132:22015/Automation/v1/objects/TaskHistories/5237",
    "method" : "DELETE",
    "parameters" : [ ]
  } ],
  "count" : 1
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-



## 2.9 プロパティ関連の API 一覧

プロパティ定義またはプロパティ値を管理するための操作を説明します。

### 2.9.1 プロパティ定義一覧の取得

#### 機能

プロパティ定義の一覧を取得します。取得できるのは、入出力種別が、in, out のサービスプロパティ、またはサービス共有プロパティです。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyDefinitions
```

API の実行ユーザーが権限を持つ、すべてのサービスおよびタスクを対象として、プロパティ定義の一覧を取得します。クエリパラメータを指定すると、取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-74 API 「プロパティ定義一覧の取得」 に指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
serviceID	指定した値と等しい
taskID	
serviceTemplateID	

そのほか指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

#### 指定例

クエリパラメータとして、serviceID に 16731 を、taskID に 512 を指定する例を次に示します。

```
?serviceID=16731&taskID=512
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"プロパティ定義機能のリソース (PropertyDefinitions) のメンバー" :
値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのサービスおよびタスクを対象として、プロパティ定義の一覧を取得する場合の使用例を次に示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyDefinitions HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVtO m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 11:23:05 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
12979bce0bbef62fd8aa38443caf71e13bfcc_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 5102,
    "keyName" : "common.targetHost",
    "displayName" : "Host name of execution target server",
    "defaultValue" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "Specifies the host name or IP address of the execution target server.
IPv6 addresses are not supported.",
    "mode" : "in",
    "required" : true,
    "maxLength" : 255,
    "minLength" : 1,
    "pattern" : "^[0-9a-zA-Z¥¥.¥¥-]*$",
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 5106
  }, {
    "instanceID" : 5055,
    "keyName" : "service.errorMessage",
    "displayName" : "Error message",
    "defaultValue" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "Set the error message displayed in the Task Details window in the
Messages area.",
    "mode" : "out",
    "required" : false,
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 5035
  }, {
    "instanceID" : 5098,
    "keyName" : "common.remoteCommand",
    "displayName" : "Command",
    "defaultValue" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "Specify the full path of the command to be executed on the execution
target server. If the path contains a space, enclose the entire path in double quotation
marks.",
    "mode" : "in",
    "required" : true,
    "maxLength" : 256,
    "minLength" : 1,
```

```

    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 5106
  }, {
    "instanceID" : 19,
    "keyName" : "com.hitachi.software.dna.sys.mail.notify",
    "displayName" : "メール通知",
    "defaultValue" : "false",
    "type" : "boolean",
    "visibility" : "config",
    "scope" : "share",
    "description" : "メール通知機能のオン/オフの設定をします。[ビルトインサービス共有プロパティ]",
    "mode" : "in",
    "required" : true,
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 5084,
    "keyName" : "common.remoteCommandParameter",
    "displayName" : "Command parameters",
    "defaultValue" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "Specify the parameters for the command to be executed on the execution target server. If a parameter contains a space, enclose the entire parameter in double quotation marks.",
    "mode" : "in",
    "required" : false,
    "maxLength" : 1024,
    "minLength" : 1,
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 5106
  }, {
    "instanceID" : 11,
    "keyName" : "com.hitachi.software.dna.sys.mail.smtp.server",
    "displayName" : "SMTPサーバのアドレス",
    "defaultValue" : "",
    "type" : "string",
    "visibility" : "config",
    "scope" : "share",
    "description" : "SMTPサーバのアドレスを設定します。IPv4, IPv6, ホスト名で指定可能です。一つのみ指定可能です。「,」（カンマ）で区切りで複数のアドレスを指定することは出来ません。[ビルトインサービス共有プロパティ]",
    "mode" : "in",
    "required" : false,
    "maxLength" : 255,
    "minLength" : 0,
    "readOnly" : false,
    "hidden" : false
  }

```

```

}, {
  "instanceID" : 5095,
  "keyName" : "common.stdoutProperty",
  "displayName" : "Standard output string",
  "defaultValue" : "",
  "type" : "string",
  "visibility" : "exec",
  "scope" : "local",
  "description" : "This property contains the character string output to standard output
by the specified command. ",
  "mode" : "out",
  "required" : false,
  "propertyGroupName" : "reserved.defaultGroup",
  "validationScript" : "",
  "readOnly" : false,
  "hidden" : false,
  "reference" : false,
  "serviceTemplateID" : 5106
}, {
  "instanceID" : 12,
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.port",
  "displayName" : "SMTPサーバのポート番号",
  "defaultValue" : "25",
  "type" : "integer",
  "visibility" : "config",
  "scope" : "share",
  "description" : "SMTPサーバのポート番号を設定します。[ビルトインサービス共有プロパ
ティ]",
  "mode" : "in",
  "required" : true,
  "minValue" : "0",
  "maxValue" : "65535",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 17,
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.userid",
  "displayName" : "SMTPサーバのユーザーID",
  "defaultValue" : "",
  "type" : "string",
  "visibility" : "config",
  "scope" : "share",
  "description" : "SMTPサーバにログインするユーザーのユーザーIDを設定します。[ビルトイン
サービス共有プロパティ]",
  "mode" : "in",
  "required" : false,
  "maxLength" : 255,
  "minLength" : 0,
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 3,
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.password",
  "displayName" : "SMTPサーバのパスワード",
  "type" : "password",
  "visibility" : "config",
  "scope" : "share",
  "description" : "SMTPサーバにログインするユーザーのパスワードを設定します。[ビルトイン

```

```

サービス共有プロパティ]",
  "mode" : "in",
  "required" : false,
  "maxLength" : 1024,
  "minLength" : 0,
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 6,
  "keyName" : "com.hitachi.software.dna.sys.mail.from",
  "displayName" : "通知メールの送信元",
  "defaultValue" : "",
  "type" : "string",
  "visibility" : "config",
  "scope" : "share",
  "description" : "通知メールの送信元を設定します。[ビルトインサービス共有プロパティ]",
  "mode" : "in",
  "required" : false,
  "maxLength" : 255,
  "minLength" : 0,
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 16,
  "keyName" : "com.hitachi.software.dna.sys.mail.to",
  "displayName" : "通知メールの送信先(T0)",
  "defaultValue" : "",
  "type" : "string",
  "visibility" : "config",
  "scope" : "share",
  "description" : "通知メールの送信先(T0)を設定します。「,」（カンマ）で区切るにより複数のメールアドレスを指定できます。[ビルトインサービス共有プロパティ]",
  "mode" : "in",
  "required" : false,
  "maxLength" : 255,
  "minLength" : 0,
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 10,
  "keyName" : "com.hitachi.software.dna.sys.mail.cc",
  "displayName" : "通知メールの送信先(CC)",
  "defaultValue" : "",
  "type" : "string",
  "visibility" : "config",
  "scope" : "share",
  "description" : "通知メールの送信先(CC)を設定します。「,」（カンマ）で区切るにより複数のメールアドレスを指定できます。[ビルトインサービス共有プロパティ]",
  "mode" : "in",
  "required" : false,
  "maxLength" : 255,
  "minLength" : 0,
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 13,
  "keyName" : "com.hitachi.software.dna.sys.mail.bcc",
  "displayName" : "通知メールの送信先(BCC)",

```

```

    "defaultValue" : "",
    "type" : "string",
    "visibility" : "config",
    "scope" : "share",
    "description"
a57
: "通知メールの送信先(BCC)を設定します。「,」(カンマ)で区切ることで複数のメールアドレス
を指定できます。[ビルトインサービス共有プロパティ]",
    "mode" : "in",
    "required" : false,
    "maxLength" : 255,
    "minLength" : 0,
    "readOnly" : false,
    "hidden" : false
}, {
    "instanceID" : 4,
    "keyName" : "com.hitachi.software.dna.sys.jp1.username",
    "displayName" : "JP1ユーザー名",
    "defaultValue" : "jp1admin",
    "type" : "string",
    "visibility" : "config",
    "scope" : "share",
    "description" : "サービスを実行するJP1ユーザー名を設定します。[ビルトインサービス共有プ
ロパティ]",
    "mode" : "in",
    "required" : true,
    "maxLength" : 31,
    "minLength" : 1,
    "pattern" : "^[a-zA-Z0-9!#%&%&-%@_`~]+$",
    "readOnly" : false,
    "hidden" : false
}, {
    "instanceID" : 22,
    "keyName" : "com.hitachi.software.dna.sys.jp1.password",
    "displayName" : "JP1ユーザーのパスワード",
    "type" : "password",
    "visibility" : "config",
    "scope" : "share",
    "description" : "サービスを実行するJP1ユーザーのパスワードを設定します。[ビルトインサー
ビス共有プロパティ]",
    "mode" : "in",
    "required" : true,
    "maxLength" : 32,
    "minLength" : 6,
    "pattern" : "^[a-zA-Z0-9!#%&%&' ¥¥(¥¥)¥¥*¥¥+, ¥¥-¥¥. ¥¥/;<=>¥¥?@¥¥[¥¥]¥¥^_`¥¥{¥¥|¥¥}~]+$",
    "readOnly" : false,
    "hidden" : false
}, {
    "instanceID" : 26,
    "keyName" : "com.hitachi.software.dna.sys.task.log.level",
    "displayName" : "タスクのログ出力レベル",
    "defaultValue" : "10",
    "type" : "list",
    "visibility" : "config",
    "scope" : "share",
    "description" : "タスクのログ出力レベルを設定します。[ビルトインサービス共有プロパ
ティ]",
    "mode" : "in",

```

```
    "required" : true,
    "valueList" : "0,10,20,30,40",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 15,
    "keyName" : "com.hitachi.software.dna.sys.ssh.privatekey.passphrase",
    "displayName" : "秘密鍵のパスフレーズ (SSH公開鍵認証用)",
    "type" : "password",
    "visibility" : "config",
    "scope" : "share",
    "description" : "SSH公開鍵認証で使用する秘密鍵のパスフレーズを指定します。[ビルトインサービス共有プロパティ]",
    "mode" : "in",
    "required" : false,
    "maxLength" : 1024,
    "minLength" : 0,
    "readOnly" : false,
    "hidden" : false
  } ],
  "count" : 18
}
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.9.2 プロパティ定義情報の取得

### 機能

指定したプロパティ定義の情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyDefinitions/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。



ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	リクエストされたリソースまたは操作がありません。または、ユーザーにリソースの読み取り権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "keyName" : "key-name",
  "displayName" : "display-name",
  "defaultValue" : "default-value",
  "type" : "type",
  "visibility" : "visibility",
  "scope" : "scope",
  "description" : "description",
  "mode" : "mode",
  "required" : {true|false},
  "maxLength" : max-length,
  "minLength" : min-length,
  "minValue" : min-value,
  "maxValue" : max-value,
  "pattern" : "pattern",
  "valueList" : "value-list",
  "propertyGroupName" : "property-group-name",
  "validationScript" : "validation-script",
  "readOnly" : {true|false},
  "hidden" : {true|false},
  "reference" : {true|false},
  "serviceTemplateID" : service-template-id,
}
```

## 使用例

instanceID=158 のプロパティ定義の情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyDefinitions/158 HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:37:24 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 3096d91c11fd92d841b3513ed988ba758237cd1_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : 158,
  "keyName" : "remoteHost",
  "displayName" : "Remote Host",
  "defaultValue" : "",
  "description" : "Specify the IP address or host name of the remote host. The remote host
must be in a network environment that can communicate with the server. You cannot specify
more than one remote host.",
  "mode" : "in",
  "required" : true,
  "maxLength" : 255,
  "minLength" : 1,
  "pattern" : "^[0-9a-zA-Z¥¥.¥¥-]*$",
  "propertyGroupName" : "reserved.defaultGroup",
  "validationScript" : "",
  "readOnly" : false,
  "hidden" : false,
  "reference" : false,
  "serviceTemplateID" : 5106
}
```

## 2.9.3 プロパティ定義に対する操作一覧の取得

### 機能

指定したプロパティ定義に対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyDefinitions/id/actions
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	リクエストがほかのリクエストと競合しています。または、リクエストが現在のオブジェクトの状態と不一致です。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"プロパティ定義機能のリソース (PropertyDefinitions) のメンバー" :
  値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

instanceID=158 のプロパティ定義に対して実行できる操作の一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyDefinitions/158/actions HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:38:20 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 78d4d9d37740a76bfe7212277228eb2db759bb10_V0300
```

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ ],
  "count" : 0
}
```

## 2.9.4 プロパティ定義およびプロパティ値一覧の取得

### 機能

プロパティ定義およびプロパティ値の情報の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyInformations
```

API を実行するときは、クエリパラメータを必ず指定して、一覧を取得する対象を絞り込んでください。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-75 API 「プロパティ定義およびプロパティ値一覧の取得」 で設定できるクエリパラメータ

クエリパラメータ	フィルタ条件
serviceID	指定した値と等しい
taskID	
scheduleID	
shared	サービス共有プロパティを取得対象にする

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"プロパティ定義およびプロパティ値機能のリソース (PropertyInformations) のメン
バー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

serviceID=2004 のサービスのプロパティ定義およびプロパティ値の情報の一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyInformations?serviceID=2004 HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 31 Jul 2015 06:27:14 GMT
Server Cosminexus HTTP Server is not blacklisted
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
94728cefd3f4c996534144711565199189dd8_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
```

Content-Type: application/json

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 2010,
    "keyName" : "common.targetHost",
    "displayName" : "Host name of execution target server",
    "defaultValue" : "",
    "value" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "Specifies the host name or IP address of the execution target server.
IPv6 addresses are not supported.",
    "mode" : "in",
    "required" : true,
    "maxLength" : 255,
    "minLength" : 1,
    "pattern" : "^[0-9a-zA-Z¥¥.¥¥-]*$",
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 560,
    "serviceID" : 2004
  }, {
    "instanceID" : 2013,
    "keyName" : "common.remoteCommand",
    "displayName" : "Command",
    "defaultValue" : "",
    "value" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "Specify the full path of the command to be executed on the execution
target server. If the path contains a space, enclose the entire path in double quotation
marks.",
    "mode" : "in",
    "required" : true,
    "maxLength" : 256,
    "minLength" : 1,
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 560,
    "serviceID" : 2004
  }, {
    "instanceID" : 2017,
    "keyName" : "common.remoteCommandParameter",
    "displayName" : "Command parameters",
    "defaultValue" : "",
    "value" : "",
    "type" : "string",
```

```

    "visibility" : "exec",
    "scope" : "local",
    "description" : "Specify the parameters for the command to be executed on the execution
target server. If a parameter contains a space, enclose the entire parameter in double
quotation marks.",
    "mode" : "in",
    "required" : false,
    "maxLength" : 1024,
    "minLength" : 1,
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 560,
    "serviceID" : 2004
  }, {
    "instanceID" : 2016,
    "keyName" : "common.stdoutProperty",
    "displayName" : "Standard output string",
    "defaultValue" : "",
    "value" : "",
    "type" : "string",
    "visibility" : "exec",
    "scope" : "local",
    "description" : "This property contains the character string output to standard output
by the specified command. ",
    "mode" : "out",
    "required" : false,
    "propertyGroupName" : "reserved.defaultGroup",
    "validationScript" : "",
    "readOnly" : false,
    "hidden" : false,
    "reference" : false,
    "serviceTemplateID" : 560,
    "serviceID" : 2004
  } ],
  "count" : 4
}

```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.9.5 プロパティ値一覧の取得

### 機能

次に示す値の一覧を取得します。

- サービス共有プロパティ

- 特定のサービスに関連するプロパティ
- 特定のスケジュールに関連するプロパティ
- 特定のタスクに関連するプロパティ

## 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyValues
```

API の実行ユーザーが権限を持つ、すべてのサービス、スケジュール、タスクを対象としてプロパティ値の一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-76 API 「プロパティ値一覧の取得」 で設定できるクエリパラメータ

クエリパラメータ	フィルタ条件
serviceID	指定した値と等しい
scheduleID	
taskID	

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

### 指定例

クエリパラメータとして、serviceID に 16731 を、taskID に 512 を指定する例を次に示します。

```
?serviceID=16731&taskID=512
```

サービス、スケジュールまたはタスク用にプロパティ値を取得したい場合、serviceID、scheduleID、または taskID のそれぞれに対応するクエリパラメータを指定する必要があります。クエリパラメータが指定されていない場合は、サービス共有プロパティだけがレスポンスとして返されます。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。



ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"プロパティ値機能のリソース (PropertyValues) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのサービス、スケジュール、タスクを対象として、プロパティ値一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyValues HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:40:06 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 56ca4c95167e4ce4aeb51fa73a85b2923d65e28e_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 25,
```

```

    "type" : "boolean",
    "keyName" : "com.hitachi.software.dna.sys.mail.notify",
    "value" : "false",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 24,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.smtp.server",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 18,
    "type" : "integer",
    "keyName" : "com.hitachi.software.dna.sys.mail.smtp.port",
    "value" : "25",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 5,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.smtp.userid",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 14,
    "type" : "password",
    "keyName" : "com.hitachi.software.dna.sys.mail.smtp.password",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 9,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.from",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 20,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.to",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 28,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.cc",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 21,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.bcc",
    "value" : ""
  }

```

```
    "readOnly" : false,
    "hidden" : false
  } ],
  "count" : 9
}
```

## 関連項目

- [2.2.14 リソースのメンバー](#)

## 2.9.6 プロパティ値の一括更新

### 機能

次に示すプロパティ値を一括更新します。

- 特定のタスクに関連するプロパティ値
- 特定のサービスに関連するプロパティ値
- サービス共有プロパティの値
- 複数のサービスのプロパティ値

### 実行権限

Admin ロール, Develop ロール, Modify ロール

### API のバージョン

v1

### リクエスト形式

```
PUT http://host:port/Automation/version/objects/PropertyValues
```

リクエストボディの構造は、次のようになっています。

```
{
  "pagination" : { },
  "data" : [...],
  "count" : X※
}
```

### 注※

X には任意の番号が入ります。

リクエストのスキーマで「data」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-77 「data」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
プロパティ値	PropertyValue	1	更新する PropertyValue リソース

このオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
PropertyValue	instanceID	0~n
	value	
	readOnly	
	hidden	

クエリパラメータを指定しない場合は、サービス共有プロパティを一括更新します。

serviceID のクエリパラメータを指定すると、一括更新する対象を絞り込めます。「readOnly」および「hidden」の属性の更新は、クエリパラメータを指定した場合に可能です。クエリパラメータは次の形式で指定してください。

?クエリパラメータ=値[&クエリパラメータ=値...]

表 2-78 API「プロパティ値の一括更新」で設定できるクエリパラメータ

クエリパラメータ	フィルタ条件
serviceID	指定した値と等しい

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	プロパティ値が不正です。または、編集できないリソースです。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	更新権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"プロパティ値機能のリソース (PropertyValues) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

サービス共有プロパティを一括更新する場合の使用例を示します。

リクエストヘッダー:

```
PUT /Automation/v1/objects/PropertyValues HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtO m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:40:16 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
1aa95d66e62d885b5583da3620bd166fd3a3_Vlo8Y30JBWokHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 25,
    "type" : "boolean",
    "keyName" : "com.hitachi.software.dna.sys.mail.notify",
    "value" : "false",
    "readOnly" : false,
    "hidden" : false
  }, {
    "instanceID" : 24,
    "type" : "string",
    "keyName" : "com.hitachi.software.dna.sys.mail.smtp.server",
    "value" : "",
    "readOnly" : false,
    "hidden" : false
  }
]
```

```

}, {
  "instanceID" : 18,
  "type" : "integer",
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.port",
  "value" : "25",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 5,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.userid",
  "value" : "",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 14,
  "type" : "password",
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.password",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 9,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.from",
  "value" : "",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 20,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.to",
  "value" : "",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 28,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.cc",
  "value" : "",
  "readOnly" : false,
  "hidden" : false
}, {
  "instanceID" : 21,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.bcc",
  "value" : "",
  "readOnly" : false,
  "hidden" : false
} ],
"count" : 9
}

```

## 2.9.7 プロパティ値の取得

### 機能

指定したプロパティ値の情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyValues/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	リクエストされたリソースまたは操作がありません。または、ユーザーにリソースの読み取り権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "type" : type,
  "keyName" : key-name,
  "value" : value,
  "readOnly" : {true|false},
  "hidden" : {true|false},
  "serviceID" : service-id,
  "scheduleID" : schedule-id,
  "taskID" : task-id
}
```

## 使用例

instanceID=7 のプロパティ値の情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyValues/7 HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:40:54 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
c9f651825563b97bf5d72fea6b1b1cde07a3f41_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : 7,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.cc",
  "value" : "",
  "readOnly" : false,
  "hidden" : false
}
```

## 2.9.8 プロパティ値の更新

### 機能

指定した ID のプロパティ値を更新します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール



## API のバージョン

v1

## リクエスト形式

```
PUT http://host:port/Automation/version/objects/PropertyValues/id
```

リクエストのスキーマは、API「プロパティ値の取得」のレスポンスボディと同じ形式です。「PropertyValue」のメンバーとして指定できるオブジェクトを、次の表で説明します。

表 2-79 「PropertyValue」のメンバーとして指定できるオブジェクト

機能	リソース名称	数	説明
プロパティ値	PropertyValue	1	指定した ID の PropertyValue リソース

このオブジェクトに指定必須のプロパティについて、次に説明します。

リソース名称	メンバー名	数
PropertyValue	value	1

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	プロパティ値が不正です。または、編集できないリソースです。
401	Unauthorized	ログイン権限がありません。
403	Forbidden	更新権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "type" : "type",
  "keyName" : "key-name",
  "value" : "value",
}
```

```
"readOnly" : {true|false},
"hidden" : {true|false},
"serviceID" : service-id,
"scheduleID" : schedule-id,
"taskID" : task-id
}
```

## 使用例

instanceID=24 のプロパティの値を更新する場合の使用例を示します。

リクエストヘッダー:

```
PUT /Automation/v1/objects/PropertyValues/24 HTTP/1.1
Host: 10.196.184.238:22015
User-Agent: useragent1
Accept: application/json
Content-Type: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:40:59 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
d37375c943b0fcff62645a210ed9a96d116e153_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : 24,
  "type" : "string",
  "keyName" : "com.hitachi.software.dna.sys.mail.smtp.server",
  "value" : "server",
  "readOnly" : false,
  "hidden" : false
}
```

## 2.9.9 プロパティ値に対する操作一覧の取得

### 機能

指定したプロパティ値に対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyValues/id/actions
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。またはリソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"プロパティ値機能のリソース (PropertyValues) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

### 使用例

instanceID=9 のプロパティ値に対して実行できる操作の一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyValues/9/actions HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 14 Jul 2014 12:41:31 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
1aa95d66e62d885b5583da3620bd166fd3a3_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "name" : "update",
    "href" : "http://10.196.184.238:22015/Automation/v1/objects/PropertyValues/9",
    "method" : "PUT",
    "parameters" : [ ]
  } ],
  "count" : 1
}
```

## 2.9.10 プロパティグループ一覧の取得

### 機能

サービスが保持するプロパティが所属しているプロパティグループの一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/PropertyGroups
```

API を実行するときは、クエリパラメータを必ず指定して、一覧を取得する対象を絞り込んでください。クエリパラメータは複数指定できません。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値
```

表 2-80 API 「プロパティグループ一覧の取得」で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
serviceTemplateID	指定した値と等しい
serviceID	
scheduleID	
taskID	

そのほかに指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"プロパティグループ機能のリソース (PropertyGroups) のメンバー" :
値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

serviceID=3134 のサービスが保持するプロパティが所属している、プロパティグループ一覧を取得する場合の使用例を示します。

#### リクエストヘッダー:

```
GET /Automation/v1/objects/PropertyGroups?serviceID=3134 HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

#### レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 03 Aug 2015 04:06:07 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
b3e9a4ed913c5b5bc941f48bfb1333ced0f1fff6_Vlo8Y30JdDBUB3ljJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

#### レスポンスボディ:

```
{
  "data" : [ {
    "keyName" : "reserved.defaultGroup",
    "displayName" : "reserved.defaultGroup",
    "description" : "",
    "ordinal" : 0,
    "validationScript" : "",
    "display" : "config,submit,taskDetail"
  } ],
  "count" : 1
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.10 サービスグループ関連の API

### 2.10.1 サービスグループ一覧の取得

#### 機能

サービスグループの一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceGroups
```

#### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
400	Bad Request	クエリパラメータが不正です。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

#### レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"サービスグループ機能のリソース (ServiceGroups) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

サービスグループの一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceGroups HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 07:09:41 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
75f7726f932537efbc38f15ea81c31a8797bab1e_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 3,
    "objectID" : "Automation_RG_DEFAULT",
    "name" : "DefaultServiceGroup",
    "description" : "default service group"
  }, {
    "instanceID" : 2,
    "objectID" : "Automation_RG_ALL",
    "name" : "All Service Groups",
    "description" : "default service groups which contains all services"
  } ],
  "count" : 2
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-



## 2.10.2 サービスグループ情報の取得

### 機能

指定したサービスグループの情報を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceGroups/id
```

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "instanceID" : instance-id,
  "objectID" : "object-id"
  "name" : "name",
  "description" : "description"
}
```

### 使用例

instanceID=3 のサービスグループに関する情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceGroups/3 HTTP/1.1
Host: 192.168.146.132:22015
User-Agent: curl/7.36.0
Accept: application/json
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 07:11:12 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
1f2d33f62adb5df5ca712acb2a0a430cb986e_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "instanceID" : 3,
  "objectID" : "Automation_RG_DEFAULT",
  "name" : "DefaultServiceGroup",
  "description" : "default service group"
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
- 

## 2.10.3 サービスグループに対する操作一覧の取得

### 機能

指定したサービスグループに対して実行できる操作の一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/objects/ServiceGroups/id/actions
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
404	Not found	権限が不正です。または、リソースがありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ ],
  "count" : 0
}
```

## 使用例

instanceID=5186 のサービスグループに対して実行できる操作の一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/ServiceGroups/5186/actions HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
Host: 192.168.146.132:22015
Accept: application/json
User-Agent: curl/7.36.0
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 07:16:43 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0
d5802c6c6df5bf91a24f7f372be1af96a241eae_Vlo8Y30JBWoKHUYTEXAMXx5iHgQ=_V0810
Access-Control-Allow-Origin: *
```

```
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ ],
  "count" : 0
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.11 タグ関連の API

### 2.11.1 タググループ一覧の取得

#### 機能

タググループの一覧を取得します。また、タググループに所属するタグ一覧を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/objects/TagGroups
```

#### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

#### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "data" : [ {"タググループ機能のリソース (TagGroups) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

#### 使用例

タググループの一覧を取得する場合の使用例を示します。

#### リクエストヘッダー:

```
GET /Automation/v1/objects/TagGroups HTTP/1.1
Authorization: Basic c3lzdGVtOmlhbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

#### レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 03:37:17 GMT
Server Cosminexus HTTP Server is not blacklisted
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
d72a9887e1aef533d4763b1adf0a391d6cfa6cb_Vlo8Y30JdDBUB3LjJSVParTjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

#### レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 34,
    "name" : "Applications",
    "tags" : "SQL Server, XenDesktop, Oracle Database, Cluster, Exchange"
  }, {
    "instanceID" : 42,
    "name" : "Hypervisors",
    "tags" : "VMware vSphere, Hyper-V"
  }, {
    "instanceID" : 45,
    "name" : "Storage Services",
    "tags" : "Replicate Storage, Add Like Storage, Snapshot, Add New Storage"
  }, {
    "instanceID" : 54,
    "name" : "Uncategorized",
    "tags" : "Basic, Hitachi, Windows, Linux"
  } ],
  "count" : 4
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.11.2 タグ一覧の取得

### 機能

指定したリソースに設定されているタグの一覧を取得します。

### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

### API のバージョン

v1

### リクエスト形式

```
GET http://host:port/Automation/version/objects/Tags
```

API の実行ユーザーが権限を持つ、すべてのタグを対象として一覧を取得します。クエリパラメータを指定すると、一覧を取得する対象を絞り込めます。クエリパラメータは、次の形式で指定してください。

```
?クエリパラメータ=値[&クエリパラメータ=値...]
```

表 2-81 API 「タグ一覧の取得」 で指定できるクエリパラメータの一覧

クエリパラメータ	フィルタ条件
detail	タグの詳細を取得する
resourceType <sup>※1</sup>	指定した値と等しい
ServiceTemplate	
Service	
Task	
TaskHistory	

注※1

resourceType を指定した場合、クエリパラメータと HQL::filter は、resourceType に指定したリソースに対して有効になります。

そのほか指定できるクエリパラメータについては、「[2.2.9 クエリパラメータ](#)」を参照してください。

### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードを、次の表に示します。

ステータスコード	メッセージ	説明
200	OK	成功しました。
401	Unauthorized	ログイン権限がありません。
412	Precondition failed	サーバが利用できません。
500	Server-side error	サーバ処理エラーが発生しました。

## レスポンスのスキーマ

レスポンスボディには、リクエストで指定したクエリパラメータの条件に合致したデータが返却されます。レスポンスボディのスキーマを次に示します。

```
{
  "data" : [ {"タグ機能のリソース (Tags) のメンバー" : 値 ... }, ... ],
  "count" : クエリパラメータの条件に合致したデータの個数 (0~n)
}
```

## 使用例

すべてのタグを対象として一覧を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/objects/Tags HTTP/1.1
Authorization: Basic c3lzdGVt0m1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Fri, 07 Aug 2015 03:38:52 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
d2729dff1c31a47ed713d92612eec93fe7919c8_Vlo8Y30JdDBUB3lJJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "data" : [ {
    "instanceID" : 35,
    "name" : "Replicate Storage",
```



```

    "tagGroupID" : 45
  }, {
    "instanceID" : 36,
    "name" : "SQL Server",
    "tagGroupID" : 34
  }, {
    "instanceID" : 37,
    "name" : "Add Like Storage",
    "tagGroupID" : 45
  }, {
    "instanceID" : 38,
    "name" : "Snapshot",
    "tagGroupID" : 45
  }, {
    "instanceID" : 39,
    "name" : "Add New Storage",
    "tagGroupID" : 45
  }, {
    "instanceID" : 40,
    "name" : "VMware vSphere",
    "tagGroupID" : 42
  }, {
    "instanceID" : 41,
    "name" : "XenDesktop",
    "tagGroupID" : 34
  }, {
    "instanceID" : 43,
    "name" : "Hyper-V",
    "tagGroupID" : 42
  }, {
    "instanceID" : 44,
    "name" : "Oracle Database",
    "tagGroupID" : 34
  }, {
    "instanceID" : 46,
    "name" : "Cluster",
    "tagGroupID" : 34
  }, {
    "instanceID" : 47,
    "name" : "Exchange",
    "tagGroupID" : 34
  }, {
    "instanceID" : 51,
    "name" : "Basic",
    "tagGroupID" : 54
  }, {
    "instanceID" : 52,
    "name" : "Hitachi",
    "tagGroupID" : 54
  }, {
    "instanceID" : 552,
    "name" : "Windows",
    "tagGroupID" : 54
  }, {
    "instanceID" : 559,
    "name" : "Linux",
    "tagGroupID" : 54
  }, {

```

```
    "instanceID" : 564,  
    "name" : "Execute Script",  
    "tagGroupID" : 54  
  }, {  
    "instanceID" : 1004,  
    "name" : "Report Volume Information to Replication Manager",  
    "tagGroupID" : 54  
  } ],  
  "count" : 17  
}
```

---

## 関連項目

- [2.2.14 リソースのメンバー](#)
-

## 2.12 情報管理の API

ユーザー情報を取得したり, JP1/AO および API のバージョン情報を取得したりするための操作を説明します。

### 2.12.1 ユーザー情報の取得

#### 機能

API を実行したユーザーの情報を取得します。

#### 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

#### API のバージョン

v1

#### リクエスト形式

```
GET http://host:port/Automation/version/user
```

#### ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードについては, 「[2.2.17 ステータスコード](#)」のトピックを参照してください。

#### レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "userName" : "user-name",
  "accessPermission" : [ ... ],
  "fullName" : "full-name",
  "description" : "description",
  "email" : "email",
  "resourceGroup" : [ {
    "instanceId" : "instance-id",
    "name" : "resource-group-name",
    "description" : "description",
    "accessPermission" : [ ... ]
  } ]
}
```

## 使用例

実行ユーザーの情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/user HTTP/1.1
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: curl/7.36.0
Host: 10.196.184.182:22015
Accept: application/json
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Thu, 30 Jul 2015 07:17:47 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSSO
31fd21f2412025969969b479f296b5be20b267_Vlo8Y30JdDBUB3LjJSVPaRtjBSA=_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-store, no-transform
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスボディ:

```
{
  "userName" : "System",
  "accessPermission" : [ "User Management" ],
  "fullName" : "",
  "description" : "Built-in account",
  "email" : "",
  "resourceGroup" : [ {
    "instanceID" : "Automation_RG_ALL",
    "name" : "All Service Groups",
    "description" : "default service groups which contains all services",
    "accessPermission" : [ "View", "Execute", "Develop", "Modify", "Admin" ]
  } ]
}
```

### 2.12.2 バージョン情報の取得

#### 機能

JP1/AO と API のバージョンを取得します。

## 実行権限

Admin ロール, Develop ロール, Modify ロール, Submit ロール

## API のバージョン

v1

## リクエスト形式

```
GET http://host:port/Automation/version/configuration/version
```

## ステータスコード

リクエストに対するレスポンスとして返却されるステータスコードについては、「[2.2.17 ステータスコード](#)」のトピックを参照してください。

## レスポンスのスキーマ

リクエストに対するレスポンスボディの構造を次に示します。

```
{
  "productName" : "product-name",
  "productVersion" : "product-version",
  "apiVersion" : "api-version"
}
```

## 使用例

バージョン情報を取得する場合の使用例を示します。

リクエストヘッダー:

```
GET /Automation/v1/configuration/version HTTP/1.1
Host: 10.196.184.238:22015
Accept: application/json
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1
Accept-Language: ja
```

レスポンスヘッダー:

```
HTTP/1.1 200 OK
Date: Mon, 28 Jul 2014 04:34:59 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 4e671d509ad3cd624d83afd9da20f55c1c261193_WIN-JLTV0PQLK2A_V0810
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
```

```
Cache-Control: no-cache  
Transfer-Encoding: chunked  
Content-Type: application/json
```

レスポンスボディ:

```
{  
  "productName" : "JP1/Automatic Operation",  
  "productVersion" : "11-00-00",  
  "apiVersion" : "01.01.00"  
}
```

## 2.13 API の使用例

API でサービスを実行する場合の使用例を示します。まず、実行したいサービスの instanceID を特定します。その後、スケジュール種別を「即時実行」に指定して、サービスを実行する手順を説明します。

1. サービス機能のリソース (Services) の一覧を表示して、実行したいサービスの instanceID を確認します。

```
GET /Automation/v1/objects/Services/ HTTP/1.1
Host:10.197.112.78:22015
User-Agent:sample rest client/1.00.0
Accept:application/json
Accept-Language: ja
Authorization: Basic c3lzdGVtOm1hbmFnZXI=
User-Agent: useragent1

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 11:15:01 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 b8712c86fcd026562182a358ea43bb23b09c62_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

{
  "data" : [ {
    "instanceID" : 2269, ← instanceIDを確認する
    "name" : "リモートコマンド実行",
    "description" : "リモートの実行対象サーバ上のコマンドを実行します。",
    "tags" : "OS_Operations,Basic",
    "serviceTemplateName" : "リモートコマンド実行",
    "createTime" : "2014-08-18T16:53:50.000+0900",
    "modifyTime" : "2014-08-18T16:53:58.000+0900",
    "serviceState" : "release",
    "serviceGroupName" : "DefaultServiceGroup",
    "iconURL" : "http://10.197.112.78:22015/Automation/icon/services/
com.hitachi.software.dna.cts.jp1/remoteCommandExe/01.10.00",
    "vendorName" : "Hitachi,Ltd.",
    "version" : "01.10.00",
    "latest" : true,
    "imageURL" : "http://10.197.112.78:22015/Automation/resources/images/overview/
overview.png",
    "serviceTemplateID" : 2204,
    "serviceGroupID" : 2
  } ],
  "count" : 1
}
```

2. 確認した instanceID を持つリソースに対して、実行できる操作の一覧を取得します。

```
GET /Automation/v1/objects/Services/2269/actions HTTP/1.1
Host:10.197.112.78:22015
```

```
User-Agent:sample rest client/1.00.0
Accept:application/json
Accept-Language: ja
Authorization: HSS0 b8712c86fcd026562182a358ea43bb23b09c62_V0300

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 11:24:41 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 b365e6a2cda2b4d195d55fee1461a6ed0889927_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

{
  "data" : [ {
    "name" : "submit", ← サービスを実行したいため、submitのhref情報を確認する
    "href" : "http://10.197.112.78:22015/Automation/v1/objects/Services/2269/actions/submit/invoke",
    "method" : "POST",
    "parameters" : [ ]
  } ],
  "count" : 1
}
```

### 3. 指定したサービスの実行操作に必要なリクエストボディーの情報を取得します。

```
GET /Automation/v1/objects/Services/2269/actions/submit HTTP/1.1
Host:10.197.112.78:22015
User-Agent:sample rest client/1.00.0
Accept:application/json
Accept-Language: ja
Authorization: HSS0 b8712c86fcd026562182a358ea43bb23b09c62_V0300

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 11:26:00 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 1b9b5891c58315e26cd0cca9aac6d43e572e3db_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json
```

レスポンスをファイル(properties.json)に出力

「properties.json」の内容

```
{
  "name" : "submit",
  "href" : "http://10.197.112.78:22015/Automation/v1/objects/Services/2269/actions/submit/invoke",
  "method" : "POST",
```



```

"parameters" : [ {
  "name" : "リモートコマンド実行_20140818202600",
  "submitter" : "System",
  "scheduleType" : "immediate",
  "description" : "",
  "scheduledStartTime" : "2014-08-18T20:26:00.536+09:00",
  "recurrenceInterval" : "daily",
  "recurrenceDayOfWeek" : "",
  "recurrenceDayOfMonth" : "",
  "recurrenceLastDayOfMonth" : false,
  "recurrenceStartDate" : "2014-08-18",
  "recurrenceTime" : "00:00:00",
  "serviceID" : 2269
}, {
  "instanceID" : 2275,
  "type" : "string",
  "keyName" : "common.targetHost",
  "value" : "",
  "serviceID" : 2269
}, {
  "instanceID" : 2271,
  "type" : "string",
  "keyName" : "common.remoteCommand",
  "value" : "",
  "serviceID" : 2269
}, {
  "instanceID" : 2273,
  "type" : "string",
  "keyName" : "common.remoteCommandParameter",
  "value" : "",
  "serviceID" : 2269
} ]
}

```

4. 取得したテンプレートの情報を必要に応じて編集します。スケジュール種別を「即時実行」に指定する場合は示します。

```

{
  "name" : "submit",
  "href" : "http://10.197.112.78:22015/Automation/v1/objects/Services/2269/actions/submit/invoke",
  "method" : "POST",
  "parameters" : [ {
    "name" : "リモートコマンド実行_20140818202600",
    "submitter" : "System",
    "scheduleType" : "immediate", ←"immediate"を指定する
    "description" : "",
    "scheduledStartTime" : "2014-08-18T20:26:00.536+09:00",
    "recurrenceInterval" : "daily",
    "recurrenceDayOfWeek" : "",
    "recurrenceDayOfMonth" : "",
    "recurrenceLastDayOfMonth" : false,
    "recurrenceStartDate" : "2014-08-18",
    "recurrenceTime" : "00:00:00",
    "serviceID" : 2269
  }, {
    "instanceID" : 2275,
    "type" : "string",

```

```

    "keyName" : "common.targetHost",
    "value" : "", ← 必要に応じてvalueを変更する
    "serviceID" : 2269
  }, {
    "instanceID" : 2271,
    "type" : "string",
    "keyName" : "common.remoteCommand",
    "value" : "hostname", ← 必要に応じてvalueを変更する
    "serviceID" : 2269
  }, {
    "instanceID" : 2273,
    "type" : "string",
    "keyName" : "common.remoteCommandParameter",
    "value" : "", ← 必要に応じてvalueを変更する
    "serviceID" : 2269
  } ]
}

```

5. 編集した情報を使用して、サービスを実行します。

```

POST /Automation/v1/objects/Services/2269/actions/submit/invoke HTTP/1.1
Host:10.197.112.78:22015
User-Agent:sample rest client/1.00.0
Accept:application/json
Accept-Language: ja
Content-Type: application/json
Content-Length: 1087
Authorization: HSS0 b8712c86fcd026562182a358ea43bb23b09c62_V0300

```

properties.jsonの内容をリクエストする

```

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 11:39:03 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 8ba382c1f2e81a65d7a252391b262624c6fa61_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

{
  "instanceID" : "4a9141e4-9566-4e42-af08-0f09926f2a5f",
  "created" : "2014-08-18T20:39:04.242+0900",
  "updated" : "2014-08-18T20:39:04.242+0900",
  "completed" : "2014-08-18T20:39:04.242+0900",
  "state" : "success",
  "affectedResource" : [ "http://10.197.112.78:22015/Automation/v1/objects/Schedules/2285" ←作成されたスケジュールリソースのURL,
    "http://10.197.112.78:22015/Automation/v1/objects/Tasks/2280" ←作成されたタスクリソースのURL ],
  "result" : [ ]
}

```

サービスの実行によって作成された Schedule リソースを取得し、内容を確認します。

```
GET /Automation/v1/objects/Schedules/2285 HTTP/1.1
Host:10.197.112.78:22015
User-Agent:sample rest client/1.00.0
Accept:application/json
Accept-Language: ja
Authorization: HSS0 b8712c86fcd026562182a358ea43bb23b09c62_V0300

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 11:43:00 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 cafcefb87199122267f8ad33772555f9357c8a2_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

{
  "instanceID" : 2285,
  "name" : "リモートコマンド実行_20140818202600",
  "submitter" : "System",
  "scheduleType" : "immediate",
  "createTime" : "2014-08-18T20:39:03.000+0900",
  "modifyTime" : "2014-08-18T20:39:03.000+0900",
  "description" : "",
  "serviceState" : "release",
  "serviceID" : 2269
}
```

6. サービスの実行によって作成された Task リソースを取得し、内容を確認します。

```
GET /Automation/v1/objects/Tasks/2280 HTTP/1.1
Host:10.197.112.78:22015
User-Agent:sample rest client/1.00.0
Accept:application/json
Accept-Language: ja
Authorization: HSS0 b8712c86fcd026562182a358ea43bb23b09c62_V0300

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 11:43:59 GMT
Server: Cosminexus HTTP Server
Access-Control-Expose-Headers: WWW-Authenticate
WWW-Authenticate: HSS0 3713abcd1e99d1481c7b92cc9892a95d1a702d6_V0300
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT, HEAD, OPTIONS
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Transfer-Encoding: chunked
Content-Type: application/json

{
  "instanceID" : 2280,
  "name" : "リモートコマンド実行_20140818202600",
  "status" : "failed",
  "startTime" : "2014-08-18T20:39:04.000+0900",
}
```

```
"completionTime" : "2014-08-18T20:39:13.000+0900",
"submitter" : "System",
"submitTime" : "2014-08-18T20:39:03.000+0900",
"modifyTime" : "2014-08-18T20:39:19.000+0900",
"serviceState" : "release",
"scheduleType" : "immediate",
"description" : "",
"serviceName" : "リモートコマンド実行",
"tags" : "",
"serviceGroupName" : "DefaultServiceGroup",
"serviceTemplateID" : 2204,
"scheduleID" : 2285,
"serviceGroupID" : 2,
"serviceID" : 2269
}
```

# 付録

## 付録 A 参考情報

---

ここでは、JP1/AO を使用する上での参考情報について説明します。

### 付録 A.1 各バージョンの変更内容

#### (1) 11-51 の変更内容

- hcnds64ssltool シンタックスを修正した。
- backupsystem コマンドの注意事項を変更した。
- サポートするメソッドの説明を追加した。
- API の使用例を変更した。

#### (2) 11-50 の変更内容

- JP1/AO は JP1/AJS3 を同梱しなくなったことにより、stopcluster コマンドが不要になったため、関連する記載を削除した。
- hcnds64ssltool コマンドの sigalg オプションで指定できる署名アルゴリズムから「MD5withRSA」を削除した。
- hcnds64dbtrans コマンドで export オプションを指定して実行したときの戻り値「5」を削除した。

#### (3) 11-10 の変更内容

- JP1/AO はタスク処理エンジンとして JP1/AJS3 を使用しなくなったため、関連する記載を削除した。
- setupcluster コマンドおよび restoresystem コマンドについて、jp1user オプションを使用しなくなったため、説明を削除した。
- サービスおよびタスクの実行時に定期実行周期を時間単位で指定可能になったので、説明を追加した。
- JP1/AO は JP1/Base を前提製品としなくなったため、関連する記載を削除した。
- エージェントレス接続先定義情報ファイルの出力項目に、Connection Status と Connected Time を追加した。
- タスクに対して実行できる操作が設定できるようになったため、サービス一覧およびサービスプレート一覧の出力項目に Supported Action Type を追加した。
- タスク一覧の出力項目およびファイルの出力例に Supported Action Type を追加し、タスク一覧の出力例の図を変更した。
- 「サービスプレート一覧の取得」、 「サービス一覧の取得」、 および 「タスク一覧の取得」 で取得できるメンバーに supportedActionType を追加した。

- 「サービステンプレート一覧の取得」, 「サービス一覧の取得」, および 「タスク一覧の取得」 の使用例に, Supported Action Type を追加した。
- 「サービステンプレートの情報取得」, 「サービスの情報取得」, および 「タスクの情報取得」 のレスポンスのスキーマおよび使用例に, Supported Action Type を追加した。
- 「サービスの編集」 および 「タスクの編集」 のレスポンスのスキーマおよび使用例に, Supported Action Type を追加した。
- サービスおよびタスクの実行時に定期実行周期を時間単位で指定可能になったので, 関連するコマンドの定期実行の場合の表にメンバー recurrenceMinutes を追加した。

## (4) 11-01 の変更内容

- submittask コマンドでサービスを定期的に行う際, 日付指定と月末指定をあわせて設定できるようになった。

## (5) 11-00 の変更内容

### (a) 資料番号 (3021-3-088-20) からの変更内容

- 適用 OS に次の OS を追加した。
  - Linux7
  - Oracle Linux 6 (x64)
  - Oracle Linux 7
  - CentOS 6 (x64)
  - CentOS 7
  - SUSE Linux 12
- 適用 OS から次の OS を削除した。
  - Linux 5 (AMD/Intel 64)
  - Linux 5 Advanced Platform (AMD/Intel 64)
- 32 ビット版から 64 ビット版の Windows に移行した。
- Windows 版 JP1/AO および共通コンポーネントのインストール先フォルダを変更した。
- 英語環境または中国語環境で JP1/AO を運用する場合についての説明を追加した。
- JP1/AO と Web ブラウザー間のポート番号を変更した。
- バージョン 11 の JP1/AJS3 と JP1/AO が共存できるようにした。
- Windows で実行できるコマンド名について, 「hcmds」 の部分を 「hcmds64」 に変更した。
- JP1/AO に登録されているエージェントレス接続先定義を削除するための deleteremoteconnection コマンドを追加した。

- JP1/AO に登録されているエージェントレス接続先定義の一覧を出力するための listremoteconnections コマンドを追加した。
- エージェントレス接続先定義を追加または更新するための setremoteconnection コマンドを追加した。
- listservices コマンドで出力できる項目を変更した。
- listtasks コマンドで出力できる項目を変更した。
- 次の API を追加した。

#### サービステンプレート関連の API

- サービステンプレート一覧の取得
- サービステンプレートの情報取得
- サービステンプレートの削除
- サービステンプレートに対する操作一覧の取得
- サービステンプレートのインポートに必要な HTML ファイルの取得
- サービステンプレートのインポート
- サービステンプレートのエクスポートに必要な情報の取得
- サービステンプレートのエクスポート
- サービステンプレートの詳細説明を表示する URL の取得
- サービステンプレートを基にしたサービスの追加に必要な情報の取得
- サービステンプレートを基にしたサービスの追加

#### サービス関連の API

- サービスの編集
- サービスの削除
- サービスのカウンターのリセットに必要な情報の取得
- サービスのカウンターのリセット
- サービスの状態を「リリース」に変更する操作に必要な情報の取得
- サービスの状態を「リリース」に変更
- サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得
- サービスの状態を「メンテナンス」に変更
- サービスの状態を「無効」に変更する操作に必要な情報の取得
- サービスの状態を「無効」に変更
- サービスの詳細説明の URL を取得
- サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得
- サービスが使用するサービステンプレートのバージョン変更



## タスク関連の API

- タスクの編集
- タスクの削除
- タスクの強制停止に必要な情報の取得
- タスクの強制停止
- 応答待ち中のタスクに応答するために必要な情報の取得
- 応答待ち中のタスクに応答
- タスクのアーカイブに必要な情報の取得
- タスクのアーカイブ

## 履歴関連の API

- 履歴一覧の取得
- 履歴の削除（条件指定）
- 履歴の取得
- 履歴の削除（ID 指定）
- 履歴に対する操作一覧の取得

## プロパティ関連の API

- プロパティ定義およびプロパティ値一覧の取得
- プロパティ値の一括更新
- プロパティ値の更新
- プロパティグループ一覧の取得

## サービスグループ関連の API

- サービスグループ一覧の取得
- サービスグループ情報の取得
- サービスグループに対する操作一覧の取得

## タグ関連の API

- タググループ一覧の取得
- タグ一覧の取得
- API の追加に伴い、次の項目を追加・変更した。
  - API で管理できるドメイン名とリソース
  - クエリパラメータ
  - リクエストヘッダー
  - レスポンスヘッダー

- リソースのメンバー
- JP1/AO を操作する API で返却されるメンバー
- 実行できる操作を取得する API で返却されるメンバー
- ステータスコードの説明を追加した。

## (b) 資料番号 (3021-3-366) からの変更内容

- 適用 OS に Linux を追加した。
- Windows 版 JP1/AO および共通コンポーネントのインストール先フォルダを変更した。
- JP1/AO と Web ブラウザー間のポート番号を変更した。
- 32 ビット版から 64 ビット版の Windows に移行した。
- バージョン 11 の JP1/AJS3 と JP1/AO が共存できるようにした。
- Windows で実行できるコマンド名について、「hcmds」の部分を「hcmds64」に変更した。
- JP1/AO に登録されているエージェントレス接続先定義を削除するための deleteremoteconnection コマンドを追加した。
- JP1/AO に登録されているエージェントレス接続先定義の一覧を出力するための listremoteconnections コマンドを追加した。
- エージェントレス接続先定義を追加または更新するための setremoteconnection コマンドを追加した。
- listservices コマンドで出力できる項目を変更した。
- listtasks コマンドで出力できる項目を変更した。
- 「タスク一覧の取得」および「ステップの一覧取得」で取得できるメンバーを、endTime から completionTime に変更した。
- 次の API を追加した。

### サービステンプレート関連の API

- サービステンプレート一覧の取得
- サービステンプレートの情報取得
- サービステンプレートの削除
- サービステンプレートに対する操作一覧の取得
- サービステンプレートのインポートに必要な HTML ファイルの取得
- サービステンプレートのインポート
- サービステンプレートのエクスポートに必要な情報の取得
- サービステンプレートのエクスポート
- サービステンプレートの詳細説明を表示する URL の取得
- サービステンプレートを基にしたサービスの追加に必要な情報の取得

- サービステンプレートを基にしたサービスの追加

#### サービス関連の API

- サービスの編集
- サービスの削除
- サービスのカウンターのリセットに必要な情報の取得
- サービスのカウンターのリセット
- サービスの状態を「リリース」に変更する操作に必要な情報の取得
- サービスの状態を「リリース」に変更
- サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得
- サービスの状態を「メンテナンス」に変更
- サービスの状態を「無効」に変更する操作に必要な情報の取得
- サービスの状態を「無効」に変更
- サービスの詳細説明の URL を取得
- サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得
- サービスが使用するサービステンプレートのバージョン変更

#### タスク関連の API

- タスクの編集
- タスクの削除
- タスクの強制停止に必要な情報の取得
- タスクの強制停止
- 応答待ち中のタスクに応答するために必要な情報の取得
- 応答待ち中のタスクに応答
- タスクのアーカイブに必要な情報の取得
- タスクのアーカイブ

#### 履歴関連の API

- 履歴一覧の取得
- 履歴の削除（条件指定）
- 履歴の取得
- 履歴の削除（ID 指定）
- 履歴に対する操作一覧の取得

#### プロパティ関連の API

- プロパティ定義およびプロパティ値一覧の取得

- プロパティ値の一括更新
- プロパティ値の更新
- プロパティグループ一覧の取得

サービスグループ関連の API

- サービスグループ一覧の取得
- サービスグループ情報の取得
- サービスグループに対する操作一覧の取得

タグ関連の API

- タググループ一覧の取得
- タグ一覧の取得

- API の追加に伴い、次の項目を追加・変更した。

- API で管理できるドメイン名とリソース
- クエリパラメータ
- リクエストヘッダー
- レスポンスヘッダー
- リソースのメンバー
- JP1/AO を操作する API で返却されるメンバー
- 実行できる操作を取得する API で返却されるメンバー

- クラスタ環境の待機系のサーバでも hcnds64get logs コマンドは実行できることの説明を追加した。

- JP1/AO サーバが停止していても hcnds64get logs コマンドは実行できることの説明を追加した。

## (6) 10-54 の変更内容

### (a) 資料番号 (3021-3-088-20) の変更内容

- 「タスク一覧の取得」および「ステップの一覧取得」で取得できるメンバーを、endTime から completionTime に変更した。

## (7) 10-52 の変更内容

### (a) 資料番号 (3021-3-088-10) の変更内容

- 適用 OS に Linux を追加した。
- 部品バージョン管理機能の追加によって次の画面を追加した。
  - [部品バージョン管理] ダイアログボックス ([一括適用] タブ)

- [部品バージョン管理] ダイアログボックス ([個別適用] タブ)
- JP1/AO にログインした際、「予期しないエラー」のメッセージダイアログが表示されたり、画面が正しく表示されなかったりした場合の対処方法についての説明を追加した。
- 操作対象の機器との SSH 接続で使用できる認証方式に、キーボードインタラクティブ認証を追加した。
- 予約部品プロパティ「plugin.publicKeyAuthentication」または「plugin.keyboardInteractiveAuthentication」を選択した場合は、[必須区分] チェックボックスが編集できないことの説明を追加した。
- クラスタ環境の待機系のサーバでも hcmdsget logs コマンドまたは hcmds64get logs コマンドは実行できることの説明を追加した。
- JP1/AO サーバが停止していても hcmdsget logs コマンドまたは hcmds64get logs コマンドは実行できることの説明を追加した。

## (8) 10-50 の変更内容

### (a) 資料番号 (3021-3-088) の変更内容

- 2014 年 9 月以降、マニュアル名称および資料番号を次のとおり変更した。

変更前

*JP1/AO 画面・コマンドリファレンス (3021-3-084)*

変更後

*JP1/AO 画面・コマンド・API リファレンス (3021-3-088)*

- Active Directory と連携してユーザーを管理する機能を追加した。
- https 接続に対応した。
- 操作対象の機器との認証方式に公開鍵認証を追加した。
- stopcluster コマンドを追加した。  
クラスタ環境の JP1/AO のサービスを停止するための準備を実行できるようにした。
- hcmdsldapuser コマンドを追加した。  
Active Directory と連携する場合に、Active Directory の登録情報検索に必要なユーザー情報を編集できるようにした。
- hcmdssltool コマンドを追加した。  
SSL 接続に必要な秘密鍵、CSR、自己署名証明書、および自己署名証明書の内容ファイルを作成できるようにした。
- backupsystem コマンドおよび restoresystem コマンドについて、次のファイルは対象外となる説明を追加した。
  - https 接続用の SSL サーバ証明書ファイル
  - https 接続用の秘密鍵ファイル

- 公開鍵認証用の秘密鍵ファイル
- API をサポートした。

## (b) 資料番号 (3021-3-366) の変更内容

- 2014 年 12 月以降, マニュアル名称および資料番号を次のとおり変更した。

変更前

*Job Management Partner 1/Automatic Operation 画面*・コマンドリファレンス (3021-3-315)

変更後

*Job Management Partner 1/Automatic Operation 画面*・コマンド・API リファレンス (3021-3-366)

- 適用 OS に Windows Server 2012 R2 を追加した。
- タスクモニタ機能およびサービステンプレートのデバッグ機能追加によって, 次の画面を追加した。
  - [タスクログ] ダイアログボックス
  - [デバッグタスク一覧] ビュー
  - [タスクモニタ] ビュー
  - [デバッグ実行] ダイアログボックス
  - [デバッグ] ビュー
  - サービステンプレートデバッグビュー
- Active Directory と連携してユーザーを管理する機能を追加した。
- https 接続に対応した。
- スライドバー変更によって, 次の画面図を変更した。
  - メイン画面
  - [サービス] 画面
  - [タスク] 画面
  - [タスク一覧] ビュー
  - [履歴] ビュー
  - [管理] 画面
  - [エージェントレス接続先定義] ビュー
  - [サービス共有プロパティ] ビュー
  - [ユーザーグループ] ビュー ([ユーザーグループ] タブ)
  - [ユーザーグループ] ビュー ([ユーザー] タブ)
  - [リソースグループ] ビュー
  - [エディタ] 画面

- サービステンプレートエディタ参照ダイアログボックス
  - サービステンプレートエディタ編集ビュー
- 次の画面にタスクログ参照機能を追加した。
- [応答待ちタスク一覧] ダイアログボックス
  - [実行中タスク一覧] ダイアログボックス
  - [正常終了タスク一覧] ダイアログボックス
  - [失敗タスク一覧] ダイアログボックス
  - [タスク一覧] ビュー
  - [タスク詳細] ダイアログボックス
- 次の画面に、タスクのリトライ機能およびタスクの強制停止機能を追加した。
- [応答待ちタスク一覧] ダイアログボックス
  - [実行中タスク一覧] ダイアログボックス
  - [タスク一覧] ビュー
- タスクモニタ機能およびサービステンプレートのデバッグ機能追加によって、次の画面の表示される条件を変更した。
- [サービス実行] ダイアログボックス
  - [タスク詳細] ダイアログボックス
  - [応答入力] ダイアログボックス
  - [部品] ビュー
  - [サービス定義参照] ダイアログボックス
  - [ステップ参照] ダイアログボックス
  - サービステンプレートエディタ編集ビュー
  - [ビルド/リリース結果] ダイアログボックス
- また、次の画面の表示項目の説明を変更した。
- [タスク] 画面
  - [タスク詳細] ダイアログボックス
  - [エディタ] 画面
  - サービステンプレートエディタ編集ビュー
  - [ビルド/リリース結果] ダイアログボックス
- [タスク詳細] ダイアログボックスに、ステップ一覧の表示項目に戻り値を追加した。また、ステップ一覧の状態の表示内容が変更できることを追加した。
- 操作対象の機器との認証方式に公開鍵認証を追加した。

- 次の画面に、root 権限のユーザーで実行するか指定できる機能を追加した。
  - [部品] ビュー
  - [部品作成] ダイアログボックス
  - [部品編集] ダイアログボックス
- リリース部品を削除できるようにした。
- Windows および Linux に加えて、AIX, HP-UX および Solaris で、コマンドまたはスクリプトを実行するコンテンツ部品に対応した。
- stopcluster コマンドを追加した。  
クラスタ環境の JP1/AO のサービスを停止するための準備を実行できるようにした。
- hcmsldapuser コマンドを追加した。  
Active Directory と連携する場合に、Active Directory の登録情報検索に必要なユーザー情報を編集できるようにした。
- hcmdssltool コマンドを追加した。  
SSL 接続に必要な秘密鍵、CSR、自己署名証明書、および自己署名証明書の内容ファイルを作成できるようにした。
- listtasks コマンドでタスク詳細情報の一括出力ができるようにした。また、submittask コマンドで、指定日時実行タスクおよび定期実行タスクの一括再登録ができるようにした。
- hcmdscheckauth コマンドの/user オプションの説明を追加した。また、戻り値 247 を追加した。
- オプションに指定するユーザー名またはパスワードについての注意事項を追加した。
- 次のコマンドに、デバッグサービスおよびデバッグタスクは対象外となることを追加した。
  - listservices コマンド
  - listtasks コマンド
  - stoptask コマンド
  - submittask コマンド
- submittask コマンドで定期実行タスクの登録ができるようにした。
- backupsystem コマンドおよびrestoresystem コマンドについて、次のファイルは対象外となる説明を追加した。
  - https 接続用の SSL サーバ証明書ファイル
  - https 接続用の秘密鍵ファイル
  - 公開鍵認証用の秘密鍵ファイル
- JP1/AO のサービスを開始すると JP1/Base のサービスも開始されるため、JP1/Base のサービスについての記述を削除した。
- クラスタシステムの場合の注意事項を追加した。



- /workpath オプションおよび/file オプションの説明を変更した。また、hcmsdbtrans コマンドの注意事項を追加した。
- restoresystem コマンドに、デバッグサービスおよびデバッグタスクについての説明を追加した。また、リストアしたタスクおよびデバッグタスクについてリトライ操作が選択できないことを追加した。
- API をサポートした。
- 画面に表示される状態を示すアイコンについて、説明を追加した。

## (9) 10-12 の変更内容

### (a) 資料番号 (3021-3-084-50) の変更内容

- 適用 OS に Windows Server 2012 R2 を追加した。
- タスクモニタ機能およびサービステンプレートのデバッグ機能追加によって、次の画面を追加した。
  - [タスクログ] ダイアログボックス
  - [デバッグタスク一覧] ビュー
  - [タスクモニタ] ビュー
  - [デバッグ実行] ダイアログボックス
  - [デバッグ] ビュー
  - サービステンプレートデバッグビュー
- スライドバー変更によって、次の画面図を変更した。
  - メイン画面
  - [サービス] 画面
  - [タスク] 画面
  - [タスク一覧] ビュー
  - [履歴] ビュー
  - [管理] 画面
  - [エージェントレス接続先定義] ビュー
  - [サービス共有プロパティ] ビュー
  - [ユーザーグループ] ビュー ([ユーザーグループ] タブ)
  - [ユーザーグループ] ビュー ([ユーザー] タブ)
  - [リソースグループ] ビュー
  - [エディタ] 画面
  - サービステンプレートエディタ参照ダイアログボックス
  - サービステンプレートエディタ編集ビュー

- 次の画面にタスクログ参照機能を追加した。
  - [応答待ちタスク一覧] ダイアログボックス
  - [実行中タスク一覧] ダイアログボックス
  - [正常終了タスク一覧] ダイアログボックス
  - [失敗タスク一覧] ダイアログボックス
  - [タスク一覧] ビュー
  - [タスク詳細] ダイアログボックス
- 次の画面に、タスクのリトライ機能およびタスクの強制停止機能を追加した。
  - [応答待ちタスク一覧] ダイアログボックス
  - [実行中タスク一覧] ダイアログボックス
  - [タスク一覧] ビュー
- タスクモニタ機能およびサービステンプレートのデバッガ機能追加によって、次の画面の表示される条件を変更した。
  - [サービス実行] ダイアログボックス
  - [タスク詳細] ダイアログボックス
  - [応答入力] ダイアログボックス
  - [部品] ビュー
  - [サービス定義参照] ダイアログボックス
  - [ステップ参照] ダイアログボックス
  - サービステンプレートエディタ編集ビュー
  - [ビルド/リリース結果] ダイアログボックス
 また、次の画面の表示項目の説明を変更した。
  - [タスク] 画面
  - [タスク詳細] ダイアログボックス
  - [エディタ] 画面
  - サービステンプレートエディタ編集ビュー
  - [ビルド/リリース結果] ダイアログボックス
- [タスク詳細] ダイアログボックスに、ステップ一覧の表示項目に戻り値を追加した。また、ステップ一覧の状態の表示内容が変更できることを追加した。
- 次の画面に、root 権限のユーザーで実行するか指定できる機能を追加した。
  - [部品] ビュー
  - [部品作成] ダイアログボックス

- [部品編集] ダイアログボックス
- 次のコマンドに、デバッグサービスおよびデバッグタスクは対象外となることを追加した。
  - listservices コマンド
  - listtasks コマンド
  - stoptask コマンド
  - submittask コマンド
- restoresystem コマンドに、デバッグサービスおよびデバッグタスクについての説明を追加した。また、リストアしたタスクおよびデバッグタスクについてリトライ操作が選択できないことを追加した。
- 画面に表示される状態を示すアイコンについて、説明を追加した。

## (10) 10-11 の変更内容

### (a) 資料番号 (3021-3-084-40) の変更内容

- リリース部品を削除できるようにした。
- Windows および Linux に加えて、AIX, HP-UX および Solaris で、コマンドまたはスクリプトを実行するコンテンツ部品に対応した。
- hcmdscheckauth コマンドの/user オプションの説明を追加した。また、戻り値 247 を追加した。
- オプションに指定するユーザー名またはパスワードについての注意事項を追加した。
- listtasks コマンドでタスク詳細情報の一括出力ができるようにした。また、submittask コマンドで、指定日時実行タスクおよび定期実行タスクの一括再登録ができるようにした。
- submittask コマンドで定期実行タスクの登録ができるようにした。
- JP1/AO のサービスを開始すると JP1/Base のサービスも開始されるため、JP1/Base のサービスについての記述を削除した。
- クラスタシステムの場合の注意事項を追加した。
- /workpath オプションおよび/file オプションの説明を変更した。また、hcmdsdbtrans コマンドの注意事項を追加した。

## (11) 10-10 の変更内容

### (a) 資料番号 (3021-3-084-30) の変更内容

- [エディタ] 画面を使用して、サービステンプレートや部品を開発できる機能を追加した。
- 次の画面の表示項目に、[構成種別] を追加した。
  - [待機中タスク一覧] ダイアログボックス
  - [応答待ちタスク一覧] ダイアログボックス
  - [実行中タスク一覧] ダイアログボックス

- [正常終了タスク一覧] ダイアログボックス
- [失敗タスク一覧] ダイアログボックス
- [サービス] 画面
- [サービス実行] ダイアログボックス
- [サービス追加] ダイアログボックス
- [サービス設定] ダイアログボックス
- [タスク一覧] ビュー
- [タスク詳細] ダイアログボックス
- [履歴] ビュー
- `submittask` コマンド, `stoptask` コマンドの同時実行数を 2 個から 10 個に変更した。
- 次のコマンドを実行できるユーザー権限に, `Develop` ロールを追加した。
  - `deleteservicetemplate` コマンド
  - `importservicetemplate` コマンド
  - `listservices` コマンド
  - `listtasks` コマンド
  - `stoptask` コマンド
  - `submittask` コマンド
- `listservices` コマンドおよび `listtasks` コマンドの出力項目に, 構成種別を追加した。
- `submittask` コマンドの `/property` オプションについて, 指定を省略した場合, [サービス設定] ダイアログボックスで入力した値が設定されることを明記した。
- `submittask` コマンドに, タスクの実行結果を出力してからコマンドを終了する, `/wait` オプションを追加した。
- `submittask` コマンドに, サービスの実行契機を指定する, `/scheduledate` オプションおよび `scheduletime` オプションを追加した。
- 画面の制限値一覧に, [エディタ] 画面の制限値を追加した。

## (b) 資料番号 (3021-3-315-10) の変更内容

- [エディタ] 画面を使用して, サービステンプレートや部品を開発できる機能を追加した。
- Windows Server 2012 の場合の画面操作上の注意事項を追加した。
- 次の画面の表示項目に, [構成種別] を追加した。
  - [待機中タスク一覧] ダイアログボックス
  - [応答待ちタスク一覧] ダイアログボックス
  - [実行中タスク一覧] ダイアログボックス

- [正常終了タスク一覧] ダイアログボックス
- [失敗タスク一覧] ダイアログボックス
- [サービス] 画面
- [サービス実行] ダイアログボックス
- [サービス追加] ダイアログボックス
- [サービス設定] ダイアログボックス
- [タスク一覧] ビュー
- [タスク詳細] ダイアログボックス
- [履歴] ビュー
- [サービス共有プロパティ設定] ダイアログボックス
- [タスク詳細] ダイアログボックスの次の項目を変更した。
  - 「ジョブネット情報」を「ステップ情報」に変更した。
  - 「ジョブネット詳細」を「ステップ詳細」に変更した。
  - ルートジョブネット名を削除した。
  - 「ユニット一覧」を「ステップ一覧」に変更した。
  - 「ユニット名」を「ステップ名」に変更した。
  - 「コメント」を「説明」に変更した。
- 使用できるプロトコルに Telnet を追加した。
- 日本語, 英語, または中国語の部品リソースファイルを選択できるようにした。
- 日本語, 英語, または中国語のサービスリソースファイルを選択できるようにした。
- `submittask` コマンド, `stoptask` コマンドの同時実行数を 2 個から 10 個に変更した。
- 監査ログに出力されるサブジェクト識別情報を変更できる機能を追加した。
- `setupcluster` コマンドは, Windows Server 2012 では使用できないことを追加した。
- 次のコマンドを実行できるユーザー権限に, Develop ロールを追加した。
  - `deleteservicetemplate` コマンド
  - `importservicetemplate` コマンド
  - `listservices` コマンド
  - `listtasks` コマンド
  - `stoptask` コマンド
  - `submittask` コマンド
- `listservices` コマンドおよび `listtasks` コマンドの出力項目に, 構成種別を追加した。

- `submittask` コマンドの `/property` オプションについて、指定を省略した場合、[サービス設定] ダイアログボックスで入力した値が設定されることを明記した。
- `submittask` コマンドに、タスクの実行結果を出力してからコマンドを終了する、`/wait` オプションを追加した。
- `submittask` コマンドに、サービスの実行契機を指定する、`/scheduledate` オプションおよび `scheduletime` オプションを追加した。
- JP1/AO のサービスを開始すると JP1/Base のサービスも開始されるため、JP1/Base のサービスについての記述を削除した。
- 画面の制限値一覧に、[エディタ] 画面の制限値を追加した。
- 画面の制限値一覧を修正した。
- 制限値一覧のうち、機能に関する項目を「機能の制限値一覧」として「JP1/AO 導入・設計ガイド」に移動した。

## (12) 10-02 の変更内容

### (a) 資料番号 (3021-3-084-20) の変更内容

- Windows Server 2012 の場合の画面操作上の注意事項を追加した。
- [タスク詳細] ダイアログボックスの次の項目を変更した。
  - 「ジョブネット情報」を「ステップ情報」に変更した。
  - 「ジョブネット詳細」を「ステップ詳細」に変更した。
  - ルートジョブネット名を削除した。
  - 「ユニット一覧」を「ステップ一覧」に変更した。
  - 「ユニット名」を「ステップ名」に変更した。
  - 「コメント」を「説明」に変更した。
- 使用できるプロトコルに Telnet を追加した。
- 監査ログに出力されるサブジェクト識別情報を変更できる機能を追加した。
- `setupcluster` コマンドは、Windows Server 2012 では使用できないことを追加した。
- 画面の制限値一覧を修正した。

## (13) 10-01 の変更内容

### (a) 資料番号 (3021-3-084-10) の変更内容

- 制限値一覧のうち、機能に関する項目を「機能の制限値一覧」として「JP1/AO 導入・設計ガイド」に移動した。

# 索引

## A

- API 103
- API 一覧 104
- API 共通の仕様 110
- API で管理できるドメイン名とリソース 114
- API の記述形式 152
- API の使用例 343

## B

- backupsystem (JP1/AO システムのバックアップ) 85

## D

- deleteremoteconnection (エージェントレス接続先定義の削除) 36
- deleteservicetemplate (サービステンプレートの削除) 38

## E

- encryptpassword (パスワードファイルの作成) 22

## H

- hcnds64checkauth (外部認証サーバとの接続確認) 24
- hcnds64chgurl (URL 情報の更新) 41
- hcnds64dbrepair (データベースの再作成) 88
- hcnds64dbsrv (データベースの開始・停止) 90
- hcnds64dbtrans (データベースのバックアップ・リストア) 92
- hcnds64fwcancel (Windows ファイアウォールの例外登録) 26
- hcnds64getlogs (ログ情報の採取) 95
- hcnds64intg (認証データの削除・確認) 27
- hcnds64ldapuser (LDAP 検索用ユーザー登録・削除) 28
- hcnds64srv (JP1/AO の開始・停止・状態表示) 42

- hcnds64ssltool (秘密鍵・自己署名証明書の作成) 31

- hcnds64unlockaccount (ユーザーアカウントのロック解除) 46
- HQL 規格の使用 119

## I

- importservicetemplate (サービステンプレートのインポート) 47

## J

- JP1/AO を操作する API で返却されるメンバー 147

## L

- listremoteconnections (エージェントレス接続先定義の一覧出力) 50
- listservices (サービス・サービステンプレートの一覧出力) 53
- listtasks (タスクの一覧出力, タスク詳細情報出力) 58

## R

- restoresystem (JP1/AO システムのリストア) 99

## S

- setremoteconnection (エージェントレス接続先定義の追加・更新) 68
- setupcluster (クラスタ環境の構築) 34
- stoptask (タスクの実行停止) 71
- submittask (サービスの実行, タスクの一括再登録) 74

## う

- 運用関連のコマンド 36

## え

- エラー情報 150



## お

- 応答待ち中のタスクに応答 264
- 応答待ち中のタスクに応答するために必要な情報の取得 262

## く

- クエリパラメータ 115

## こ

- 構築関連のコマンド 22
- コマンド 14
- コマンド一覧 15
- コマンド使用時の注意事項 18
- コマンドの記述形式 21
- コマンドの引数に指定できる文字 20

## さ

- サービス一覧の取得 177
- サービスが使用するサービステンプレートのバージョン変更 213
- サービスが使用するサービステンプレートのバージョン変更に必要な情報の取得 211
- サービス関連の API 177
- サービスグループ一覧の取得 327
- サービスグループ関連の API 327
- サービスグループ情報の取得 329
- サービスグループに対する操作一覧の取得 330
- サービス実行に必要な情報の取得 190
- サービステンプレート一覧の取得 153
- サービステンプレート関連の API 153
- サービステンプレートに対する操作一覧の取 160
- サービステンプレートのインポート 164
- サービステンプレートのインポートに必要な HTML ファイルの取得 162
- サービステンプレートのエクスポート 168
- サービステンプレートのエクスポートに必要な情報の取得 166
- サービステンプレートの削除 158
- サービステンプレートの詳細説明を表示する URL の取得 170

- サービステンプレートの情報取得 156
- サービステンプレートを基にしたサービスの作成 174
- サービステンプレートを基にしたサービスの作成に必要な情報の取得 171
- サービスに対する操作一覧の取得 186
- サービスのカウンターのリセット 197
- サービスのカウンターのリセットに必要な情報の取得 195
- サービスの削除 185
- サービスの実行 192
- サービスの詳細説明の URL を取得 210
- サービスの状態を「無効」に変更 208
- サービスの状態を「無効」に変更する操作に必要な情報の取得 206
- サービスの状態を「メンテナンス」に変更 204
- サービスの状態を「メンテナンス」に変更する操作に必要な情報の取得 202
- サービスの状態を「リリース」に変更 201
- サービスの状態を「リリース」に変更する操作に必要な情報の取得 199
- サービスの情報取得 179
- サービスの編集 182
- サポートするメソッド 114
- 参考情報 350

## し

- 実行できる操作を取得する API で返却されるメンバー 148
- 失敗したステップからリトライ 269
- 失敗した次のステップからリトライ 273
- 情報管理の API 339

## す

- スケジュール一覧の取得 217
- スケジュール関連の API 217
- スケジュールに対する操作一覧の取得 222
- スケジュールの一時停止 230
- スケジュールの一時停止に必要な情報の取得 228
- スケジュールのキャンセル 226
- スケジュールのキャンセルに必要な情報の取得 224



スケジュールの再開 234  
スケジュールの再開に必要な情報の取得 232  
スケジュールの情報取得 220  
ステータスコード 149  
ステップ一覧の取得 279

## せ

セキュリティと認証 110

## た

タグ一覧の取得 335  
タグ関連の API 333  
タググループ一覧の取得 333  
タスク一覧の取得 237  
タスク関連の API 237  
タスク実行停止に必要な情報の取得 249  
タスク操作一覧の取得 246  
タスクのアーカイブ 276  
タスクのアーカイブに必要な情報の取得 275  
タスクの強制停止 255  
タスクの強制停止に必要な情報の取得 253  
タスクの再実行 259  
タスクの再実行に必要な情報の取得 257  
タスクの削除 245  
タスクの実行停止 251  
タスクの情報取得 240  
タスクの編集 242  
タスクのリトライに必要な情報の取得 (失敗したステップからリトライ) 267  
タスクのリトライに必要な情報の取得 (失敗した次のステップからリトライ) 271  
タスクログの取得 281

## つ

通信方式 110

## と

ドメインオブジェクトの形式 121

## に

入出力形式 111

## ね

ネームスペース 111

## は

バージョン情報の取得 340

## ふ

プロパティ値の一括更新 315  
プロパティ値の更新 320  
プロパティ関連の API 一覧 297  
プロパティグループ一覧の取得 324  
プロパティ値一覧の取得 311  
プロパティ値に対する操作一覧の取得 323  
プロパティ値の取得 319  
プロパティ定義一覧の取得 297  
プロパティ定義およびプロパティ値一覧の取得 308  
プロパティ定義情報の取得 304  
プロパティ定義に対する操作一覧の取得 306

## ほ

保守・メンテナンス関連のコマンド 85

## ゆ

ユーザー情報の取得 339

## り

リクエスト形式 112  
リクエストヘッダー 118  
リソースのメンバー 123  
履歴一覧の取得 287  
履歴関連の API 一覧 287  
履歴に対する操作一覧の取得 295  
履歴の削除 (ID 指定) 293  
履歴の削除 (条件指定) 289  
履歴の取得 291

## れ

レスポンス形式 113

レスポンスヘッダー 123

---

 株式会社 日立製作所

〒 100-8280 東京都千代田区丸の内一丁目 6 番 6 号

---