



目次

概要 .....	11
仮想アプライアンスのパッケージアップグレード .....	11
バイナリ (.rpm) パッケージを使用したアップグレード手順 .....	11
ソース ディストリビューションを使用したアップグレード手順 .....	11
12.1_Patch_v16 で導入 .....	11
ユーティリティの消去 .....	11
消去を実行する手順 .....	12
AMQP 認証 .....	12
UCSD 6.7 認定 .....	12
12.1_Patch_v15 で導入 .....	12
一括タスク アクション .....	12
一括タスク アクション フラグの有効化 .....	13
[認証 (Authorization) ] タブ .....	13
[通知 (Notification) ] タブ .....	13
一括タスク アクションを実行する nsAPI .....	13
CloudCenter 4.10.0.9 のサポート .....	14
Apache Solr 8.6.2 での認定 .....	14
12.1_Patch_v14 で導入 .....	14
デュオ Web セキュリティ二要素認証 .....	14
デュオ設定 .....	14
デュオセットアップの PSC 設定 .....	15
管理 (Administration) .....	15
統合モジュール .....	15
デュオ二要素認証 .....	15
デュオ Web セキュリティ二要素認証を登録する手順 .....	15
これが必要な理由 .....	16
Apache Solr 8.4.1 での認定 .....	16
12.1_Patch_v13 で導入 .....	16
Service Link メッセージの再発行 .....	16
12.1_Patch_v12 で導入 .....	17
CCS 5.1x ワークロード マネージャの統合 .....	17
PSC へのワークロード マネージャ接続の追加 .....	17
PSC でワークロード マネージャ接続を追加する手順 .....	17

newscale.properties に追加するプロパティ.....	18
newscale.properties に追加するプロパティ.....	18
サービス アカウント ベースの統合.....	19
<b>Brownfield 展開/VM のインポート.....</b>	<b>19</b>
選択したアプリケーション プロファイルのインポート.....	19
グループのワークロード マネージャサービスの注文.....	20
アプリケーション プロファイル メタデータ タグのインポート.....	20
展開アクション.....	20
同期.....	21
移行.....	21
削除 (Delete) .....	21
一時停止 (Suspend) .....	21
<b>VM アクション.....</b>	<b>22</b>
電源オン/電源オフ/再起動.....	22
ボリュームのアタッチ.....	22
ボリュームの接続解除.....	22
スナップショットの作成.....	22
同期.....	22
インポート.....	22
終了.....	22
<b>VM のコンソール アクセス.....</b>	<b>23</b>
サービス項目の検索.....	23
ワークロード マネージャ VM および展開でのカスタム アクションのサポート.....	23
展開のカスタム アクション.....	24
管理対象仮想マシンのカスタム アクション.....	24
管理対象 VM の WM カスタム アクションを使用するためのディクショナリ、サービス フォーム、およびサービスの設計.....	24
SI ベース ディクショナリの作成.....	24
サービス フォームの作成.....	26
短縮名フィールドに入力する DDR ルールの作成.....	26
管理対象 VM でのカスタム アクションのサービスの作成.....	27
WM でカスタム アクションを実行するサービスのタスクの作成.....	27
サービス タスクの作成.....	28
FTL の作成とプロパティ ファイルの変更.....	30
展開用のカスタム アクションを使用するディクショナリ、サービス フォーム、およびサービスの設計.....	32
SI ベース ディクショナリの作成.....	32
サービス フォームの作成.....	33
カスタム アクションを使用するサービスの作成.....	33
サービス タスクの作成.....	34

FTL の作成とプロパティ ファイルの変更.....	37
記入済みのサービス オーダー フォームをテンプレートとして保存する機能.....	39
Jar のアップグレード.....	40
Patch_v12 で導入された nsAPI.....	40
ローカリゼーション API.....	40
テンプレート API として保存.....	41
ワークロード マネージャをサポートするように変更された CloudCenter API.....	42
12.1_Patch_v11 で導入.....	43
新しいサービス マネージャ UI からタスクのリストをエクスポートする機能.....	43
ブラウザのアップグレード.....	43
認定.....	43
データベースのアップグレード.....	43
Jar のアップグレード.....	43
jQuery ライブラリ.....	43
タスクを取得する API.....	43
12.1_Patch_v10 で導入.....	45
アクション オーケストレーションのポラー .....	45
newscale.properties に追加するプロパティ.....	45
support.properties に追加するプロパティ.....	45
分離された UI のサポート.....	45
UI 分離のための Tomcat クラスタの設定.....	46
上記ノードの LB の設定.....	46
WFクラスタノードでのCROS設定の有効化.....	47
IIS の Cors の有効化.....	48
異なる Web サーバでの PSC UI の設定.....	49
LoginConfig.json の例.....	49
トラブルシューティング.....	50
グリッドとページネーション.....	50
電子メール テンプレートでのサービス項目ネームスペースのサポート.....	53
ServiceLink を使用しない PSC の Host RequestCenter.....	54
Artemis サーバのインストールと設定の手順.....	54
Linux の場合.....	55
Windows の場合.....	55
Requestcenter への Active MQ Artemis サーバの設定.....	56
Weblogic で SL なしの RC の設定.....	56
CCS 5.0.1 AO 統合.....	56
AO PSC 統合ワークフローを設定する手順.....	56
トラブルシューティング.....	57

PSC への AO 接続の追加 .....	57
PSC で AO 接続を追加する手順 .....	57
newscale.properties に追加するプロパティ .....	58
support.properties に追加するプロパティ .....	58
AO 管理統合のその他の機能 : .....	59
Service Catalog のレスポンス UI .....	61
個人プロフィール API .....	62
現在のユーザ情報を取得する API .....	62
最終ログイン情報を取得する API .....	64
使用可能なタイムゾーンをリストする API .....	64
ロケールをリストする API .....	65
カレンダー稼働時間のタイムゾーンの API .....	65
ユーザが特定のカレンダーにエントリするための API .....	67
初期設定 .....	67
権限ページを取得する API .....	69
ユーザの更新 .....	73
カレンダーの更新 .....	74
通常のカレンダーでは使用できない日付を追加する API .....	76
更新 : 基本設定 .....	76
リソース文字列を取得する API .....	77
AO ワークフローをインポートする API .....	78
AO ワークフロー サービスを再インポートする API .....	78
ログ API を表示する API .....	78
ログ反復のリストを取得する .....	78
反復ログを 1 個取得 .....	78
AO ワークフローのリストを取得する API .....	79
12.1_Patch_v9 で導入 .....	80
スタンドアロン Wildfly 向け PSC サービス カタログと Service Manager Web サイト UI 分離 .....	80
概要 .....	80
Tomcat Web サーバの設定 .....	80
別の Web サーバでの PSC UI の設定 .....	80
LoginConfig.json の例 .....	81
分離 PSC UI での認証およびセッション管理 .....	81
PSC DB クレデンシャル .....	82
LDAP 認証 .....	83
次に、IIS の web.config の例を示します。 .....	83
LDAP 外部認証 .....	84

SAML SSO .....	84
認証およびセッション管理中のトラブルシューティング .....	85
パスワードをリセットする API.....	85
システム準拠のサンプル パスワードを生成する API.....	85
パスワードを変更する API.....	85
ログインしてセッション ID を取得する API.....	86
システムからログアウトする API .....	86
グローバル設定用 API.....	86
ブラウザ ベース イベントの DDR ルールを実行する API.....	87
12.1_Patch_v8 で導入.....	87
注：パーティクル フォーム サーバのサポートは廃止され、フォルダは次の場所から削除する必要があります。 .....	87
マテリアル UI コンポーネントを使用したサービス フォーム レンダリング。 .....	87
新しいサービス フォームの場所.....	88
マテリアル フォーム サーバのブラウザ互換性.....	88
マテリアル フォーム サーバのテクノロジー スタック .....	88
カスタム サービス フォームの変更 .....	88
サービス フォームに関するマテリアル UI の制限.....	89
スライダ コンポーネント.....	89
日付および日時コンポーネント .....	89
個人プロフィール UI.....	89
サービス カタログ Web サイト .....	90
新しいサービス フォームのレンダラ設定の変更.....	90
ローカライズされた JavaScript 文字列へのアクセス.....	90
JavaScript レンダラのサポート .....	90
認証トークン API.....	90
12.1_Patch_v7 で導入.....	91
新しいサービス フォーム UI.....	91
コンポーネント.....	91
サービスフォームのタイプ .....	91
新しいサービス フォームの場所.....	91
パーティクル フォーム サーバのブラウザ互換性.....	91
パーティクル フォーム サーバのテクノロジー スタック .....	91
フォーム ルール用の ISF ライブラリ .....	92
フォーム ルール固有の JavaScript を生成する FTL テンプレート .....	92
外部 JavaScript.....	93
サービス フォーム UI レンダリング フロー .....	94

新しいカスタム サービス フォーム用に変更するファイル .....	94
新しいサービス フォーム UI の API 拡張 .....	95
サービス オーダー フォーム コンテキスト API .....	95
申請エントリ API .....	96
追加のクエリ パラメータ .....	98
更新要求のためのドラフト API として保存 .....	98
SIBD デictionaryのサービス項目タイプ .....	99
サービス項目 API .....	100
サービス フォームの Expression API .....	100
サービス項目請求の概算コスト API .....	101
DDR API .....	101
サービス フォーム URL の API 拡張 .....	102
サービス フォームの拡張既存 API .....	102
新しいサービス フォームのレンダラ設定 .....	103
FTL ファイル パスの Newscale プロパティ設定 .....	104
パーティクル フォーム UI のフィールド イベント .....	104
12.1_Patch_v6 で導入 .....	104
新しい注文のサービス オーダー フォームのメタデータを取得する API .....	105
応答ペイロードの説明 .....	106
API 応答の例 .....	107
既存の注文のサービス オーダー フォームのメタデータを取得する API .....	132
応答ペイロードの説明 .....	133
サービス名を使用したサービス詳細 UI へのアクセス .....	134
DDR コールを実行する API .....	134
DDR ルールを実行する API .....	134
DDR 実行の値を含む要求ペイロード。 .....	134
フォームのロード中イベントに関連付けられている DDR ルールの要求と応答の例 .....	134
サービス オーダー フォーム アクション API .....	136
要求 API を送信 .....	136
カートに追加する API .....	137
ドラフトとしてサービスを単独で保存する .....	138
ドラフト内のサービスを含むドラフトとしてサービスを保存する .....	139
サービスをエクスポート/インポートする API .....	139
エクスポート サービス API .....	139
インポート サービス API .....	140

並列スレッドでの独立した Web サービス DDR の実行 .....	140
アクティブ フォームの動作 .....	141
管理設定 .....	141
フォーム ルール javascript FTL の変更 .....	141
UCSD VM プロビジョニング .....	147
サービス/ユーザ コンテキストを取得する API .....	147
要求エントリを更新する API .....	148
サービス項目請求の概算コスト .....	151
12.1_Patch_v4 で導入 .....	154
オーダー フォームをドラフトとして保存する .....	154
確認に戻る .....	154
要請のアーカイブまたはページ .....	156
代理オーダーの自動サジェスト検索 .....	156
検索ファセットの表示順序の設定 .....	156
サービス カタログのカテゴリ内でのサービスのソート .....	157
承認の認可委任割り当て .....	157
サービス フォーム詳細 API のキー フィールド設定 .....	157
要求のアーカイブと消去 .....	158
Service Manager の Web サイト .....	160
CloudCenter 展開の電子メール通知 .....	161
展開および VM でのカスタム アクションのサポート .....	161
拡張 API .....	161
必要なデータのみを返す認証 API .....	161
進行中の認証の要求ヘッダー .....	162
Closed Authorization API の追加のカスタム データ .....	163
オープン認証数を取得する .....	165
ID によるコンシューマ サービス カテゴリの取得 .....	166
未処理の注文の追加情報を取得する .....	168
クローズしたオーダーの追加情報を取得する .....	170
新しい API .....	173
アーカイブまたは消去する要求を取得する .....	173
要求をアーカイブまたは消去する .....	174
確認のために戻る .....	174
指定された要求のキー フィールドを取得する .....	174

要求データを使用して、指定された要求のキー フィールドを取得する .....	175
[Ordered For MySelf] および [Ordered-For-Other] のすべての要求エントリを取得する。 .....	178
優先的に検索ファセットを取得する .....	180
検索ファセットの優先度を更新する .....	180
<b>新しいサービス マネージャ API .....</b>	<b>181</b>
タスクを取得する .....	181
objectType のタスクカウント API を取得します .....	183
キューの詳細を取得する .....	183
カレンダーの詳細を取得する .....	184
カレンダーの詳細をエクスポートする .....	187
複数のタスクの手順を取得する .....	188
複数のタスクの情報を取得する .....	188
複数のタスクのチェックリスト情報を取得する .....	190
提供プロセス .....	191
<b>SM : タスク操作 API .....</b>	<b>193</b>
タスク利用可能なアクションの取得 .....	193
指定タスクのスタッフの詳細を取得する .....	195
複数のタスクのスタッフ情報を取得する .....	195
実行者を更新する .....	197
実行者リストを取得する .....	197
スーパーバイザの更新 .....	199
要求データを取得する .....	199
タスク チェックリスト .....	200
サンプル応答 .....	200
提供プロセス API .....	200
省略されたタスクを表示する .....	203
指定されたタスクの価格詳細を取得する .....	204
指定したタスクの価格設定の詳細を更新する .....	205
要求 ID による要求エントリを取得する .....	205
<b>SM : エフォート API .....</b>	<b>206</b>
ユニット タイプのリストを取得する .....	206
カテゴリのリストを取得する .....	210
タスク ID によるエフォート エントリを取得する .....	210
エフォート エントリを作成する .....	212
エフォート エントリを更新する .....	213
エフォート エントリを削除する .....	213
<b>SI 一括更新のクォータとポリシー .....</b>	<b>214</b>



12.1_Patch_v3 で導入 .....	216
UCSD 6.6 認定 .....	216
CloudCenter の機能拡張 .....	216
サービス アカウント ベースの CloudCenter 統合 .....	216
ユーザ アカウント ベースの統合 .....	217
Brownfield 展開/VM のインポート .....	217
ブラウフィールド展開と VM のユーザの関連付け .....	217
管理対象 VM の同期 .....	218
選択したアプリケーション プロファイルのインポート .....	218
展開の機能拡張 .....	218
展開されたアプリケーションでの完全同期の実行 .....	218
グループ用の CloudCenter サービスの注文 .....	218
アプリケーション プロファイル メタデータ タグのインポート .....	219
CloudCenter アプリケーション展開中のネットワークのオーバーライド .....	219
スケーリング ポリシーのサポート .....	220
展開時の新しいアクション .....	220
VM の機能拡張 .....	221
管理対象 VM の新しいアクション .....	221
管理対象外 VM のインポート .....	222
VM のコンソール アクセス .....	222
拡張サービス項目の検索 .....	222
VM でのカスタム ライフサイクル操作のサポート .....	222
展開でのカスタム アクション操作のサポート .....	225
VM 操作タスクの実行 .....	228
履歴の更新タスク .....	229
同期アプリケーション VM プラグイン タスク .....	231
同期展開プラグイン タスク .....	231
GDPR サポート .....	232
コンプライアンス モジュールの機能 .....	232
PII としてのユーザ データの識別 .....	232
グローバル個人プロフィールの PII 設定 .....	233
ユーザの PII データの表示 .....	233
PII 要求およびサービス項目データのエクスポート .....	233
ユーザ データの匿名化 .....	233
匿名化されたデータの表示 .....	233
ログイン画面でのデータ保護ポリシー URL の設定 .....	233

12.1_Patch_v3 で導入された API.....	233
CloudCenter APIs .....	233
Prime Service Catalog へのアプリケーション プロファイルを更新する .....	233
CloudCenter 接続IDを取得する .....	234
VM 詳細 .....	234
展開へのユーザのマッピング.....	234
管理対象 VM の同期.....	234
各展開のメタデータの表示.....	234
各展開のコストと時間を取得.....	235
PII API.....	236
サービス項目消去 PII データ .....	236
指定したユーザ ID の要求のすべての PII フィールド データを取得する .....	236
指定したユーザ ID のサービス項目のすべての PII フィールド データを取得する.....	237
顧客の PII 要求データをエクスポートする.....	239
顧客の PII サービス項目データをエクスポートする.....	239
指定したユーザの要求を匿名化する.....	239
個人フィールドを PII として設定する .....	240
個人の PII フィールドを取得する .....	240
コンプライアンス モジュールを設定する.....	242
データが匿名化されるユーザのリストを取得する .....	244
ユーザを匿名化する .....	244
匿名化されたデータのリストを取得する.....	245
その他 API .....	251
要求の添付ファイルを取得する .....	251
添付ファイルを削除する.....	252
ドキュメント ID の添付ファイルをダウンロードする .....	252
要求と承認待ちのステータスを取得する.....	252
要求のコメント (ユーザ コメントとシステム コメントの両方) を取得する .....	253
サービス項目一括操作 (作成、更新、削除) .....	254
MySelf および Ordered-For-Other で注文されたすべての要求を取得する .....	256
サービス定義を取得する.....	264
12.1_Patch_v2 で導入.....	266

## 概要

このドキュメントの目的は、最新の 12.1 パッチを通じて出荷された 12.1 リリースのメジャーな拡張機能のすべてを強調することです。

12.1 パッチの zip ファイルには、インストール手順、設定変更、お客様から報告された不具合、およびマイナーな機能拡張に関するすべての情報が記載された readme ドキュメントが含まれています。

したがって、これは 12.1 の readme ドキュメントにすでにある情報に追加情報を足した補足ドキュメントです。

## 仮想アプライアンスのパッケージ アップグレード

パッケージ名 - sudo package

バージョン - 1.9.5.p2

パッケージをアップグレードするには、次の 2 つの方法があります。

- バイナリ (.rpm) パッケージのダウンロードとインストール
- ソース ディストリビューションのダウンロードと構築

### バイナリ (.rpm) パッケージを使用したアップグレード手順

1. プライマリ アプリケーション サーバ ノードに root ユーザとしてログインします。
2. 次のコマンドを実行して、CentOS のバージョンを確認します（予想される値は 7.X です）。

**cat /etc/centos-release.**

3. ご使用の OS バージョンの **sudo 1.9.5p2 RPM** パッケージ ダウンロード URL を <https://www.sudo.ws/download.html#binary> からメモします。
4. 次のコマンドを使用して、アプリケーション サーバ ノードにパッケージをダウンロードします。  
**wget {package\_url}**
5. ダウンロードが完了したら、次のコマンドを使用してパッケージをインストールします。  
**yum localinstall {package\_name}**
6. インストール後、次のコマンドを使用して sudo バージョンを確認します。  
**sudo -V**
7. 仮想アプライアンスのすべてのノードで上記の手順を繰り返します。

### ソース ディストリビューションを使用したアップグレード手順

1. プライマリ アプリケーション サーバ ノードに root ユーザとしてログインします。
2. Web ページ <https://www.sudo.ws/download.html#source> から **sudo 1.9.5p2** ソース ディストリビューションをダウンロードします。
3. リンク (<https://www.sudo.ws/install.html>) からインストール手順を実行します。
4. インストール後、コマンド **sudo -V** を使用して sudo バージョンを確認します。
5. 仮想アプライアンスのすべてのノードで上記の手順を繰り返します。

## 12.1\_Patch\_v16 で導入

### ユーティリティの消去

Prime Service Catalog パッチ v16 には、新しい消去ユーティリティ（サービス項目サブスクリプションの消去）が追加されました。

サービス項目サブスクリプション消去ユーティリティは、サービス項目に関連するデータベースからサブスクリプションデータを削除します。このデータが本番環境で使用されなくなった場合は、これらを削除することによりデータベースのサイズを小さくすることができます。また、この消去ユーティリティを定期的に行うと、全体的なパフォーマンスが改善される可能性があります。

## 消去を実行する手順

- ステップ 1 [サービス項目サブスクリプション エンジン (Service Item Subscription Engine) ] の横にあるオプション ボタンをクリックして消去の種類を選択します。
- ステップ 2 消去するデータをフィルタリングする日付範囲を入力します。[日付範囲 (Date Range) ]には、サービス項目の送信日が入ります。
- ステップ 3 ユーザは、バッチ サイズとサービス項目プロセスの総数を入力できます。  
[バッチサイズ (Batch Size) ]: ユーザは、反復で一度に消去する必要があるサブスクリプションの数を入力する必要があります。
- ステップ 4 [未処理の要求のサブスクリプションを含める (Include Subscriptions of Open Requisition) ]: ユーザは、未処理の要求のサブスクリプションを消去することを選択できます。デフォルトでは、処理済みの要求のみが消去されます。
- ステップ 5 [消去 (Purge) ] をクリックして、消去を開始します。
- ステップ 6 確認ダイアログ ボックスで、[はい (Yes) ] をクリックして続行します。
- ステップ 7 消去が開始されます。[OK] をクリックします。
- ステップ 8 一定時間が経過した後に [消去履歴 (Purge History) ] の [更新 (Refresh) ] をクリックします。消去または分析が完了すると、新しい日時エントリが、[消去履歴 (Purge History) ] ペインのリスト上部に追加されます。
- ステップ 9 [消去履歴 (Purge History) ] ペインで、消去完了日時エントリをクリックし、右側の [ログコンテンツ (Log Content) ] ペインにある消去または分析情報を確認します。

## AMQP 認証

Prime Service Catalog は、AMQP 認証タスクをサポートするようになりました。これは、Service Link を介した外部タスクを使用した認可に似ています。

この AMQP エージェントでは、RabbitMQ アダプタをアウトバウンドとインバウンドの両方として指定して作成する必要があります。このエージェントは、認証ページでワークフロー タイプとして指定する必要があります。

外部 Rabbitmq サーバへのアウトバウンド接続の詳細は、通常どおりエージェントのアウトバウンド プロパティで指定できますが、Rabbitmq サーバのインバウンド接続の詳細は統合モジュールで設定する必要があります。これは、統合モジュールで提供される AMQP 接続の詳細について、PSC は接続用の「psc\_inbound\_queue」という名前の特別なキューを自動的に作成し、インバウンド メッセージを処理するためのキューにメッセージ リスナーを作成/管理するためです。可能なインバウンドアクションは、承認、拒否、OK、確認に戻ることで承認タスクを完了することです。これは、「psc\_inbound\_queue」で関連するアクション メッセージをパブリッシュすることによって実行できます。

### UCSD 6.7 認定

Prime Service Catalog パッチ v16 は、APIC UCSD 6.7.4.3 およびフェンス付き UCSD 6.7.3.0 で検証および認定されています。

注: ユーザは、APIC コンテナまたは VDC の作成用に newscale.properties ファイルで以下に示すプロパティを更新できます。

# APIC VDC を作成する際に、テナント名にプレフィックスとして tenantID を追加/削除するためのフラグ

# デフォルトではフラグは false であり、プレフィックスは VDC の作成時にテナント名に追加されません

ucsd.apic.tenant.addprefix = false

### 12.1\_Patch\_v15 で導入

#### 一括タスク アクション

Prime Service Catalog パッチ v15 では、ユーザがさまざまなオプションを確認できる一括タスク アクションがサポートされるようになりました。(承認/拒否/確認に戻る)。これは非同期 API コールです。一括タスク アクションは、Service Catalog モジュールから実行できます。

## 一括タスク アクションフラグの有効化

一括タスク アクションフラグを有効にするために `newscale.properties` ファイルに追加するプロパティは次のとおりです。

注：デフォルトでは、プロパティは無効になっています。

注：一括タスク アクションを個別に記録するため、ユーザは有効なログファイルの場所を入力できます。

```
#####
#Added to enable bulk approval feature.
#To enable : servicecatalog.bulkapproval.enable=true
#To disable : servicecatalog.bulkapproval.enable=false
#####
servicecatalog.bulkapproval.enable=false
#####
#Enable Bulk action log path here
servicecatalog.bulkapproval.log.path=C:\log-folder
```

### [認証 (Authorization) ] タブ

- 承認、拒否、および確認に戻るアクションは、Service Catalog モジュールの [認証 (Authorization) ] タブと通知セクションで実行できます。
- フラグを有効にすると、[認証 (Authorization) ] タブで要求の選択ドロップダウンとチェックボックスが有効になります。ユーザは要求を選択し、必要な一括操作を実行できます。
- ユーザは「承認中」フィルタを使用して承認タスクを検索できます。

### [通知 (Notification) ] タブ

- 一括承認フラグが `true` に設定されている場合、各エントリのチェックボックスと ([一括承認 (Bulk Approve) ]、[一括拒否 (Bulk Deny) ]、[レビューに一括返信 (Bulk Return to Review) ]) ボタンが表示されます (無効ですがデフォルト)。
- 少なくとも 1 つのチェックボックスをオンにすると、ボタンが有効になり、ユーザは一括タスク アクションを実行できます。

## 一括タスク アクションを実行する nsAPI

- API : `http://localhost:8080/RequestCenter/nsapi/transaction/tasks/action`
- メソッド : POST
- ヘッダー : Content-Type = application/json

ペイロード :

```
{
  "taskActionList" : {
    "actionId" : 1,
    "comment" : "comment here",
    "taskActions" : [
      {
        "taskId" : 1097
      },
      {
        "taskId" : 1100
      },
      -----
      ----
      ---
    ]
  }
}
```

注：「actionId」を追加

1 承認、2 拒否、3 確認に戻る

## CloudCenter 4.10.0.9 のサポート

デフォルトでは、Prime Service Catalog は CloudCenter 4.9.0.1 統合をサポートします。パッチv15 では、4.10.0.9 もサポートするようになりました。

- CloudCenter 4.10.0.9 統合を有効にするには、次のプロパティを **newscale.properties** ファイルと **support.properties** ファイルに追加します。

```
#####
# Flag to enable CloudCenter 4.10.0.9 version support
# By default below flag is false and it supports CloudCenter 4.9.0.1 version
#####
cloudcenter.4.10.0.9.enable=true
#####
```

- ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config\newscale.properties
  - ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config\support.properties
- CloudCenter 統合の接続を作成するために、選択する基本テンプレートは次のとおりです。
    - CC 4.9.0.1 の場合：Cloud Center Base Template Service
    - CC 4.10.0.9 の場合：Cloud Center 4.10.0.9 Base Template Service

注：CloudCenter 4.10.0.9 を使用した Prime Service Catalog パッチ v15 では、一時停止ポリシーとスケーリング ポリシーはサポートされません。

## Apache Solr 8.6.2 での認定

Prime Service Catalog は現在、Microsoft SQL Server データベースに対してのみ Solr の統合をサポートしています。

### 12.1\_Patch\_v14 で導入

## デュオ Web セキュリティ二要素認証

二要素認証では、オンライン アカウントに 2 つ目のセキュリティ レイヤが追加されます。2 つ目の要素（電話やその他のモバイル デバイス）を使用して身元を確認することで、あなた以外の誰かがパスワードが漏えいした場合でもログインできないようにすることができます。



## デュオ設定

PSC は、単一要素ログイン プロセスでのデュオ セキュリティ二要素認証をサポートしています。ユーザーが Prime Service Catalog アプリケーションにログインすると、2 つ目の認証レイヤがデュオ セキュリティ 2 要素認証によって実行されます。

1. デュオ サーバは、Web SDK アプリケーションで設定する必要があります。詳細については、以下のリンクを参照してください。<http://duo.com/docs/duoweb>
2. Web SDK アプリケーションの設定時に、統合キー、秘密キー、および API ホスト名をコピーします。
3. 簡単にログインできるように、記憶したデバイスを有効にします。
4. [ポリシー (Policy) ] [グローバルポリシーの編集 (Edit global policy) ] [記憶されたデバイスを有効にする (Enable 記憶済みデバイス) ]
5. デュオ モバイルの有効化 (<https://duo.com/docs/trusted-endpoints-duo-mobiles>)
6. デュオ セルフサービス ポータルの有効化 (<https://duo.com/docs/self-service-portal>)

## デュオセットアップのPSC設定

### 管理 (Administration)

1. サイト管理者としてログインします。
2. [管理 (Administration) ]、[設定 (Settings) ] の順に移動し、[デュオ ログインの有効化 (Enable DUO Login) ] フラグを有効にします。デフォルトでは、この機能は無効になっています。  
「デュオ ログインの有効化」フラグはオフです。

### 統合モジュール

1. サイト管理者としてログインします。
2. 統合モジュールから新しいデュオ統合接続を追加します。
3. デュオ Web SDK アプリケーションから統合キー、秘密キー、および API ホスト名を入力します。

注：

デュオ統合テスト接続を有効または無効にする `Newscale` プロパティ。

```
#####
# Duo Two-Factor Authentication
# enables Duo integration test connection functionality with true/false values
# default value will be false
duo.integration.enable.test.connection = false
#####
```

4. 注：デュオ二要素認証に接続するには、秘密キーとともにアプリケーション キーを使用します。アプリケーション キーを変更するには、デュオ統合管理接続から [アプリケーション キーの再生成 (Regenerate App key) ] ボタンを使用します。
5. ログアウトしてから、ログインします。
6. PSC ログインに成功すると、でう二要素認証のプロンプトが表示されます。
7. [デュオ二要素認証](#) に示す手順に従って、ログイン プロセスを完了します。
8. PSC はデフォルトのログイン モジュールでリダイレクトされます。

注：

- PSC がインストールされたマシンは、マシンのタイムゾーンに基づく正しい日時形式でセットアップされています。
- PSC デュオ二要素認証は、ローカル DB、LDAP、SAML、および Windows ベースの認証をサポートします。

## デュオ二要素認証

### デュオ Web セキュリティ二要素認証を登録する手順

1. 自分のクレデンシャルでログインします。
2. デバイスを使用して信頼性を確認します。管理者は、SMS、音声コール、ワンタイム パスコード、デュオ モバイル スマートフォン アプリなどを通じて本人確認を行うようにシステムをセットアップできます。
3. 携帯電話以外を使用する場合、固定電話やタブレットを使用することも、管理者にハードウェア トークンを要求することもできます。デュオでは、1 つのアカウントに複数のデバイスを関連付けることができるため、携帯電話と固定電話、固定電話とハードウェアトークンのように、2 つの異なるモバイル デバイスなどを使用することができます。

## これが必要な理由

1. パスワードの侵害はますます容易になっています。パスワードは、盗まれたり、推測されたり、ハッキングされたりする可能性があります。第三者が自分のアカウントにアクセスしていることに気付かないことさえあります。
2. 二要素認証では、2 つ目のセキュリティ レイヤが追加されるため、パスワードが侵害された場合でもアカウントの安全性が確保されます。デュオ Push を使用すると、第三者が自分になりすましてログインしようとする（スマートフォン上で）すぐに警告されます。
3. この 2 つ目の認証要素は、ユーザ名やパスワードとは無関係です。Duo からパスワードの入力を要求されることはありません。

## Apache Solr 8.4.1 での認定

Prime Service Catalog は現在、SQL Server データベースに対してのみ Solr のインテグレーションをサポートしています。

### 12.1\_Patch\_v13 で導入

## Service Link メッセージの再発行

Prime Service Catalog には、ポーラー機能を使用してサービス リンク メッセージの再パブリッシュを自動化する機能が導入されました。ポーラーは、サービス リンク メッセージがパブリッシュされていない要求を取得し、同じメッセージを再パブリッシュするために毎週起動します。

ポーラーのカットオフ日付パラメータは、デフォルトで 7 日間です。これは、ポーラーがカットオフ期間（7 日間）内に作成された要求を検索することを意味します。

サービス リンク メッセージの再パブリッシュの手動設定の詳細については、

[https://www.cisco.com/c/en/us/td/docs/net\\_mgmt/datacenter\\_mgmt/intel\\_auto/service\\_portal/v\\_12\\_1/integration/Guide/CiscoPrimeServiceCatalog-12-1-IntegrationGuide.pdf](https://www.cisco.com/c/en/us/td/docs/net_mgmt/datacenter_mgmt/intel_auto/service_portal/v_12_1/integration/Guide/CiscoPrimeServiceCatalog-12-1-IntegrationGuide.pdf) を参照してください

ポーラーは次のように設定できます

```
#Message Republish Data Poller
#####
#Cron Expression wakes up poller every day at 2 AM
messagerepublish.poller.cron=0 0 2 * * ?
#Cron Expression wakes up health check every 7day 2:30 AM
messagerepublish.poller.health.check.cron=0 30 2 * * ?
#High Availability Health checks threshold, this should be greater than Poller cron time specified in minutes, Poller will be killed if its
running more than 2 hours 7 minutes
messagerepublish.healthCheck.threshold=127
messagerepublish.cutOffDate.days=7
```

support.properties

```
##### Message Republish Data Poller Settings #####
#非クラスタ モード：これは、メッセージ再パブリッシュ スクリプトをポーラーから実行するために有効にする必要があります
#クラスタ モード：これは、クラスタ内のすべてのノード、またはクラスタ内の特定のノードで有効にできます
#-クラスタ内で複数のノードが有効になっている場合でも、クラスタ内の 1 つのノードのみが実行されます（いずれかのノード
が最初に開始する）
messagerepublish.poller.enable = true
手動設定の詳細については、
https://www.cisco.com/c/en/us/td/docs/net\_mgmt/datacenter\_mgmt/intel\_auto/service\_portal/v\_12\_1/integration/Guide/CiscoPrimeServiceCatalog-12-1-IntegrationGuide.pdf を参照してください
```



## 12.1\_Patch\_v12 で導入

## CCS 5.1x ワークロード マネージャの統合

Prime Service Catalog は CloudCenter の次の機能をサポートするようになりました。

## PSC へのワークロード マネージャ接続の追加

## PSC でワークロード マネージャ接続を追加する手順

1. [統合モジュール (Integration Module) ] で [統合 (Integration) ] をクリックします。
2. 統合タイプとして「CCS ワークロード マネージャ」を選択します。

The screenshot shows the 'Configure Integration' page for 'CCS Workload Manager'. The breadcrumb trail is 'Integrations / Integration Types / Configure Integration - CCS Workload Manager'. The form contains the following fields and options:

- Identifier: [Text Input]
- Name \*: [Text Input]
- Host Name or IP Address \*: [Text Input]
- Protocol: [http] (Dropdown)
- Port: [80] (Text Input)
- User Name \*: [Text Input]
- Password \*: [Text Input]
- Tenant Name \*: [Text Input]
- Select Base Template \*: [VM Base Template Service] (Dropdown)
- Enable Poller: [ ] (Checkbox)
- Integration Type: Service Account
- Import Brownfield Deployments: [x] (Checked)
- Import Brownfield VMs: [x] (Checked)

注：ユーザ - 統合管理者ユーザまたは管理者ユーザのみが接続を作成できます。

- ID：固有 ID
- 名前：統合の名前
- ホスト名または IP アドレス：CC501 スイートがホストされている IP
- プロトコル：http/https
- ポート：CC501 スイートのポート
- ユーザ名：CC501 ユーザにはテナント管理者グループとスイート管理者グループが含まれている必要があります。
- テナント名：テナントの名前
- パスワード：システムのパスワード
- ブラウンフィールド導入のインポート：有効にすると、ワークロード マネージャが PSC にインポートされます。
- ブラウンフィールド VM のインポート：有効にすると、ワークロード マネージャ VM が PSC にインポートされます
- ポーラーの有効化 (Enable Poller)：ポーラーは、Newscale のポーラー設定に基づいて同期し、プロパティをサポートします (ユーザが以下のプロパティを有効にする場合は、Newscale とサポートの一部にする必要があります)。

## newscale.properties に追加するプロパティ

API トークンと同期するために、クラウドセンター スイートはポーラー設定 (CCS トークンポーラー) に従って最新のトークンを取得します。たとえば、ポーラーが 10 分ごとに実行されるように設定されている場合、ポーラーは CCS から最新のトークンを取得して PSC を更新します。

注：PSC サーバを起動し稼働させるには、以下の設定が**必須**です。エントリはワークロード マネージャに関連するため、この機能を使用していない場合でも追加する必要があります。CCS トークンが有効になっていない場合、DDR の Web サービスは機能しません。

```
#WM Poller settings
#####
#Cron Expression wakes up poller every 10 minutes of an hour
wm.poller.cron=0 0/10 * * * ?
#Cron Expression wakes up health check for 10th min and 15 mins thereafter of the hour ex : 10,25,40,55 minutes
wm.poller.health.check.cron=0 10/15 * * * ?
#High Availability Health checks threshold, this should be greater than Poller cron time specified in minutes
wm.healthCheck.threshold=91
#####
```

```
#CCS Token Poller settings
#####
#Cron Expression wakes up poller every 10 minutes of an hour
ccstoken.poller.cron=0 0/10 * * * ?
#Cron Expression wakes up health check for 10th min and 15 mins thereafter of the hour ex : 10,25,40,55 minutes
ccstoken.poller.health.check.cron=0 10/15 * * * ?
#高可用性ヘルス チェックのしきい値。これは分で指定されるポーラー式時間よりも大きくなる必要があります
ccstoken.healthCheck.threshold=91
#####
```

## newscale.properties に追加するプロパティ

```
##### WM Poller Settings #####
#Flag to enable/disable the poller
wm.poller.enable=true
```

```
##### CCS Token Poller Settings #####
#Flag to enable/disable the poller
ccstoken.poller.enable=true
```

3. すべてのオブジェクトの [インポート (Import)] をクリックします (アプリケーション プロファイル、アクティベーション プロファイル、展開、VM のインポートには時間がかかります)。
4. アプリケーション プロファイルは PSC のサービスとして変換され、同じものが WM サービスの SD で使用可能になります。(ID名) ワークロード マネージャ サービスにも同じディクショナリ グループと AFC が作成されます。
5. 次のスクリーンショットに示すように、[統合の管理-アプリケーション プロファイルとサービス (Manage Integration - Application Profile and Service)] リストが表示されます。

Details
EDIT

Identifier	RWK
Name	Workload Mgr connection
Host Name or IP Address	172.25.5.53
Protocol	https
Port	64177
User Name	vspancha@cisico.com
Password	
Select Base Template	WM Base Template Service
Enable Poller	true
Tenant Name	PSC
Integration Type	Service Account
Import Brownfield Deployments	true
Import Brownfield VMs	true
STATUS	ACTIVE

---

Discovered
SERVICES OBJECTS

[Application Profiles](#)

RE-IMPORT

NAME	DESCRIPTION	SERVICE TIER ID	PROFILE CATEGORY	SELECT BASE TEMPLATE	SETTINGS
Windows2012R2	Basic App with Windows Server 2012 R2	Windows2012R2-31	NTier	WM Base Template Servic4	⚙
PetClinic	Best wm app profile to locate a pet clinic or veterinarian near you.	PetClinic-33	NTier	WM Base Template Servic4	⚙

## サービス アカウント ベースの統合

ワークロード マネージャ接続は、作成したユーザに関係なく、アプリケーション展開、操作、アクションなどのすべての要求に使用される共通のサービス アカウントに関連付けられます。アプリケーションを展開する権限を持つユーザが、これらのアプリケーションの所有者になります。これらの展開または VM で読み取り/書き込みアクセス権を持つユーザは、アクションを実行できます。

## Brownfield 展開/VM のインポート

統合が作成されると、Prime Service Catalog は、PSC によって検出され、ワークロード マネージャを使用して [サービス項目 (Service Items) ] ページに表示される Cloud Center スイートで作成された展開および VM のインポートをサポートします。

[ブラウンフィールド展開のインポート (Import Brownfield Deployments) ] と [ブラウンフィールド VM のインポート (Import Brownfield VM) ] は、ワークロード マネージャの作成統合で提供される 2 つのオプションです。作成時に、その接続の展開や VM をインポートすることを選択できます。

注：上記の設定は、接続の作成後いつでも編集できます。

## 選択したアプリケーション プロファイルのインポート

ワークロード マネージャ接続用に選択したアプリケーション プロファイルのみを更新および再インポートするための再インポート オプションが導入されました。ワークロード マネージャでアプリケーション プロファイルに変更が加えられた場合、選択したアプリケーション プロファイルの [再インポート (Re-Import) ] をクリックすると、変更が Prime Service Catalog に反映され、サービスも再生成されます。

このオプションは、[統合 (Integrations) ] > [ワークロード マネージャ統合 ( Workload Manager Integration ) ] > [統合の管理 ( Manage Integrations ) ] > [アプリケーション プロファイル ( Application profiles ) ] で使用できます。

## グループのワークロードマネージャサービスの注文

ワークロードマネージャ展開を注文する権限を持つユーザは、ユーザが属するグループのワークロードマネージャサービスを注文できるようになりました。注文したサービス項目に対する読み取り/書き込み権限が選択したグループに提供され、グループのすべてのユーザが展開に対してアクションを実行できます。

サービスフォームにグループリストを表示するオプションが導入されました。[管理 (Administration)] > [設定 (Settings)] にあります。

有効にすると、注文されているいずれかのワークロードマネージャサービスのサービスフォームにグループフィールドが表示されます。このリストには、注文のユーザが属するすべてのグループが含まれます。管理設定でこのオプションを有効にできるのは、サイト管理者だけです。

## アプリケーションプロファイルメタデータタグのインポート

アプリケーションプロファイルにカスタムプロパティのメタデータが設定されている場合、ワークロードマネージャ接続のインポート中に、そのメタデータも **Prime Service Catalog** にインポートされます。次に、メタデータ情報を使用して新しいディクショナリと新しいサービスフォームを作成し、このサービスフォームをワークロードマネージャアプリケーションサービスに関連付けます。

注：システムによって作成されたディクショナリおよびサービスフォームは改ざんしないでください。改ざんされている場合、ワークロードマネージャの展開は失敗します。

注：ワークロードマネージャサービスの注文時に、メタデータフォームに特殊文字を追加できます。ただし、“” 特殊文字の場合は、次のエスケープシーケンスを追加する必要があります。

\ “System is having 32 GB\”

Metadata_Name	Metadata_Value

## 展開アクション

Prime Service Catalog は、アプリケーションと VM の両方の完全同期を実行します。ステータスは、ワークロードマネージャ上のアプリケーションまたは VM のステータスに応じて取得され、VM の詳細 (CPU、RAM、HD、パブリック IP アドレスおよびブ

プライベート IP アドレスなど)、VMのステータスが [サマリーの詳細 (Summary Details) ] ページに表示されます。

Name	422d7217-9c20-dfdf-d979-16aef2b5a42e
Display Name	cqjw-24b889053
External ID	422d7217-9c20-dfdf-d979-16aef2b5a42e
Status	Active
Host Name	cqjw-24b889053
Public IP	172.25.5.10
Private IP	172.25.5.10
Start Time	2019-11-21 19:02:31.331
Cloud Connection ID	76
Cloud VM Type	DEPLOYMENT_VM
Cloud ID	2
Cloud Account ID	2
Cloud Region ID	2
External VM ID	380
External User ID	1
Termination Protection	0
Job ID	720
Virtual CPUs	1
Memory (MB)	2048
Instance Type	1CPU_2048MBMEM
Tags	New_MetaData_AppProfile1:App profile centOS,IsThisCluster:Yes 2VM Cluster,DeploymentScaling:Scaled up to one VM,ProfileName:CentOS,CliQrUserId:1,Name:shNov21_CentOS,associatedTags:PrivateCloud,PrivateCloud:USER_DEFINED_TAG
OS Family	Linux
Parent Job ID	719

## 同期

[展開の同期 (Sync Deployment) ] アクションを使用すると、各展開をワークロード マネージャと同期し、特定の展開の現在のステータスを取得できます。たとえば、ワークロード マネージャからの展開で変更が行われた場合、展開が中断されたり、新しいノードが追加されたりした場合、そのような変更は、指定されたポーリング間隔または手動インポートを待機するのではなく、このアクションを使用して、すぐに **Prime Service Catalog** に同期できます。

## 移行

クラウド センターで展開操作を移行するには、[展開の移行 (Migrate Deployment) ] アクションをトリガーします。

デフォルトでは、展開のコピーがターゲット クラウドに作成され、展開は元のクラウドから削除されます。ただし、展開サービス フォームのオプション **keepExistingDeployment** から [はい (Yes) ] を選択することで、元のクラウドで展開を維持することもできます。

## 削除 (Delete)

削除アクションは、展開と関連する VM を削除します。

## 一時停止 (Suspend)

[一時停止 (Suspend) ] アクションは、展開と関連する VM の電源をオフにします。

## VM アクション

これらの操作は、展開または VM の読み取り/書き込み権限を持つユーザのみが使用できます。（サービス アカウント ベース）

### 電源オン/電源オフ/再起動

VM のステータスに応じて、次の操作を実行できます。

- 電源オン：電源を入れ、VM を起動します。
- 電源オフ：VM を停止し、電源を切ります。
- 再起動：VM を正常にシャットダウンし、再起動します。

### ボリュームのアタッチ

ボリュームの接続オプションを使用すると、アプリケーション全体のすべての階層タイプに複数のボリュームを接続できます。ボリュームごとに、各ルート ディスクのサイズとストレージタイプを指定する必要があります。ボリュームの接続操作は、他または自分用に注文できるサービス呼び出します。

### ボリュームの接続解除

ボリュームの接続解除オプションは、任意のユーザの VM からディスク ボリュームを削除するために使用されます。

### スナップショットの作成

指定の VM のイメージ スナップショットを作成します。

### 同期

この VM オプションは、ポーラー トリガーまたは手動インポートを待機するのではなく、ワークロード マネージャから **Prime Service Catalog** に VM（管理対象外 VM、展開済み VM、およびインポート済み VM）の最新情報を取得します。展開の一部である場合も、展開の詳細を同期します。

### インポート

**Prime Service Catalog** にインポートできる brownfield VM には 2 種類あります。

- 管理対象 VM：ワークロード マネージャによって管理される VM。例：ワークロード マネージャで作成された展開の結果として作成された VM。または、クラウドからワークロード マネージャにインポートされた VM。
- 管理対象外 VM：ワークロード マネージャに接続されているクラウド アカウントに存在する VM。これらは、ワークロード マネージャにインポートされません。ユーザは管理対象外 VM に制限付き権限を持っているため、**Prime Service Catalog** VM からこれらの VM を管理するには、まず VM を **Prime Service Catalog** にインポートする必要があります。[サービス項目 (Service Items)] ページで、管理対象外の VM で許可されたアクションが終了します
- インポート：VM を **Prime Service Catalog** とワークロード マネージャにインポートし、VM ステータスを管理対象 VM に変換します。

### 終了

この操作により、VM が完全に削除されます。

## VM のコンソール アクセス

VM の [サービス項目 (Service Items) ] ページの Web ベース コンソールからワークロード マネージャ VM にアクセスできるようになりました。管理対象 VM で読み取り/書き込み権限を持つユーザは、VM コンソールにアクセスできます。

- Windows VM の場合 :

右側の [RDP] ボタンをクリックすると、新しい Web ページでコンソールが起動します。新しい Web ページでコンソールが起動します。

- Linux VM の場合 :

右側の [SSH] ボタンをクリックすると、新しい Web ページでコンソールが起動します。

## サービス項目の検索

検索条件の表示名を使用して、サービス項目の表示名別にサービス項目をフィルタリングできます。

## ワークロード マネージャ VM および展開でのカスタム アクションのサポート

ワークロード マネージャ VM および展開でカスタム アクションをサポートするには、次の手順に従います。

前提条件 :

- WM アクション ライブラリでのカスタム アクションの作成
- アクション ライブラリに移動し、[新規アクション (New Action) ] をクリックします。
- VM のアクション タイプ コマンド スクリプトを選択し、展開の Web サービス タイプを呼び出します
- カスタム アクションを作成する必要があります。アクション メニューで、コマンド スクリプト アクション タイプが管理対象 VM に表示されます。
- 次のスクリーンショットに示すように、ワークロード マネージャ展開アクション メニューに、Web サービスを呼び出すアクション タイプが表示されます。

## 展開のカスタムアクション

NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCRUED COST	ACTIONS
Test_Nov27 Centos7 (V1.1) Vcenter67 default	Deployed	Private-Cloud	Nov 27, 2019 at 02:46 PM	1 hr	\$0.00/hr \$0.00/mo		\$0.07	-Actions- Suspend Terminate Terminate A... Upgrade Promote Migrate Share VM A... Custom Actions iw ping_google startslagent stopslagent
FinalMigrate Centos7 (V1.1) Vcenter67 default	Terminated	Private-Cloud	Nov 25, 2019 at 04:59 PM	12 hrs 15 mins	\$0.00/hr \$0.00/mo		\$0.83	
Nov15Brownfield Centos7 (V1.1) Vcenter67 default	Stopping	Private-Cloud	Nov 15, 2019 at 02:25 PM	Not Available	\$0.00/hr \$0.00/mo		\$0	
WM_15NovTermntTest Centos7 (V1.1) Vcenter67 default	Stopping	Private-Cloud	Nov 15, 2019 at 12:32 PM	Not Available	\$0.00/hr \$0.00/mo		\$0	
WM_15NovTestSup Centos7 (V1.1) Vcenter67 default	Stopping	Private-Cloud	Nov 15, 2019 at 12:31 PM	Not Available	\$0.00/hr \$0.00/mo		\$0	
WM_Dep15Nov-1 Centos7 (V1.1)	Stopping	Private-Cloud	Nov 15, 2019 at 11:57 AM	Not Available	\$0.00/hr \$0.00/mo		\$0	

## 管理対象仮想マシンのカスタムアクション



## 管理対象VMのWMカスタムアクションを使用するためのディクショナリ、サービスフォーム、およびサービスの設計

## SIベースディクショナリの作成

1. WMアプリケーション仮想マシンを選択して、サービス項目ベースのディクショナリを作成します。

1. [サービスデザイナー (Service Designer)] > [ディクショナリ (Dictionaries)] を選択します。
2. [新規 (New)] > [新しいディクショナリ (New Dictionary)] を選択して、[新しいディクショナリ (New Dictionary)] ページを表示します。
3. [サービス項目 (Service Item)] フィールドの [新しい内部ディクショナリの追加 (Add New Internal Dictionary)] セクションで、「WMアプリケーション仮想マシン」と入力し、サービス項目を選択します。
4. ディクショナリ名やグループ名などの詳細を入力します。
5. [ディクショナリ属性 (Dictionary Attributes)] セクションで、[フィールドの追加 (Add Field)] をクリックして、ユーザー定義フィールド ShortName および ExecutionID を追加します。



6. カスタム アクションに必要なディクショナリ内のすべてのフィールド（カスタム FTL で使用される任意のフィールド）を選択します。注：ExecutionID フィールドがディクショナリに追加されていることを確認します。
7. ディクショナリを保存します。

Use	Name	Type	Maximum	Decimals	Multivalue	Show in Grid	Encrypt	PII
<input checked="" type="checkbox"/>	Name	ServiceIdentifier	128	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	DisplayName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ExternalID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Status	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	HostName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	PublicIp	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	PrivateIp	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	StartTime	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	EndTime	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	CloudConnectionID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ClientID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	CloudVMType	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CloudID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CloudAccountID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CloudRegionID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ExternalVMID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ExternalUserID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ProtectTermination	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Actions	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CPU	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	JobID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RAM	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Storage	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	InstanceType	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Tags	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ScalingPolicy	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AgmgPolicy	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	SecurityProfile	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	NumberCPUS	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Network	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OSFamily	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ContainerID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RequestorID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RequestorEntryID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OrganizationalUnitID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AccountID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AssignedDate	Date	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	SubmittedDate	Date	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	EstimatedPrice	ServiceItemPrice	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ErrorCode	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ErrorMessage	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ShortName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ExecutionID	Number	25	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## サービス フォームの作成

2. アクティブ フォーム コンポーネントを作成し、上記で作成したディクショナリをフォームに追加します。

1. **[サービス デザイナー ( Service Designer ) ] > [アクティブ フォーム コンポーネント ( Active Form Component ) ]** を選択します。
2. **[新規 ( New ) ] > [アクティブフォームコンポーネント ( Active Form Component ) ]** を選択します。
3. 新しいフォームの名前と簡単な説明を入力します。
4. フォーム グループ フィールドをクリックし、フォームに関連付けるグループを 1 つ選択します。
5. **[フォームの保存 ( Save Form ) ]** をクリックします。
6. [フォーム コンテンツ (FormContent) ] タブを選択し、**[ディクショナリの追加 ( Add Dictionaries ) ]** をクリックします。

[ディクショナリの追加 (Add Dictionaries) ] ダイアログ ボックスで、上記のセクション (SI ベース ディクショナリの作成) で作成したディクショナリを検索し、選択します。それから **[追加 ( Add ) ]** をクリックします。

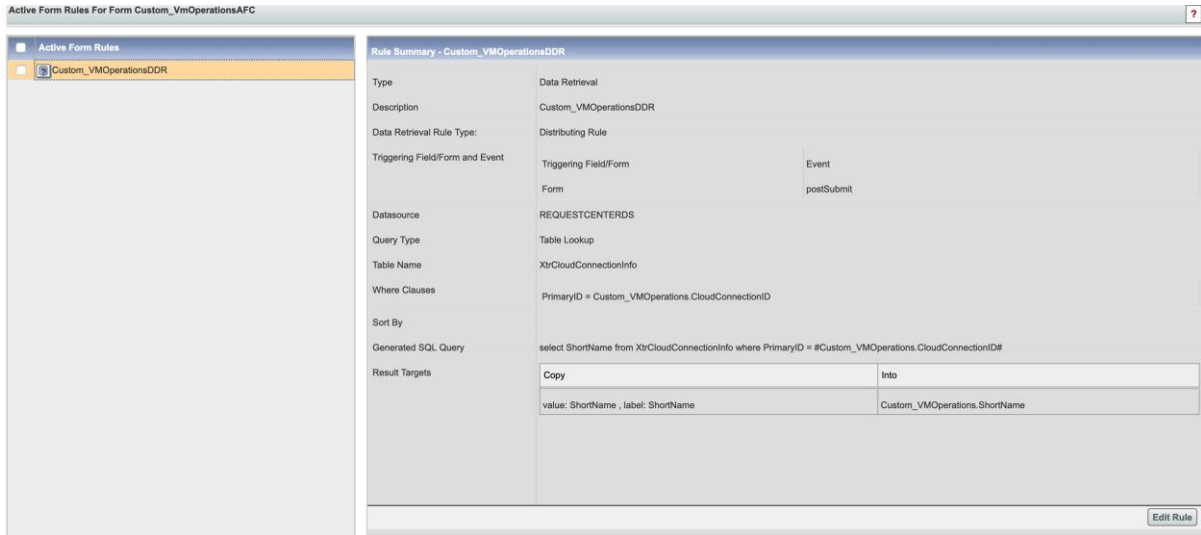
**注：** フォームを保存する前に、ワークロード マネージャに属するVMを選択し、すべての必須ディクショナリ フィールドに値が入力されていることを確認してください。

7. **[フォームの保存 ( Save Form ) ]** をクリックします。

## 短縮名フィールドに入力する DDR ルールの作成

3. データ取得ルールを作成します。[短縮名 ( Short Name ) ] フィールドに入力できるもの

1. 上記で作成したアクティブ フォーム コンポーネント (サービス フォームの作成) の **[アクティブ フォーム ルール ( Active Form Rules ) ]** タブに移動します。
2. **[新しいルール ( New Rule ) ] > [新しいデータ取得ルール ( New Data Retrieval Rule ) ]** の順に選択します。
3. [データ取得ルール (Data Retrieval Rule) ] ウィザードの最初のページで、ルールの一意の名前と説明を入力し、[ルール タイプ (Rule Type) ] と [クエリーのタイプ (Query Type) ] を指定します。  
[クエリーのタイプ (Query Type) ] を指定します。
4. 次の図のようにウィザードでその他詳細を指定し、**[ルールの保存 ( Save Rule ) ]** で短縮名を入力します。



## 管理対象 VM でのカスタムアクションのサービスの作成

4. カスタム サービス (CustomOperationService など) を作成し、サービスにフォームを追加します。
  1. **[サービス デザイナー ( Service Designer ) ] > [新規 ( New ) ] > [新規サービス ( New Service ) ]** を選択します。
  2. 提供されたフィールドに詳細を入力します。
  3. **[新しいサービスの追加 (Add New Service)]** をクリックします。
  4. サービスを追加した後、**[全般 ( General ) ]** タブに情報を入力することで設定を開始できます。
  5. **[保存 ( Save ) ]** をクリックします。
  6. 作成したサービスの **[フォーム ( Form ) ]** タブをクリックします。
  7. ウィンドウの左下の **[フォームの追加 ( Add Forms ) ]** をクリックします。[フォームの追加 (Add Form) ] ポップアップウィンドウが表示されます。
  8. [検索 (Search) ] フィールドに、ステップ (サービス フォームの作成) で作成したフォーム名を入力します。
  9. フォームを確認し、**[追加 ( Add ) ]** をクリックします。

## WM でカスタムアクションを実行するサービスのタスクの作成

5. サービスの **[プラン ( Plan ) ]** で下に指定されている 2 つのタスクを作成します。

VM 操作タスクの実行と操作完了タスクは必須です。[ステータスの更新 (Update Status) ] タスクはオプションです。

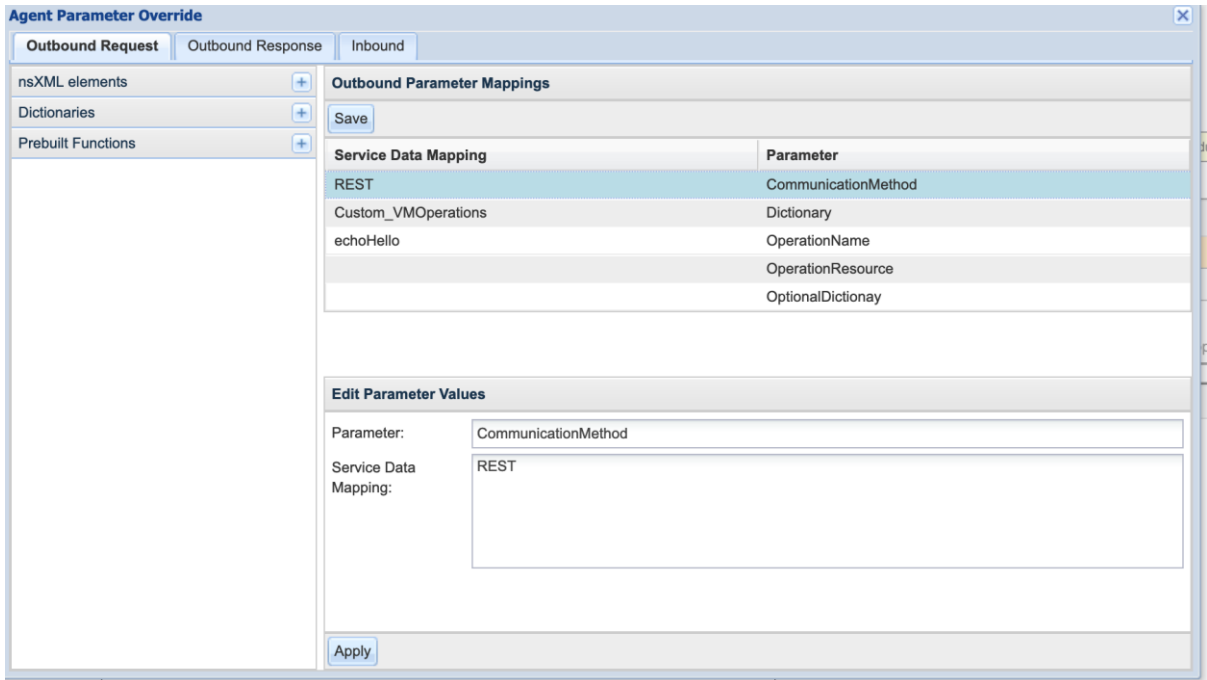
The screenshot displays the configuration page for a service task. At the top, there are tabs for 'Tasks', 'Escalations', and 'Graphical Designer'. Below these are various configuration options such as 'Project Manager', 'Subject for plan monitoring task', and 'Working hours per day'. A table in the middle lists tasks with columns for 'Task', 'By', 'This', 'Subtasks', and 'Subtotal'. The bottom section contains a 'General' tab with fields for 'Workflow Type', 'Task name', 'Subtasks execute', 'Duration', and 'Condition', along with 'Validate...' buttons.

### サービスタスクの作成

- [VM 操作タスクの実行 (Perform VM Operation Task) ] : このタスクのワークフロータイプは WMAgent です (ドロップダウンから選択)。エージェントパラメータを次のように設定します。

パラメータ	サービスデータマッピング
通信方法	REST
Dictionary	ステップ (SI ベース デictionary の作成) で作成した dictionary の名前を入力します。
OperationName	カスタム動作名を入力します。ワークロード マネージャのアクション名と一致する <Cus- tomOperationName> を使用することをお勧めします。 たとえば、ワークロード マネージャで指定されたアクション名が <b>echohello</b> の場合、カスタム操作名は <b>echohello</b> になります。

以下は、エージェントパラメータのスクリーンショットです。



- 操作完了タスク

このタスク ワークフロー タイプは、同期タスク（Javaプラグイン）（ドロップダウンから選択）になり、次のようにプラグイン タスク パラメータを設定します。

Sync Task Plugin Class- com.celosis.event.SyncTaskWMPlugin

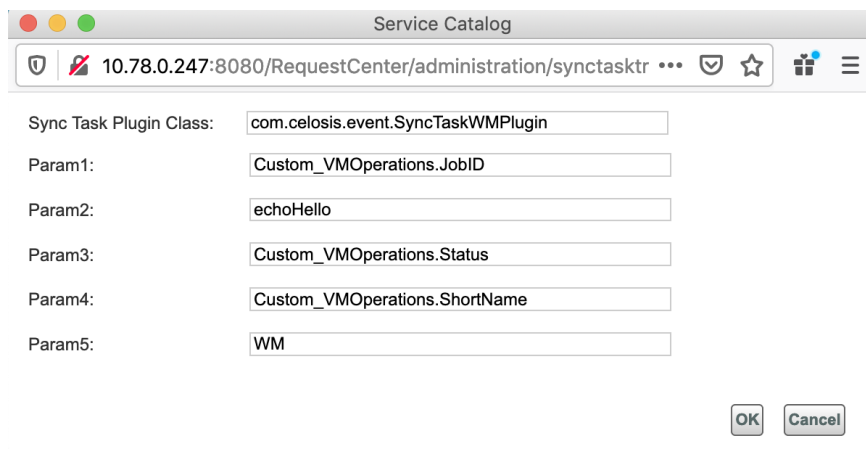
Param1 : <Dictionary Name (Created in the Create SI based Dictionary).JobID

Param2 : カスタム アクション名。ワークロード マネージャのアクション名と一致するものを使用することを推奨します。

Param3 : <Dictionary Name (Created in the Create SI based Dictionary).Status

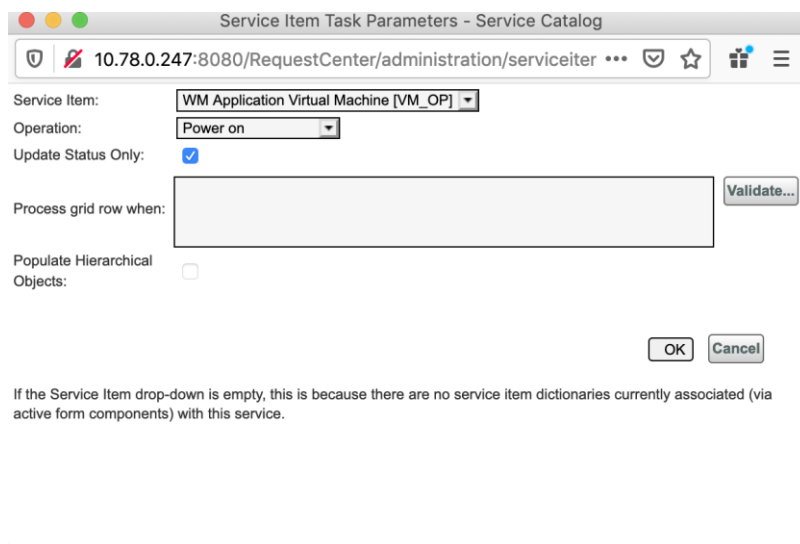
Param4 : <Dictionary Name (Created in the Create SI based Dictionary).ShortName

Param5 : WM



サービスを保存します

注：ユーザがライフサイクル操作（電源オン/電源オフ/再起動など）のカスタムアクションを作成しようとしており、ユーザが VM の中間ステータスを更新する必要がある場合は、次のようにステータス更新タスクを最初のタスクとして追加できます



上記のサービスをサービス項目にマッピングします（このサービスを vm のアクションとして作成）。

1. [サービス項目マネージャ ( Service Item Manager ) ] > [サービス項目の設計 ( Design Service Items ) ] > [CCS ワークロード マネージャ ( CCS Workload Manager ) ] > [WM アプリケーション仮想マシン ( WM Application Virtual Machines ) ] > [関連サービス ( Associated Services ) ] でカスタム サービスを追加します。
2. [サービスを追加 ( Add Service ) ] をクリックします ([サービスを作成 ( Create a Service ) ] セクションで作成されたサービスを検索し、サービスを選択して保存します)。これにより、VM の歯車アイコンにサービスが表示されます。

## FTL の作成とプロパティファイルの変更

カスタムアクション FTL ファイルの例は次のとおりです（Echo Hello コマンドカスタムアクションの例）。

注：FTL ファイルの場合、ディクショナリ (SI ベースディクショナリの作成セクション) を更新し、ワークロードマネージャで作成されたカスタムアクションのアクション ID も更新する必要があります（カスタムアクション WM API を使用して取得できます）。

FTL ファイルの準備ができれば、次の場所に配置する必要があります。

- -wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config\cloud
- -wildfly-10.1.0.Final\ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config\cloud

```
<!--
* FTL will support Headers,Post parameters, url (get) Parameters and payload.
*
* =====
* Header ***** Header_propertyname = Propertyvalue
* Post ***** Post_propertyname = Propertyvalue
* BaseUrl ***** give you the url and fallowd by query parameters
* Payload ***** Payload = payload content
* Header_AuthToken ***** auth token / cookie used for authentication
* =====
*
* Header_Accept-Encoding = UTF-8
* Header_Content-Type = application / xml
```

\* These parameters will apply to the payload type only. Property values can be changed based on the type.

\*

\*InputMap should provide BaseUrl and authKey values.

\*

group, catalog, container, comments

-->

```
Protocol=https
RequestMethod=POST
CommunicationMethod=REST
AuthenticationMethod=Header
Header_Content-Type=application/json
Header_Accept=application/json
Header_Accept-Encoding=UTF-8
Header_Authorization=${authKey}
BaseUrl=https://${authority}/cloudcenter-ccm-backend/api/v1/actions/56/executions
Payload={
    "resourceType" : "VIRTUAL_MACHINE",
    "executionResources" : [
        {
            "id" : "<#list doc['message']['task-started']['requisition']['requisition-entry']['data-values']['data-value'] as
datavalue><#if (datavalue['name'])=='Custom_VMOperations.ExternalVMID'>${datavalue['value']}</#if></#list>"
        }
    ]
}
AssertResponseStatus=${.id}::notNullValue
ExtractResponseStatus=${.id}
ExtractResponseStatusError=${.errors[0].message}
```

### Intercloud.properties ファイルを変更します

以下の場所から Intercloud.properties ファイルを開きます

wildfly-10.1.0.Final \ ServiceLinkServer \ 導入 \ ServiceLink.war \ WEB-INF \ クラス \ config  
ファイルの最後に、以下の形式で FTL ファイルのエントリを追加します

WM\_<name of the CustomAction created in the WM>=json,config/cloud/<FTL File Name>

Ex- WM\_echoHello=json,config/cloud/WM\_echoHello.ftl

### プロパティ ファイルを変更します。

- 次のパスにある SyncCustomWMVMExecutionOperations.properties ファイルを開きます。

wildfly-10.1.0.Final \ ServiceCatalogServer \ deployments \ RequestCenter.war \ WEB-INF \ classes \ config \ cloud  
wildfly-10.1.0.Final \ ServiceLinkServer \ deployments \ ServiceLink.war \ WEB-INF \ classes \ config \ cloud  
ファイルに次のエントリを追加します

<Operation\_Name>=Operation\_Name

例 : echoHello=echoHello

注 : ユーザがライフサイクル操作 (電源オン/電源オフ/再起動など) のライフサイクル操作のカスタム アクションを作成する場合は、以下のファイルを変更する必要があります。以下の場所から以下のファイルを開きます。

wildfly-10.1.0.Final \ ServiceCatalogServer \ deployments \ RequestCenter.war \ WEB-INF \ classes \ config \ cloud  
wildfly-10.1.0.Final \ ServiceLinkServer \ deployments \ ServiceLink.war \ WEB-INF \ classes \ config \ cloud

UpdateWMVMHistoryOperations.properties

<Operation\_Name>=Operation\_Name

例 : powerOnVMCustom=powerOnVMCustom

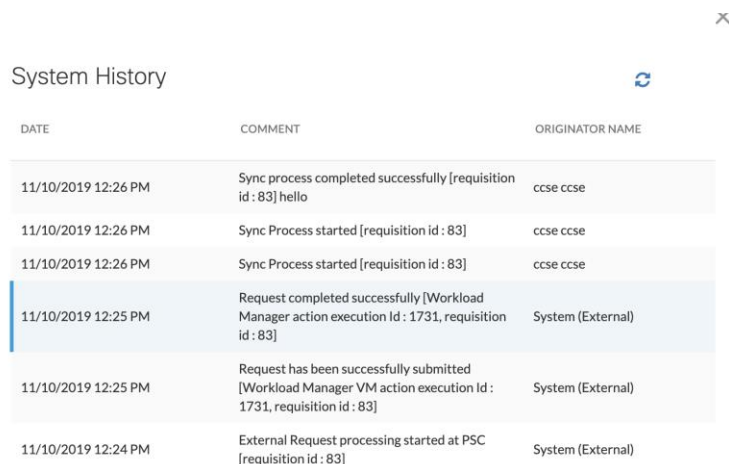
UpdateWMVMStatusOperations.properties

<Operation\_Name>=Operation\_Name

例 : powerOnVM = powerOnVM

- サーバ キャッシュをクリアし、サーバを再起動します。  
カスタム サービスを注文すると、すべてのタスクが完了します  
注文要求のコメントと履歴を確認します (適切なコメントは、コメントと履歴のセクションで更新する必要があります)。

以下はスクリーン ショットの例です。



DATE	COMMENT	ORIGINATOR NAME
11/10/2019 12:26 PM	Sync process completed successfully [requisition id : 83] hello	ccse ccse
11/10/2019 12:26 PM	Sync Process started [requisition id : 83]	ccse ccse
11/10/2019 12:26 PM	Sync Process started [requisition id : 83]	ccse ccse
11/10/2019 12:25 PM	Request completed successfully [Workload Manager action execution id : 1731, requisition id : 83]	System (External)
11/10/2019 12:25 PM	Request has been successfully submitted [Workload Manager VM action execution id : 1731, requisition id : 83]	System (External)
11/10/2019 12:24 PM	External Request processing started at PSC [requisition id : 83]	System (External)

## 展開用のカスタム アクションを使用するディクショナリ、サービス フォーム、およびサービスの設計

ワークロード マネージャ展開でカスタム アクションをサポートするには、次の手順に従います。

### SI ベース ディクショナリの作成

- サービス項目タイプとして WM アプリケーション スタックを選択して、サービス項目ベースのディクショナリを作成します。
  - [サービス デザイナー ( Service Designer ) ] > [ディクショナリ ( Dictionaries ) ]** を選択します。
  - [新規 ( New ) ] > [新しいディクショナリ ( New Dictionary ) ]** を選択して、**[新しいディクショナリ ( New Dictionary ) ]** ページを表示します。
  - [新しい内部ディクショナリの追加 ( Add New Internal Dictionary ) ]** セクションの**[サービス項目 ( Service Item ) ]** フィールドに **WM Application Stack** と入力し、サービス項目を選択します。
  - ディクショナリ名やグループ名などの詳細を入力します。
  - [ディクショナリ属性 ( Dictionary Attributes ) ]** セクションで**[フィールドの追加 ( Add Field ) ]** をクリックして、ユーザ定義フィールド **ShortName**、**JobID**、および **ExecutionID** を追加します。
  - カスタム アクションに必要なディクショナリ内のすべてのフィールド (カスタム FTL で使用されるすべてのフィールド) を選択します。**[JobID]** フィールドが選択され、**[ExecutionID]** フィールドがディクショナリに追加されていることを確認します。
  - [ディクショナリを保存 ( Save Dictionary ) ]** をクリックします



Use	Name	Type	Maximum	Decimals	Multivalue	Show In Grid	Encrypt	PII
<input checked="" type="checkbox"/>	Name	ServiceIdentifier	128	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	DisplayName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Description	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ServiceTypeID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AppVersion	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	DeploymentEnvironment	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Cloud	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CloudAccount	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ShortName	Text	25	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	JobID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AppProfileName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AppProfileDisplayName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	StackID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Status	Text	32	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ApplicationTemplateID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ApplicationURL	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Flag	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ImageURL	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	StartTime	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	CloudConnectionID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ExternalJobID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CustomerID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RequestorID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RequestorEntryID	Number	8	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OrganizationalUnitID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AccountID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AssignmentDate	Date	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	SubmissionDate	Date	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	EstimatedPrice	ServiceItemPrice	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ErrorCode	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ErrorDescription	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ExceptionID	Number	25	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## サービス フォームの作成

2. アクティブ フォーム コンポーネントを作成し、上記で作成したディクショナリをフォームに追加します。

1. **[サービス デザイナー ( Service Designer ) ] > [アクティブ フォーム コンポーネント ( Active Form Component ) ]** を選択します。
2. **[新規 ( New ) ] > [アクティブフォームコンポーネント ( Active Form Component ) ]** を選択します。
3. 新しいフォームの名前と簡単な説明を入力します。
4. フォーム グループ フィールドをクリックし、フォームに関連付けるグループを 1 つ選択します。
5. **[フォームの保存 ( Save Form ) ]** をクリックします。
6. [フォーム コンテンツ (FormContent) ] タブを選択し、**[ディクショナリの追加 ( Add Dictionaries ) ]** をクリックします。
7. [ディクショナリの追加 (Add Dictionaries) ] ダイアログ ボックスで、**[SI ベース ディクショナリ セクションの作成 (Create SI Based Dictionary Section) ]** で作成したディクショナリを検索して選択します。**[追加 ( Add ) ]** をクリックします。

**注：** 必ずワークロード マネージャに属する [展開 (Deployments) ] を選択し、[保存 (Saving) ] フォームの前にすべての必須ディクショナリ フィールドに値が入力されていることを確認してください。

8. **[フォームの保存 (Save Form) ]** をクリックします。

## カスタム アクションを使用するサービスの作成

3. カスタム サービス (CustomOperationService など) を作成し、サービスにフォームを追加します。

1. **[サービス デザイナー ( Service Designer ) ] > [新規 ( New ) ] > [新規サービス ( New Service ) ]** を選択します。
2. 提供されたフィールドに詳細を入力します。
3. **[新しいサービスの追加 (Add New Service) ]** をクリックします。

4. サービスを追加した後、**[全般 ( General )]** タブに情報を入力することで設定を開始できます。
5. **[保存 ( Save )]** をクリックします。
6. 作成したサービスの **[フォーム ( Form )]** タブをクリックします。
7. ウィンドウの左下の **[フォームの追加 ( Add Forms )]** をクリックします。[フォームの追加 (Add Form) ] ポップアップウィンドウが表示されます。
8. [検索 (Search) ] フィールドに、[サービス フォームの作成 (Create Service Form) ] セクションで作成したフォーム名を入力します。
9. フォームを確認し、**[追加 ( Add )]** をクリックします。

The screenshot displays the 'Form' configuration page for a service. At the top, there are tabs for 'General', 'Offer', 'Presentation', 'Form', 'Plan', 'Authorizations', 'Permissions', and 'Extensions'. The 'Form' tab is active.

The main content area is titled 'Service Form For startslagent' and is divided into two columns:

- Forms used in this service:** A list of forms associated with the service. It includes a checked item 'Tenant-Information Form' which contains sub-items: 'XX\_Tenant\_Information\_XX', 'XX\_XX\_TMEnabled\_XX\_XX', and 'XX\_XX\_TeamName\_XX\_XX'. There is also an unchecked item 'Custom\_DepOperationsAFC'.
- Triggering Event:** A list of events that can trigger the service. The events listed are:
  - When the form is submitted (browser-side)
  - When the form is loaded (browser-side)
  - When the form is unloaded (browser-side)
  - After the form is submitted (server-side)
  - Before the form is loaded (server-side)

## サービス タスクの作成

4. サービスの **[プラン ( Plan )]** で下に指定されている 2 つのタスクを作成します。  
実行展開操作タスクと操作完了タスクは必須です。[ステータスの更新 (Update Status) ] タスクはオプションです。

Task	By	This	Subtasks	Subtotal
startslagent		10.00	0.00	10.00
Update history		10.00	0.00	10.00
Total project duration				20.00
Approximate days (as per working hours per day)				2.50

General Participants Email Task Instructions Checklist

Save

Workflow Type: WMAgent

Task name: startslagent

Subtasks execute: one after the other (sequentially) Priority: Normal

Duration: 10.00 hours Effort: 10.00 hours

Condition:

Allow a scheduled start date Form data for start date:

以下の展開でカスタム アクションを実行するには、2つのタスクを作成する必要があります。

- **展開時のアクションの実行** : ワークフロータイプのこのタスクを WMAgent にマッピングします (ワークフロータイプのドロップダウンメニューから選択します)。以下は詳細を更新する必要があるエージェントパラメータです。

パラメータ	サービス データ マッピング
通信方法	REST
Dictionary	[SI ベース ディクショナリの作成 (Create a SI Based Dictionary) ] セクションで作成されたディクショナリの名前を入力します。
OperationName	カスタム アクション名を入力します。ワークロード マネージャのアクション名と一致する <CustomOperationName> を使用することをお勧めします たとえば、ワークロード マネージャで指定されたアクション名が <b>startslagent</b> の場合、カスタム操作名は <b>startslagent</b> である必要があります。

The screenshot shows the 'Agent Parameter Override' window with the following details:

- Outbound Parameter Mappings Table:**

Service Data Mapping	Parameter
REST	CommunicationMethod
CustomDepOperations	Dictionary
startslagent	OperationName
	OperationResource
	OptionalDictionary
- Edit Parameter Values Section:**
  - Parameter: CommunicationMethod
  - Service Data Mapping: REST

- **[操作完了タスク ( Operation Complete Task ) ]** : 同期タスク (Java プラグイン) を使用してタスクを作成し、ドロップダウン メニューから選択して、ポップアップの詳細を更新します。

Sync Task Plugin Class- com.celosis.event.SyncTaskWMPlugin

Param1 : <Dictionary Name (Created in the Create SI based Dictionary).JobID

Param2 : カスタム アクション名。ワークロード マネージャのアクション名と一致するものを使用することを推奨します。

Param3 : <Dictionary Name (Created in the Create SI based Dictionary).Status

Param4 : <Dictionary Name (Created in the Create SI based Dictionary).ShortName

Param5 : WM

5. サービスを保存します。

上記のサービスをサービス項目にマッピングします（このサービスを展開のアクションとして作成します）。

6. [サービス項目マネージャ (Service Item Manager)] > [サービス項目の設計 (Design Service Items)] > [CCS ワークロード マネージャ (CCS Workload Manager)] > [WM アプリケーション スタック - 関連サービス (WM Application Stack-Associated Services)] でカスタム サービスを追加します。
7. [サービスの追加 (Add Service)] をクリックし ([サービス セクションの追加 (Create a Service Section)] で作成したサービスを検索します)、サービスを選択して保存します。これにより、展開の歯車アイコンにサービスが表示されます。

## FTL の作成とプロパティ ファイルの変更

以下は、展開用のカスタム アクション FTL ファイルの例です (start SL agent Invoke webservice Custom アクションの例)

注：FTL ファイルでは、ワークロード マネージャで作成されたカスタム アクション (カスタム アクション WM API を使用して取得可能) のディクショナリ (SI ベースのディクショナリ セクションの作成) とアクション ID を更新する必要があります。

**FTL ファイルの準備ができれば、次の場所に配置する必要があります。**

- wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config\cloud
- wildfly-10.1.0.Final\ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config\cloud

```
<!--
* FTL は、ヘッダー、Post パラメータ、URL (get) パラメータ、およびペイロードをサポートします。
*
* =====
* Header ***** Header_propertyname=Propertyvalue
* Post ***** Post_propertyname=Propertyvalue
* BaseUrl ***** give you the url and followd by query parameters
* Payload ***** Payload=payload content
* Header_AuthToken **** auth token/cookie used for authentication
* =====
*
* Header_Accept-Encoding=UTF-8
* Header_Content-Type=application/xml
```

\* These parameters will apply to the payload type only. Property values can be changed based on the type.

\* InputMap should provide baseUrl and authKey values.

group, catalog, container, comments

-->

```

Protocol=https
RequestMethod=POST
CommunicationMethod=REST
AuthenticationMethod=Header
Header_Content-Type=application/json
Header_Accept=application/json
Header_Accept-Encoding=UTF-8
Header_Authorization=${authKey}
BaseUrl=https://${authority}/cloudcenter-ccm-backend/api/v1/actions/57/executions
Payload={
  "resourceType":"DEPLOYMENT",
  "executionResources":[
    {
      "id":"<#list doc['message']['task-started']['requisition']['requisition-entry']['data-values']['data-value'] as datavalue><#if
(datavalue['name'])=='CustomDepOperations.JobID'>${datavalue['value']}</#if></#list>"
    }
  ]
}
AssertResponseStatus=${id}::notNullValue
ExtractResponseStatus=${id}
ExtractResponseStatusError=${errors[0].message

```

### Intercloud.properties ファイルを変更します

以下の場所から Intercloud.properties ファイルを開きます

wildfly-10.1.0.Final\ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config  
 ファイルの最後に、以下の形式で FTL ファイルのエントリを追加します。

WM\_<name of the CustomAction created in the WM>=json,config/cloud/<FTL File Name>

Ex- WM\_echoHello=json,config/cloud/startslagnet.ftl

### プロパティ ファイルを変更します。

- 次のパスにあるファイル SyncCustomWMDeploymentExecutionOperations.properties を開きます。  
 wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config\cloud  
 wildfly-10.1.0.Final\ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config\cloud  
 ファイルに次のエントリを追加します。  
 <Operation\_Name>=Operation\_Name  
 例 : startslagnet= startslagnet
8. サーバキャッシュをクリアし、サーバを再起動します。  
 カスタム サービスを注文すると、すべてのタスクが完了します  
 注文要求のコメントと履歴を確認します (適切なコメントは、コメントと履歴のセクションで更新する必要があります)。

以下はスクリーンショットの例です。



## System History



DATE	COMMENT	ORIGINATOR NAME
11/10/2019 12:24 PM	Sync process completed successfully [requisition id : 81] Agent SILIST started successfully.	ccse ccse
11/10/2019 12:24 PM	Sync Process started [requisition id : 81]	ccse ccse
11/10/2019 12:24 PM	Sync Process started [requisition id : 81]	ccse ccse
11/10/2019 12:23 PM	Request completed successfully [Workload Manager action execution Id : 1730, requisition id : 81]	System (External)
11/10/2019 12:23 PM	Request has been successfully submitted [Workload Manager VM action execution Id : 1730, requisition id : 81]	System (External)
11/10/2019 12:23 PM	External Request processing started at PSC [requisition id : 81]	System (External)

## 記入済みのサービス オーダー フォームをテンプレートとして保存する機能

- ユーザは、サービス フォームをテンプレートとして保存できるようになりました。これにより、ドラフト ステータスの新しい要求がテンプレートとして作成されます。
- ユーザは、ドラフト ステータスの新しい要求をコピーして作成することや、ログインしているユーザまたはその他のユーザの両方に対して新しい要求として送信することができます。
- テンプレート要求は送信できません。
- テンプレート機能を有効または無効にする管理設定については、以下を参照してください。
  - [管理 (Administration)] -> [設定 (Settings)] -> [テンプレートとして名前を付けて保存 (Save As Template)]
- テンプレート機能を有効にして使用するには、ドラフトとして保存機能も有効にする必要があります。
  - [管理 (Administration)] -> [設定 (Settings)] -> [ドラフトとして保存 (Save As Draft)]
- テンプレート要求が作成されると、[サービス カタログの注文 (Service Catalog Orders)] ページから [テンプレート (Template)] を使用してフィルタリングできます。
- テンプレート要求は以下のアクションを実行できます

注文の作成	顧客およびイニシエータをログイン ユーザとして、新しいドラフト要求としてテンプレート要求をコピーします。
その他の注文の作成	選択した顧客とログイン ユーザをイニシエータとして使用して、新しいドラフト要求としてテンプレート要求をコピーします。
注文の送信	テンプレート要求をコピーし、顧客およびイニシエータをログイン ユーザとして送信します。
その他の注文の送信	テンプレート要求をコピーし、選択した顧客とログイン ユーザをイニシエータとして送信します
添付ファイルの追加	添付ファイルを追加するには
添付ファイルの削除	添付ファイルを削除するには
注文のキャンセル	注文をキャンセルするには

## Jar のアップグレード

- dom4j jar が 2.1.1 にアップグレードされます

```
RequestCenter.war\WEB-INF\lib\dom4j-2.1.1.jar
ServiceLink.war\WEB-INF\lib\lib\dom4j-2.1.1.jar
```

- Commons beanutils jar が 1.9.4 バージョンにアップグレードされます

```
ServiceLink.war\WEB-INF\lib\commons-beanutils-1.9.4.jar
RequestCenter.war\WEB-INF\lib\commons-beanutils-1.9.4.jar
```

- Jackson-databind とその依存 jar は、バージョン 2.10.0 にアップグレードされます

```
RequestCenter.war\WEB-INF\lib\jackson-databind-2.10.0.jar
RequestCenter.war\WEB-INF\lib\jackson-annotations-2.10.0.jar
RequestCenter.war\WEB-INF\lib\jackson-core-2.10.0.jar
```

## Patch\_v12 で導入された nsAPI

### ローカリゼーション API

- メソッド : GET
- URL : <http://10.78.0.241:8080/RequestCenter/nsapi/localization/v1/productStrings?resourceId=109&startIndex=0&recordsPerPage=20>

```
QueryParams:
name = { product String }
resourceId = { ResourceId}
startIndex = 1
recordsPerPage = 20
```

- 応答 :

```
{
```



```

"Result" : {
  "totalRecords" : 1,
  "recordsReturned" : 1,
  "records" : {
    "109" : [
      {
        "id" : 0,
        "outOfSync" : 0,
        "languageName" : "US English",
        "propertyId" : 109,
        "value" : "Text",
        "localeId" : 1
      },
      {
        "id" : 0,
        "outOfSync" : 0,
        "languageName" : "Chinese (Simplified)",
        "propertyId" : 109,
        "value" : "文本",
        "localeId" : 6
      }
    ]
  }
}

```

## テンプレート API として保存

- カート ページからテンプレートとして保存
  - URL : <http://localhost:8080/RequestCenter/nsapi/transaction/saverequisitionasdraft?responseType=json&customerId=1&saveAsTemplate=true>
  - メソッド : POST
  - ヘッダー : Content-Type : application/json
  - ペイロード :
 

```

[
  {
    "requisitionId" : "2122",
    "requisitionEntryId" : "2135",
    "quantity" : "1",
    "pricePerUnit" : "$ 0.00"
  }
]

```
  - 応答 : `<nsapi-response>` 要求がドラフトとして正常に保存されました。 `</nsapi-response>`
- 注文の作成またはその他向けの注文作成
  - URL : [/RequestCenter/nsapi/catalog/v1/requisition/{id}/copyAsDraft/customer/{customer\\_id}/initiator/{initiator\\_id}](/RequestCenter/nsapi/catalog/v1/requisition/{id}/copyAsDraft/customer/{customer_id}/initiator/{initiator_id})
  - Method : POST
  - ヘッダー : Accept : application/json
  - 応答 :
 

```

{
  "HashMap" : {
    "RequisitionId" : "2103",
    "Message" : "The Template Requisition has been successfully copied."
  }
}

```

- 注文の送信またはその他の注文の送信
  - URL :  
[/RequestCenter/nsapi/catalog/v1/requisition/{id}/orderFromTemplate/customer/{customer\\_id}/initiator/{initiator\\_id}](/RequestCenter/nsapi/catalog/v1/requisition/{id}/orderFromTemplate/customer/{customer_id}/initiator/{initiator_id})
  - メソッド : POST
  - ヘッダー : Accept : application/json
  - 応答 :
 

```
{
            "RequisitionSubmit" : {
              "id" : 2126,
              "customer" : "admin admin",
              "initiator" : "admin admin",
              "startedDateRaw" : 1574136178633,
              "startedDate" : "11/19/2019 1:32 AM",
              "status" : "Ordered"
            }
          }
```

## ワークロードマネージャをサポートするように変更された **CloudCenter API**

- cloudProvider クエリ パラメータで変更された Cloud Center API
  - Prime Service Catalog に対する単一/複数のアプリケーション プロファイルの更新
    - /RequestCenter/nsapi/cloudCenter/applicationProfiles?connectionId=2&cloudProvider=<WM or CLIQR>
  - VM 詳細の取得
    - /RequestCenter/nsapi/cloudCenter/vm/<vm\_id>/console? cloudProvider=<WM or CLIQR>
  - VM コンソール アクセスの取得
    - /RequestCenter/nsapi/cloudCenter/vm/<vm\_id>/console? cloudProvider=<WM or CLIQR>
  - Windows VM ログインの詳細の取得
    - /RequestCenter/nsapi/cloudCenter/vm/windows/<vm\_id>/logindetails/?connectionId=2&cloudProvider=<WM or CLIQR>

## 12.1\_Patch\_v11 で導入

### 新しいサービス マネージャ UI からタスクのリストをエクスポートする機能

Prime Service Catalog 12.1 では、[エクスポート (Export) ] ボタンが Service Manager の右下隅にあるその他のアクションとともに導入されました。データは CSV および EXCEL 形式でエクスポートできます。

### ブラウザのアップグレード

- IE-11
- Firefox-67
- Chrome-75
- Safari-12

### 認定

- **CCS 5.0.1 AO 統合 : CCS 5.1 AO に対する認定**

CCS 5.0 AO は、このパッチではサポートされていません。CCS 5.0.1 AO 統合 とすでに統合している場合は、次から PSC AO ワークフローを再インポートする必要があります。

[https://github.com/cisco/ActionOrchestratorContent/tree/master/workflow-examples/PSCAOIntegrationWorkflow\\_definition\\_workflow\\_016QL80Y3D2J62KI5hebU1SiGXzunNcCkQi](https://github.com/cisco/ActionOrchestratorContent/tree/master/workflow-examples/PSCAOIntegrationWorkflow_definition_workflow_016QL80Y3D2J62KI5hebU1SiGXzunNcCkQi)

- **UCSD 6.7 認定 : UCSD 6.7 ( Fenced )、UCSD 6.5 ( MSP )、UCSPM 2.5 に対して認定**

### データベースのアップグレード

- **Microsoft SQLServer 2016(SP2-CU7) (KB4495256) - 13.0.5337.0 (X64) )**
- **Oracle 19c 公式 Weblogic 12C は Oracle 19C をサポートしていません。**

注 : Cognos 10.2.2 向け Oracle 19C の公式サポートはありません。

### Jar のアップグレード

セキュリティの修正は、パッチの zip ファイルとともに readme ドキュメントで行われ、更新されます。

#### jQuery ライブラリ

Prime Service Catalog で使用される JQuery バージョンがバージョン 3.4.1 にアップグレードされます。

## タスクを取得する API

これは、queryParams を追加することで拡張された既存の API です。

- メソッド : GET
- URL : RequestCenter/nsapi/transaction/v2/tasks
- 応答メッセージ :  
"Priority","RequisitionID","DueOn","ServiceTask","ServiceName","Initiator","CustomerOU","CustomerName","Performer",  
"Queue","Status","Effort","RequisitionEntryID","ScheduledStart","TaskType"

#### QueryParam :

- Action : EXPORT

- ViewID : {viewId}



## 12.1\_Patch\_v10 で導入

### アクション オーケストレーションのポーラー

注：フラグ `ao.poller.cron` および `ao.poller.health.check.cron` は、ポーラーが有効になっていない場合でも追加する必要があります。

AO ポーラーを設定するには、一部のプロパティを `newscale.properties` および `support.properties` に追加する必要があります。

次の 2 つのフィールドは、サーバを起動するために `newscale.properties` に追加する必要があります。

```
ao.poller.cron=0 0/5 * * * ?
```

```
ao.poller.health.check.cron=0 10/15 * * * ?
```

#### `newscale.properties` に追加するプロパティ

```
#AO Poller
```

```
#####
```

```
#Cron Expression wakes up poller every 10 minutes of an hour
```

```
ao.poller.cron=0 0/10 * * * ?
```

```
#Cron Expression wakes up health check for 10th min and 15 mins thereafter of the hour ex : 10,25,40,55 minutes
```

```
ao.poller.health.check.cron=0 10/15 * * * ?
```

```
#High Availability Health checks threshold, this should be greater than Poller cron time specified in minutes
```

```
ao.healthCheck.threshold=91
```

```
#####
```

#### `support.properties` に追加するプロパティ

```
##### AO Poller Settings #####
```

```
#Flag to enable/disable the poller
```

```
ao.poller.enable=true
```

### 分離された UI のサポート

V9 バッチ分離 UI は、スタンドアロン Wildfly トポロジでサポートされていましたが、このパッチでは、スタンドアロン Wildfly トポロジに加えて、クラスタ トポロジをサポートしています。

Prime Service Catalog がインストールされているサーバとは別のサーバで、分離された Service Catalog および Service Manager の Web サイトをホストします。

- クラスタ化された WildFly アプリケーション サーバで実行される PSC のサポート。
- スタンドアロンおよびクラスタ化された WebLogic アプリケーション サーバで実行される PSC のサポート。
- クラスタ化されたセットアップで、分離された Service catalogue および Service manager Web サイトのホストをサポートします。
  
- アプリケーション サーバ
  - Wildfly スタンドアロン
  - Wildfly クラスタ トポロジ
  - Weblogic スタンドアロン
  - Weblogic クラスタ
  
- [LB サーバ (LB Servers) ]
  - IIS および Apache httpd

- データベース サーバ
  - SQL および Oracle
- UI サーバ
  - Apache Tomcat 9.0 および Nginx 1.15.8
- ブラウザのサポート
  - IE11
  - Chrome 74
  - Firefox 66
  - safari 12.x

## UI 分離のための Tomcat クラスタの設定

**前提条件** : Wildfly 10.1 2 VM ノードまたは PSC と LB (IIS または Apache httpd) を備えた Weblogic 2 ノードが稼働している必要があります。すべてのノードが同じドメインにある必要があります。

2つの異なるノードに tomcat ノードを設定する手順は次のとおりです。

1. tomcat 9.0.14 バージョンをインストールします。
2. Apache ディレクトリに移動し、server.xml を開きます。ファイルの場所 : C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf
3. defaultHost を検索し、システムの FQDN で更新し、同じ行に jvmRoute = "node1" を追加します。以下に例を示します。  
<Engine name="Catalina" defaultHost="<FQDN>" jvmRoute="node1">
4. ホスト名を検索し、システムの FQDN を更新します。以下に例を示します。

```
<Host name="<FQDN>" appBase="webapps"
  unpackWARs="true" autoDeploy="true">
```

他のノードにも同じ設定を行う必要があります。

以下の URL にアクセスすると、tomcat のホームページが表示されます。

```
http://<Host_Name>:<PORT>/
```

上記ノードの LB の設定

1. Apache 2.4 のダウンロードとインストール  
参照 : <https://httpd.apache.org/docs/2.4/platform/windows.html>
2. httpd.conf ファイルを開き、ファイルの最後に以下のエントリを追加します。  

```
ServerName <FQDN>
KeepAlive On
DocumentRoot "c:/Apache24/htdocs"
#ServerName VMDEVSLw2k8-ovf
<Directory "c:/Apache24/htdocs">
  EnableSendfile off
</Directory>
Listen FQDN:80
AcceptFilter http none
AcceptFilter https none
<VirtualHost *:80>
  ServerName <FQDN>
  ProxyRequests Off
  ProxyPreserveHost On
  <Location /balancer-manager>
    SetHandler balancer-manager
    Order allow,deny
    Allow from all
  </Location>
  Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/"
  env=BALANCER_ROUTE_CHANGED
  <Proxy balancer://mycluster>
    BalancerMember ajp:// <FQDN>:8009 route=1
```

```

BalancerMember ajp://abc.cisco.com:8009 route=2
#BalancerMember ajp://IP_Address:8009 route=3
#BalancerMember ajp://IP_Address:8009 route=4
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass / balancer://mycluster/
ProxyPassReverse / balancer://mycluster/
</VirtualHost>

```

3. httpd Apache サーバを再起動します。
4. 以下の URL にアクセスすると、ホーム ページが表示されます。HA を確認するために、上記のノードのいずれかを停止して更新すると、Tomcat のホームページが表示されます。  
http://<Host\_Name>:<PORT>/

#### WFクラスタノードでのCROS設定の有効化

注：以下の設定は Wildfly 専用です。

1. 「wildfly-10.1.0.Final\domain\configuration」から domain.XML を開きます。
2. 強調表示されたエントリをファイルに追加する  
注：domain.xml には 2 つのプロファイルがあります。
3. プロファイル名はデフォルトであり、両方のプロファイル エントリの HA を作成する必要があります

```

<buffer-cache name="default"/>
<server name="default-server">
  <ajp-listener name="ajp" socket-binding="ajp" scheme="https" max-ajp-packet-size="65536"/>
  <http-listener name="default" socket-binding="http" redirect-socket="https" enable-http2="true"/>
  <host name="default-host" alias="localhost">
    <location name="/" handler="welcome-content"/>
    <filter-ref name="server-header"/>
    <filter-ref name="x-powered-by-header"/>
    <filter-ref name="Access-Control-Allow-Origin"/>
    <filter-ref name="Access-Control-Allow-Headers"/>
    <filter-ref name="Access-Control-Allow-Credentials"/>
  </host>
</server>
<servlet-container name="default">
  <jsp-config/>
  <websockets/>
</servlet-container>
<handlers>
  <file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
</handlers>
<filters>
  <response-header name="server-header" header-name="Server" header-value="WildFly/10"/>
  <response-header name="x-powered-by-header" header-name="X-Powered-By" header-
value="Undertow/1"/>
  <response-header name="Access-Control-Allow-Origin" header-name="Access-Control-Allow-Origin"
header-value="http://abc.cisco.com"/>
  <response-header name="Access-Control-Allow-Headers" header-name="Access-Control-Allow-
Headers" header-value="accept, authorization, content-type, x-requested-with, Content-Type"/>
  <response-header name="Access-Control-Allow-Credentials" header-name="Access-Control-Allow-
Credentials" header-value="true"/>
</filters>
</subsystem>

```

4. Newscale プロパティの変更：

#### WildFly Newscale:

```

#####
#decoupled UI domain name
decoupled.cookie.domain=abc.cisco.com
#####
#decoupled UI authentication mode
decoupled.authentication=true
#####
#decoupled UI landing page url for servicecatalog module

```

```

decou-
pled.servicelog.url=http://abc.cisco.com/RequestCenter/website/CustomSC/application/index.html
#####
#decoupled UI landing page url for servicemanager module
decou-
pled.servicemanager.url=http://abc.cisco.com/RequestCenter/website/CustomSM/application/servicemanager.htm
l?route=servicemanager#homepage
#####
decoupled.saml.logout.url=http://abc.cisco.com/Logout/ssoLogout.html
decoupled.servicelog.mobile.url=http://orac-estr-
w2k12.cisco.com/RequestCenter/website/ServiceCatalogMobileWebsite/application/index.html

```

```

Weblogic Newscale:
#####
#decoupled UI domain name
decoupled.cookie.domain=.abc.com
#####
#decoupled UI authentication mode
decoupled.authentication=true
decoupled.domain.url=http:// abc.cisco.com
#####
#decoupled UI landing page url for servicelog module
decoupled.servicelog.url=http:// abc.cisco.com
/RequestCenter/website/ServiceCatalogWebsite/application/index.html
#####
#decoupled UI landing page url for servicemanager module
decoupled.servicemanager.url=http:// abc.cisco.com
/RequestCenter/website/ServiceManagerWebsite/application/servicemanager.html?route=servicemanager#homep
age
#####
decoupled.saml.logout.url=http://abc.cisco.com/Logout/ssoLogout.html

```

5. サーバの再起動は、アプリケーション サーバのタイプに関係なく常に適用されます。

#### IIS の Cors の有効化

**注：**アプリケーション サーバが IIS を使用している場合は、以下のエントリを作成する必要があります。

1. IIS が配置されている VM に移動し、以下のリンクから CORS モジュールをインストールします。  
<https://www.iis.net/downloads/microsoft/iis-cors-module>
2. X64 インストーラをダウンロードし、CORS モジュールをインストールします。
3. IIS の Web.config を開き、system.webserver 内に以下のタグを追加します。

```

<cors enabled="true">
  <add origin="http://vm236.cisco.com" allowCredentials="true" >
    <allowMethods>
      <add method="PUT" />
      <add method="GET" />
      <add method="POST" />
      <add method="DELETE" />
      <add method="OPTIONS" />
    </allowMethods>
  </add>
</cors>
<httpProtocol>
  <customHeaders>
    <remove name="X-Powered-By" />
    <add name="Access-Control-Allow-Headers" value="x-requested-with, Content-Type, origin, Authoriza-
tion, Accept, locale, access_token, utid, Range" />
    <add name="X-Powered-By" value="ASP.NET,Undertow/1" />
  </customHeaders>
</httpProtocol>

```



4. IIS を再起動します。

## 異なる Web サーバでの PSC UI の設定

1. Tomcat がインストールされている VM に移動します。
2. Web サーバの展開フォルダ（この場合は C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps）に移動します。
3. 次のようなフォルダ構造を作成します
  - i. RequestCenter、ログインおよびログアウト フォルダを作成します
  - ii. RequestCenter 内に、website というサブフォルダを作成します
    - a. PSC ベース インスタンス「wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website and place under the webapps\RequestCenter/website/ of tomcat VM」から ServiceCatalog、Service Manager、ServiceCatalogMobileWebsite、およびFormServer フォルダをコピーし、tomcat VM の webapps\RequestCenter/website/ の下に配置します。
4. dist フォルダの内容を wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website>Login Web サイトからコピーし、tomcat VM の webapps / Login に配置します。
5. LoginConfig.Json の形式を、wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website にコピーし、webapps\RequestCenter/website に配置します。
6. wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war からヘルプと jsbundle フォルダをコピーし、tomcat vm の webapps / RequestCenter に貼り付けます。
7. ssologout フォルダの内容をwildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website>LoginWebsite\ssologout から tomcat VM のwebapps / Logout フォルダにコピーします。
8. 次に、json ファイルを次のように変更します。
 

ホスト : PSC ベース インスタンス : 分離された Web サイトの場合は必須  
Cookie ドメイン : ベース システムが属するドメイン。例 : VM が cisco.com に属し、Cookie ドメインが.cisco.com である場合

serviceCatalogUrl : ホストされている tomcat サーバの IP またはホスト名である SC URL を指定します。  
serviceManagerUrl : ホストされている tomcat サーバの IP またはホスト名である SC URL を指定します。

ログイン ページ情報を設定するためのプロパティはここでパスすることができます。

copyRightText : ログイン ページに表示するコピー メッセージ  
displayString : 製品名 (Cisco Service Prime Catalog がデフォルトです)  
MarketingVersion : 製品バージョン

## LoginConfig.json の例

```
{
  "host": "http://<HostName>:<Port>",
  "cookieDomain": ".cisco.com",
  "ssoUrl": "",
  "loginModule": {
    "serviceCatalogUrl": "http://<Host_Name_of_Tomcat>/RequestCenter/website/ServiceCatalogWebsite/application/index.html",
    "serviceManagerUrl": "http://
<Host_Name_of_Tomcat>/RequestCenter/website/ServiceManagerWebsite/application/servicemanager.html?route=servicemanager#homepage"
    "serviceCatalogMobileUrl": "http://orac-estr-w2k12.cisco.com/RequestCenter/website/ServiceCatalogMobilewebsite/application/index.html"
  },
  "loginPageInfo": {
    "copyRightText": "",
    "displayString": "",
    "marketingVersion": ""
  }
}
```

9. 両方の VM 上で Tomcat サーバ ノードを再起動し、Apache LB を再起動します。
10. 以下のように Apache LB URL サーバにアクセスします。  
[http://<Apache\\_LB>:<Apache\\_Port>/Login](http://<Apache_LB>:<Apache_Port>/Login)  
URL で PSC ログインページに移動したら、ユーザはログインでき、Service Catalog と Service Manager を表示できる必要があります。

## トラブルシューティング

## 1. IE 11 : UI の LB を使用して PSC にアクセスできない場合

Java スクリプト : インターネットおよび信頼済みサイトで有効にする必要があります

SCRIPT7002 : XMLHttpRequest : Network Error 0x80070005, Access is denied.

[セキュリティ (Security) ] 領域 ([インターネット オプション (Internet Options) ]) の [セキュリティ (Security) ] が有効になっている必要があります。

次の設定 : [その他 (Miscellaneous) ] : [ドメイン全体のデータソースにアクセス (Access data source between domain) ]。

## 2. SAML ログアウトが正しく動作していませんか? 初回ログインに成功しましたが、ログアウト時に再ログインに失敗し、500 個の内部エラーが発生しました。(再ログインすると、RC.war にリダイレクトされます)

セッションのタイムアウトを 30 から 20 に変更します。

これは web.xml の元の設定です (両方の tomcat クラスタ ノード)。

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

20 に変更

```
<session-config>
  <session-timeout>20</session-timeout>
</session-config>
```

Tomcat ノードを再起動します。この設定後に問題は発生しません。

## 3. セッションが期限切れになると、ストアの再ログイン空白ページでログアウトし、コンソールでエラーが表示され、ブラウザの更新時にログアウトされ、ユーザはログイン ページに移動されます。この問題を解決するには、毎回ブラウザのキャッシュをクリアする必要があります。

要求URL :

[https://vmqa20\\_node1.cisco.com/RequestCenter/nsapi/v1/common/resourcestrings?utid=C0EEB917AB123082EC83DBB34CFF87CD](https://vmqa20_node1.cisco.com/RequestCenter/nsapi/v1/common/resourcestrings?utid=C0EEB917AB123082EC83DBB34CFF87CD)

要求メソッド : PUT

ステータス コード : 401 Unauthorized

解決策 : </Directory> タグの後に以下のエントリで httpd.conf ファイルを更新します。

```
AcceptFilter http none
```

```
AcceptFilter https none
```

## 4. ユーザが SC で検索を実行すると、LB システムがハングする

解決策 : </Directory> タグの後に以下のエントリで httpd.conf ファイルを更新します。

```
AcceptFilter http none
```

```
AcceptFilter https none
```

## グリッドとページネーション

PSC は現在、マテリアル UI サービス フォームのグリッドとページネーションをサポートしています。

## グリッドのカラーマッピング

グリッドは、グリッドのプロパティ (フォント、色) をスクリプト タブの servicedesigner モジュールで定義できるカラーマップによってサポートされます。これは、アクティブ フォーム イベントに関連付ける必要があります。

以下は、グリッド カラーマップのサンプル スクリプトです。

```
Grid_DDRReqStatus(form){
  var colorMap={
    "dictionaryName":"GridRequisitionStatus","colorMap":{
      "fieldName":"StatusID",
      "colorList":[
        {"value":"1","rowColor":"#FF0000","cellColor":"#00FF00","textColor":"#0000FF"},
        {"value":"2","rowColor":"green","cellColor":"yellow","textColor":"black"},
        {"value":"3","rowColor":"blue","cellColor":"red","textColor":"green"},
        {"value":"-5","rowColor":"red","cellColor":"red","textColor":"black"}
      ]
    }
  }
```

```

    }
  ]
  });

  ISF.isf_gridCellColor(form,colorMap);
}

```

グリッド内のすべてのデータをJSON形式で取得する方法。

```

/**
 *method to get grid data in json format
 @param form : object - service form object which contains all form specific information
 @param gridDictionaryName : string-ID of grid dictionary
 */
isf_getGridData(form, gridDictionaryName)

```

セルインデックスによってグリッドセル値を取得する方法

```

/**
 *method to get grid cell value by cell index
 @param form : object - service form object which contains all form specific information
 @param gridDictionaryName : string-ID of grid dictionary
 @param rowIndex : int-row index(0 based) of grid
 @param collIndex : int-column index(0 based) of grid
 */
isf_getGridCellValue(form, gridDictionaryName, rowIndex, collIndex)

```

フィールド名でグリッドセル値を取得する方法

```

/**
 @param form : object - service form object which contains all form specific information
 @param gridDictionaryName : string-ID of grid dictionary
 @param rowIndex : int-row index(0 based) of grid
 @param colKey : string-column key of grid(field name)
 */
isf_getGridCellValueByKey(form, gridDictionaryName, rowIndex, colKey)

```

セルインデックスによってグリッドセル値を設定する方法

```

/**
 @param form : object - service form object which contains all form specific information
 @param gridDictionaryName : string-ID of grid dictionary
 @param rowIndex : int-row index(0 based) of grid
 @param collIndex : int-column index(0 based) of grid
 @param value : string-new value to be updated
 */
isf_setGridCellValue(form, gridDictionaryName, rowIndex, collIndex, value)

```

フィールド名でグリッドセル値を設定する方法

```

/**
 *method to get grid cell value by cell index
 @param form : object - service form object which contains all form specific information
 @param gridDictionaryName : string-ID of grid dictionary
 @param rowIndex : int-row index(0 based) of grid
 @param colKey : int-column name of grid(fieldName)
 @param value : string-new value to be updated
 */
isf_setGridCellValueByKey(form, gridDictionaryName, rowIndex, colKey, value)

```

## グリッドの行の色、セルの色、またはセルのテキストの色を設定する方法

特定のディクショナリとフィールドのセルの値の一致に基づいて、グリッドの行の色/セルの色/セルのテキストの色を設定します。色を設定するためのデータは、ディクショナリ名、フィールド名、照合するセル値、および設定する色の値を含む json の形式で提供されます。

```
/**
 *
 * @param form : object - service form object which contains all form specific information
 * @param jsonColorMap : json-standard json containing color map for cell value
 */
isf_gridCellColor(form, jsonColorMap)
```

### 色を設定するサンプルJSON

```
var colorMap = [{
  "dictionaryName" : "",
  "colorMap" : [{
    "fieldName" : "Text",
    "colorList" : [
      { "value" : "abc", "rowColor" : "red", "cellColor" : "green", "textColor" : "blue" },
      { "value" : "xyz", "rowColor" : "green", "cellColor" : "yellow", "textColor" : "black" },
    ]
  },
  {
    "fieldName" : "Phone",
    "colorList" : [
      { "value" : "1234", "rowColor" : "red", "cellColor" : "green", "textColor" : "blue" },
      { "value" : "5678", "rowColor" : "green", "cellColor" : "yellow", "textColor" : "black" },
    ]
  }
]
```

### カラーマッピングを使用したグリッドサービスフォームの作成手順

1. [サービスデザイナー (Service Designer)] -> [スクリプト (Scripts)] ページを使用して JavaScript 関数を作成します。

```
var colorMap={
  "dictionaryName":"Grid_UCSDVMStatus",
  "colorMap":{
    "fieldName":"Status",
    "colorList":[
      {"value":"Active","rowColor":"yellow","cellColor":"","textColor":"green"},
      {"value":"Inactive","rowColor":"orange","cellColor":"","textColor":"red"}
    ]
  }
};
```

VMId	Status
110	Inactive
111	Inactive
180	Inactive
1470	Inactive
1471	Active

2. この JS 関数を [AFC] -> [アクティブフォームの動作 (Active Form Behavior)] -> [イベントのトリガー (Triggering Event)] -> [フォームがロードされたとき (When when the form is load)] に追加します
3. この関数は、マテリアル サービス フォームのロード中にロードされます。

## 電子メール テンプレートでのサービス項目ネームスペースのサポート

サービス項目のネームスペース (freemarker ベース) は、電子メール テンプレートでサポートされています。ネームスペース サポートは、プロビジョニング vm サービスの仮想マシンなど、サービスのコンテキスト内のサービス項目用です。サービス項目のネームスペースは、次のシナリオでサポートされます。

[プラン (Plan) ] タブのタスク

1. サービス項目タスクとそれに続くサービス リンク タスク
2. UCSD 同期タスク
3. Cloud Center 同期タスク

UCSD VM の例は次のようになります。

```
[#if serviceitem.SiUCSDVM.Name??] Name : [=serviceitem.SiUCSDVM.Name][/#if]
[#if serviceitem.SiUCSDVM.RAM??] RAM : [=serviceitem.SiUCSDVM.RAM][/#if]
[#if serviceitem.SiUCSDVM.HostName??] HostName : [=serviceitem.SiUCSDVM.HostName][/#if]
この機能は、newscale.properties で次のプロパティを設定して有効にする必要があります。-email.use.serviceitem.namespace=true
[#if serviceitem.SiUCSDVM.Name??] Name : [=serviceitem.SiUCSDVM.Name][/#if]
[#if serviceitem.SiUCSDVM.RAM??] RAM : [=serviceitem.SiUCSDVM.RAM][/#if]
[#if serviceitem.SiUCSDVM.HostName??] HostName : [=serviceitem.SiUCSDVM.HostName][/#if]
```

サービス項目のサブスクリプション レコードは、次のように呼ぶことができます

```
[#if serviceitem.<serviceItemTypeName>.header.<subscriptionAttributeName>??] Name : [= serviceitem.
<serviceItemTypeName> .header. <subscriptionAttributeName> ] [/#if].
```

UCSD VM の例は次のようになります。

```
[#if serviceitem.SiUCSDVM.header.RequisitionID??]ServiceItem_ReqID : [=serviceitem.SiUCSDVM.header.RequisitionID][/#if]
[#if serviceitem.SiUCSDVM.header.CustomerID??]ServiceItem_CustomerID : [=serviceitem.SiUCSDVM.header.CustomerID][/#if]
[#if serviceitem.SiUCSDVM.header.AssignedDate??]ServiceItem_AssignedDate :
[=serviceitem.SiUCSDVM.header.AssignedDate?date][/#if]
[#if serviceitem.SiUCSDVM.header.SubmittedDate??]ServiceItem_SubmittedDate :
[=serviceitem.SiUCSDVM.header.SubmittedDate?date][/#if]
[#if serviceitem.SiUCSDVM.header.OrganizationUnitID??]ServiceItem_OUID :
[=serviceitem.SiUCSDVM.header.OrganizationUnitID][/#if]
[#if serviceitem.SiUCSDVM.header.AccountID??]ServiceItem_AccID : [=serviceitem.SiUCSDVM.header.AccountID][/#if] [#if
serviceitem.SiUCSDVM.header.AgreementID??]ServiceItem_AgreementID : [=serviceitem.SiUCSDVM.header.AgreementID][/#if]
```

サービス項目には、親サービス項目 (コンテナの一部である仮想マシンなど) を設定できます。コンテナは次のように参照できます。

```
[#if serviceitem.SiUCSDVM.parent.Name??]Application Name : [=serviceitem.SiUCSDVM.parent.Name][/#if]
[#if serviceitem.SiUCSDVM.parent.Status??] Status : [=serviceitem.SiUCSDVM.parent.Status][/#if]
[#if serviceitem.SiUCSDVM.parent.ContainerID??]Container ID : [=serviceitem.SiUCSDVM.parent.ContainerID][/#if]
[#if serviceitem.SiUCSDVM.parent.DisplayName??]DisplayName : [=serviceitem.SiUCSDVM.parent.DisplayName][/#if]
```

サービス項目に子サービス項目がある場合 (複数の仮想マシンを持つコンテナなど) 。これらは次のように参照できます。

```
[#if serviceitem.SiUCSDContainer.VirtualMachines?has_content]
Details of VMs : -
[#assign indx=1]
[#list serviceitem.SiUCSDContainer.VirtualMachines as vm]
VM [= indx] : -
[#if vm.Name ??] Name : [=vm.Name][/#if]
[#if vm.UCSDName??] UCSDName : [=vm.UCSDName][/#if]
[#if vm.HostName??] HostName : [=vm.HostName][/#if]
[#if vm.IPAddress??] IPAddress : [=vm.IPAddress][/#if]
[#if vm.MACAddress??] MACAddress : [=vm.MACAddress][/#if]
[#if vm.VMId??] VMId : [=vm.VMId][/#if]
[#if vm.Description??] Description : [=vm.Description][/#if]
[#assign indx=indx+1]
[/#list]
```

[/#if]

詳細については、フリーマーカーのマニュアルを参照してください。  
 ネームスペース電子メールテンプレートは、次のタスク イベントでサポートされます。

- i. Notify when activity completes
- ii. Notify when activity is cancelled
- iii. Notify when external task fails

## ServiceLink を使用しない PSC の Host RequestCenter

### PSC を使用した JMS スタンドアロン設定

パッチ v10 には、ServiceLink なしで RequestCenter をホストするオプションがあります。  
 次の手順を実行する必要があります

**前提条件：** ユーザは Active MQ Artemis Server をインストールし、クライアント マシンから管理コンソールにアクセスできる必要があります。

### Artemis サーバのインストールと設定の手順

1. <https://activemq.apache.org/components/artemis/download/> から Artemis ソフトウェアをダウンロードし、解凍して C ドライブに配置します。  
 インストールの詳細については、以下のリンクを参照してください。  
<https://activemq.apache.org/components/artemis/documentation/1.0.0/running-server.html>
2. インストール後、ブローカー フォルダを作成します。
3. 次のコマンドを実行します

例：次の場所に移動し、コマンドを実行します。  
 C:\apache-artemis-2.7.0-bin\apache-artemis-2.7.0\bin

1. `activemq create mybroker`  
 bin フォルダの下に `mybroker` フォルダを作成し、プロンプトが表示されたらユーザ名とパスワードを入力します。  
 注：ユーザは、ファイルのセットの一部の設定を変更し、変更する前にファイルのバックアップを取得する必要があります（例：場所： `apache-artemis-2.7.0 \ bin \ mybroker` ）。
    - `broker.xml`
    - `bootstrap.xml`
    - `jolokia-access.xml`
    - `login.config`
2. `mybroker/etc` 内で `bootstrap.xml` を検索し、`bootstrap.xml` を開いて `<web bind="http://localhost:8161" path="web">` を検索し、`localhost` を ARTEMIS サーバがインストールされている VM の IP に置き換えます。
  3. 同じ場所で `broker.xml` を開き、以下を検索し、`0.0.0.0` を Active MQ ARTEMIS Server `<name>0.0.0.0</name>` のホスト名に置き換えます  

```
<acceptor
name="artemis">tcp://0.0.0.0:61616?tcpSendBufferSize=1048576;tcpReceiveBufferSize=1048576;protocols=CORE,AMQP,STOMP,HORNETQ,MQTT,OPENWIRE;useEpoll=true;amqpCredits=1000;amqpLowCredits=300</acceptor>
<!-- AMQP アクセプタ。AMQP トラフィックをデフォルトの AMQP ポートでリッスンします。
<acceptor
name="amqp">tcp://0.0.0.0:5672?tcpSendBufferSize=1048576;tcpReceiveBufferSize=1048576;protocols=AMQP;useEpoll=true;amqpCredits=1000;amqpLowCredits=300</acceptor>
<!-- STOMP Acceptor. -->
<acceptor
name="stomp">tcp://0.0.0.0:61613?tcpSendBufferSize=1048576;tcpReceiveBufferSize=1048576;protocols=STOMP;useEpoll=true</acceptor>
<!-- HornetQ Compatibility Acceptor. Enables HornetQ Core and STOMP for legacy HornetQ clients. -->
<acceptor
name="hornetq">tcp://0.0.0.0:5445?anycastPrefix=jms.queue.;multicastPrefix=jms.topic.;protocols=HORNETQ,STOMP;useEpoll=true</acceptor>
<!-- MQTT Acceptor -->
```

```
<acceptor
name="mqtt">tcp://0.0.0.0:1883?tcpSendBufferSize=1048576;tcpReceiveBufferSize=1048576;protocols=MQTT;useEpoll=
true</acceptor>
```

- broker.xml ファイルの `<auto-create-jms-topics>true</auto-create-jms-topics>` を検索します  
以下の行を追加し、ファイルを保存して閉じます。  
`<auto-delete-queues>false</auto-delete-queues>`
- 同じ場所で `jolokia-access.xml` にアクセスし、`<allow-origins>*://localhost*</allow-origins>` を検索し、ARTEMIS サーバがインストールされている VM の IP に置き換えます。
- 同じ場所で `login.config` ファイルを開き、十分に `org.apache.activemq.artemis.spi.core.security.jaas.PropertiesLoginModule` を検索し、十分な単語を必須に置き換えます。同じファイルで `org.apache.activemq.artemis.spi.core.security.jaas.GuestLoginModule` を検索し、十分な単語を必要に応じて置き換えます。

Active MQ Management コンソールにアクセスします。 [http://<VM\\_IP>:8161/console](http://<VM_IP>:8161/console)

クレデンシャルでログインします。ブローカの作成時に使用したものと同一パスワードを使用する必要があります。次に、`jmsrunner` スクリプトを実行する構文を示します。このスクリプトは、スタンドアロン JMS が Wildfly で動作するように設定を変更するために使用されます。

## Linux の場合

### Microsoft SQLServer の使用 :

```
jmsrunner.sh SqlServer {CPSCServer} {CPSCPort} {CPSCDatabaseName}
{CPSCUser} {CPSCPassword} {ActiveMQServer} {ActiveMQPort} {ActiveMQUser}
{ActiveMQPassword} {WildflyConfigPath} {RequestCenterWarPath} {QueueSuffix} {isWildfly18}
```

### Oracle の使用メソッド :

```
jmsrunner.sh Oracle {CPSCServer} {CPSCPort} {CPSCSID} {CPSCUser}
{CPSCPassword} {ActiveMQServer} {ActiveMQPort} {ActiveMQUser}
{ActiveMQPassword} {WildflyConfigPath} {RequestCenterWarPath} {QueueSuffix} {isWildfly18}
```

QueueSuffix は、特定の PSC インスタンスに使用されるキューを他の PSC インスタンスと区別するためのオプション パラメータです。このパラメータは、キューの名前にサフィックスを追加します。

```
./jmsrunner.sh Oracle localhost 1521 XE RCDB RC 192.168.174.1 61616 admin admin
"/Users/user/_svn_repo/thirdparty/org/wildfly/wildfly-dist/10.1.0.Final/wildfly-10.1.0.Final_oracle/CPSC-12-1-0/configuration/standalone-full.xml" "/Users/user/_svn_repo/thirdparty/org/wildfly/wildfly-dist/10.1.0.Final/wildfly-10.1.0.Final_oracle/CPSC-12-1-0/deployments/RequestCenter.war"
```

## Windows の場合

### Microsoft SQLServer の使用 :

```
jmsrunner.cmd SqlServer <CPSCServer> <CPSCPort> <CPSCDatabaseName>
<CPSCUser> <CPSCPassword> <ActiveMQServer> <ActiveMQPort> <ActiveMQUser>
<ActiveMQPassword> <WildflyConfigPath> <RequestCenterWarPath> <QueueSuffix> <isWildfly18>
```

### Oracle の使用

```
jmsrunner.cmd Oracle <CPSCServer> <CPSCPort> <CPSCSID> <CPSCUser>
<CPSCPassword> <ActiveMQServer> <ActiveMQPort> <ActiveMQUser>
<ActiveMQPassword> <WildflyConfigPath> <RequestCenterWarPath> <QueueSuffix> <isWildfly18>
jmsrunner.cmd Oracle localhost 1521 XE RCDB RC 192.168.174.1 61616 admin admin
Z:\user\_svn_repo\thirdparty\org\wildfly\wildfly-dist\10.1.0.Final\wildfly-10.1.0.Final_oracle\CPSC-12-1-0\configuration\standalone-full.xml Z:\user\_svn_repo\thirdparty\org\wildfly\wildfly-dist\10.1.0.Final\wildfly-10.1.0.Final_oracle\CPSC-12-1-0\deployments\RequestCenter.war
```

## 注記

- Kek ファイルは、パッチ パスの現在のデータベースに対応する適切な値で更新する必要があります (psc-12.1-patch-main-614/database/classes/config/kek\_new.txt, psc-12.1-patch-main-614/database/classes/config/kek\_old.txt)
- RequestCenter.war パスは、`beeeCamelContext.xml` および `rcjms.properties` を変更するために必要です。
- キューがすでにサフィックス付きで作成されており、サフィックスを削除してデフォルトのままにする場合は、コマンドの最後に `noSuffix` 引数をパスします。
- `servicelink` を使用せずに Wildfly18 アプリケーション サーバを設定する場合は、値を `true` にするために引数「`isWildfly18`」を追加して同じくアップグレードします。このパラメータは、Wildfly10 の設定ではオプションです。

## Requestcenter への Active MQ Artemis サーバの設定

前提条件：アクティブ：MQ Artemis Server が稼働中で、PSC 12.1 の最新パッチが利用可能である必要があります。

### スタンドアロン手順：

1. PSC サーバを停止し、db インストーラを実行し、RequestCenter で最新の WAR ファイルを解凍します。
2. JMSruneer スクリプトを実行して、Active MQ ARTMEIS サーバを PSC に設定します。以下の例で検索

```
例- jmsrunner.cmd SQLSERVER 10.78.0.166 1433 VM184_RCDB_121GA CPSCUser RC 10.78.0.249 61616 admin admin
C:\CiscoPrimeServiceCatalog_121GAPatchV7\wildfly-10.1.0.Final\ServiceCatalogServer\configuration\standalone-full.xml
C:\CiscoPrimeServiceCatalog_121GAPatchV7\wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war
Winqueue
```

3. 上記のスクリプトから：standalone-full.xml、rcjms、および beecameltext ファイルが変更されます（ユーザは成功メッセージを取得する必要があります）
4. CiscoPrimeServiceCatalog 12.1/bin フォルダに移動し、startServiceCatalog.cmd を実行します。
5. サーバが稼働してから、PSC にアクセスしてサービス リンクに移動すると、赤くなるはずですが（サービス リンクが停止していることを意味します）。
6. 任意のサービスを注文します。ユーザは問題なく注文できます。

### クラスタの手順：

1. Db インストーラを実行し、RequestCenter で最新の WAR ファイルを解凍します。
2. JMSruneer スクリプトを実行して、Active MQ ARTMEIS サーバを PSC に設定します。以下の例を参照してください（注：クラスタ ユーザがファイルを変更するためにフォルダを抽出する必要がある場合、実行中の事前カスタマイズ済みスクリプトで RC.war が解凍され、フォルダが tmp ディレクトリに配置されます。同じ場所を RequestCenter.war ファイルとして使用する必要があります）

```
例： jmsrunner.cmd SQLSERVER 10.78.0.166 1433 VM184_RCDB_121GA CPSCUser RC 10.78.0.249 61616 admin admin
C:\CiscoPrimeServiceCatalog_121GAPatchV7\wildfly-10.1.0.Final\domain\configuration\domain.xml
C:\CiscoPrimeServiceCatalog_121GAPatchV7\tmp\RequestCenter.war
```

3. 上記のスクリプトから：Domain.xml、rcjms、および beecamelcontext ファイルが変更されます（ユーザは成功メッセージを取得する必要があります）
4. CiscoPrimeServiceCatalog 12.1/bin フォルダに移動し、パッチが完了するまでカスタマイズ後のスクリプトを実行します。
5. ユーザが複数のノードを持っている場合は、各ノードで apply-customization スクリプトを実行する必要があります。
6. サーバが稼働してから、PSC にアクセスしてサービス リンクに移動すると、赤くなるはずですが（サービス リンクが停止していることを意味します）。
7. 任意のサービスを注文します。ユーザは問題なく注文できます。

## Weblogic で SL なしの RC の設定

WebLogic のパッチ適用セクションの Weblogic パッチ手順に従います。

1. RequestCenter のみを展開します。
2. サービス リンク アプリケーションは展開しないでください。
3. JMS サーバを指している管理対象サーバが稼働している必要があります。（必須）
4. RCが展開されました。
5. システムにログインします。
6. サービス リンクのステータスが赤になっていることを確認します。
7. サービスを注文します。
8. ユーザは注文する必要があります。

## CCS 5.0.1 AO 統合

AO を PSC と統合するには、AO\_PSC\_Integration\_Workflow を使用します。このワークフローは、PSC からの AMQP イベントに基づいてトリガーされます。このワークフローを設定する手順は次のとおりです。

### AO PSC 統合ワークフローを設定する手順

1. [https://github.com/cisco/ActionOrchestratorContent/tree/master/workflow-examples/PSCAOIntegrationWorkflow\\_definition\\_workflow\\_016QL80Y3D2J62KI5hebU1SiGXzunNcCkQj](https://github.com/cisco/ActionOrchestratorContent/tree/master/workflow-examples/PSCAOIntegrationWorkflow_definition_workflow_016QL80Y3D2J62KI5hebU1SiGXzunNcCkQj) からのワークフローをインポートします。



2. 適切なクレデンシャルで AO に AMQP ターゲットを作成します。
3. AO で AMQP イベントを作成し、キュー名に PSC アウトバウンドキューの名前を使用します。
4. ステップ 3 で作成したイベントを使用して AO\_PSC\_Integration\_Workflow にトリガーを追加します。
5. 「Get input variable」アクティビティのソース クエリ json を変更します。
  - a. Trigger> Name\_of\_trigger\_created\_in\_step4> Output> MessageBody
6. 必要に応じて、「AOToPSCExchange」変数の値を変更します。
7. 必要に応じて、「AmqpRoutingKey」変数の値を、ステップ 6 で設定した変換に対応するルーティング キーで変更します。
8. すべての「Publish AMQP Message」アクティビティ（4 アクティビティ）について、ステップ 2 で作成したターゲットを選択します。
9. ワークフローをインポートする env に CloudCenter エンドポイント ターゲットを作成します。
10. ステップ 9 で作成したターゲットを、スイート内部ターゲットではなく、すべての「汎用 CCS API 要求」（3 アクティビティ）アクティビティに追加します。

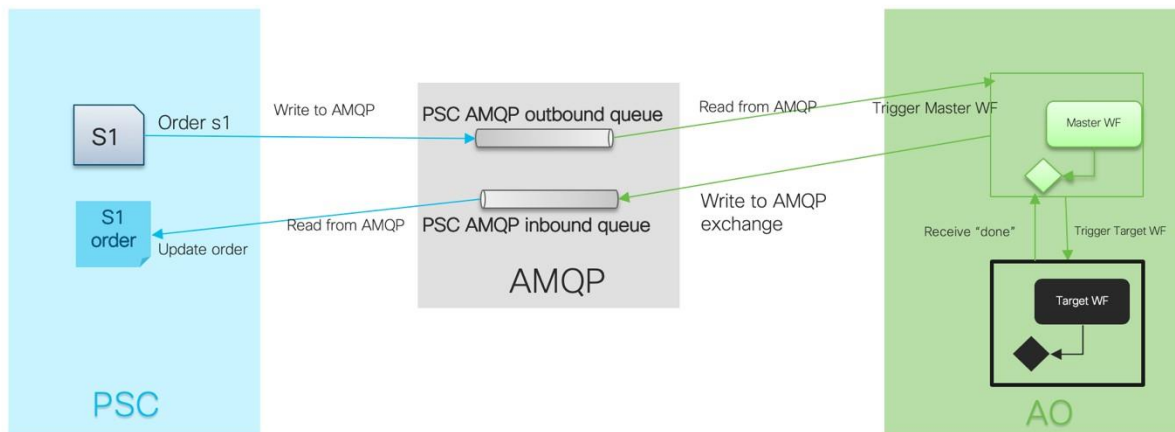
**注：**インポートされたワークフローで設定された出力変数への変更はすべて検出され、更新された値は次の形式で AMQP nsxml メッセージを介して PSC に送信されます。出力変数に XML 固有の文字を含めることはできません。そうしないと、nsxml が無効になります。

```
<message channel-id="D06AE3C8-CBEB-4F03-8D4E-6EACF074BA92"><add-comments><comment>ワークフローが正常に完了しました</comment></add-comments></message>
```

```
出力変数：#OUTPUT_VARIABLE_VALUE_GOES_HERE# </comment></add-comments><take-action action="done"></take-action></message>
```

## トラブルシューティング

1. PSC からサービスを注文した後にワークフローがトリガーされない場合：
  - a. 発信キューに AMQP のコンシューマがあるかどうかを確認します。
    - i. コンシューマが存在しない場合は、トリガー AO\_PSC\_Integration\_Workflow が開始ポーリング状態であるかどうかを確認します。開始状態でない場合は、ワークフローを更新し、ワークフローが更新されるのを待ちます。



## PSC への AO 接続の追加

**前提条件：**AMQP 接続を作成し、インバウンドキューを AMQP システムで作成する必要があります。CCS 5.0.1 (Cloud Center Suite) が稼働中であり、AO が CCS 5.0.1 セットアップの一部である必要があります。マスター ワークフローを AO にインポートする必要があります (上記のセクションを参照)。PSC が最新のパッチ v10 で稼働している必要があります。

AO で PSC\_SERVICE カテゴリを作成し、PSC アプリケーションに関連するワークフローをこのカテゴリに含める必要があります。

### PSC で AO 接続を追加する手順

1. PSC の統合モジュールに移動し、[新規統合 (New Integration)] をクリックして、[CCS アクション オーケストレーション (CCS Action Orchestrator)] 統合タイプを選択します。  
ユーザ：統合管理者ユーザまたは管理者ユーザは接続を作成できます

Configure Integration

Integrations / Integration Types / Configure Integration - CCS Action Orchestrator

Identifier

Name \*

Host Name or IP Address \*

Protocol

Port

User Name \*

Password \*

Categories \*

AMQP Connection \*

Enable Poller

- ID：固有 ID
- 名前：統合の名前
- ホスト名または IP アドレス：CC501 スイートがホストされている IP
- プロトコル：http/https
- ポート：CC501 スイートのポート
- ユーザ名：CC501 ユーザにはテナント管理者グループとスイート管理者グループが含まれている必要があります。
- パスワード：システムのパスワード
- カテゴリ：これは PSC\_SERVICE です（読み取り専用）
- AMQP 接続：関連する接続を選択する必要があります
- ポーラーの有効化：これは、Newscale のポーラー設定に基づいて同期するポーラーで、プロパティをサポートします（ユーザが以下を有効にする場合は、プロパティを Newscale とサポートの一部にする必要があります）。

### newscale.properties に追加するプロパティ

```
#AO ポーラー
#####
#Cron Expression wakes up poller every 10 minutes of an hour
ao.poller.cron=0 0/5 * * * ?
#Cron Expression wakes up health check for 10th min and 15 mins thereafter of the hour ex : 10,25,40,55
minutes
ao.poller.health.check.cron=0 10/15 * * * ?
#High Availability Health checks threshold, this should be greater than Poller cron time specified in minutes
ao.healthCheck.threshold=91
#####
```

### support.properties に追加するプロパティ

```
##### AO Poller Settings #####
#Flag to enable/disable the poller
ao.poller.enable=true
```

2. 統合の作成と接続が正常に保存されます。
3. [ワークフローのインポート (Import workflows)] をクリックします（ワークフローの数によっては、ワークフローのインポートに時間がかかります）。
4. PSC にインポートされたワークフローの数を示すプロンプトが表示されます。

5. ワークフローは **PSC** でサービスとして変換され、**CCS AO** サービス (ID 名) の下の **SD** で使用可能になり、同じディクショナリ グループと **AFC** が **AO** サービス用に作成されます。
6. 統合の管理：ワークフローとサービスは、次のスクリーンショットのように表示されます
7. **AO**で指定されたテナント名は**AO\_getToken.ftl**ファイルで設定する必要があります。例：`tenantName " : " cisco "`  
このファイルは、**RequestCenter** と **ISEE** の両方の展開にあります。

WEB-INF/classes/config/cloud/AO\_getToken.ftl

Host Name or IP Address	172.23.14.128
Protocol	https
Port	34128
User Name	
Password	
Categories	PSC_SERVICE
AMQP Connection	AO_AMQP(AMQ)
Enable Poller	true
STATUS	ACTIVE

Discovered SERVICES WORKFLOWS

Action Orchestrator Services RE-IMPORT

<input type="checkbox"/>	NAME	DESCRIPTION
<input type="checkbox"/>	an issue workflow (AO)	an issue workflow (AO)
<input type="checkbox"/>	Attach Volume (AO)	Attach Volume (AO)
<input type="checkbox"/>	Calculate date for reimport (AO)	Calculate date for reimport (AO)

ユーザはこれらのサービスを **PSC** から注文でき、**AMQP** タスクによって処理され、結果がコメント履歴で **PSC** に対して更新されます。

**PSC** が注文されると、メッセージは **AMQP** キューを介して処理され、ワークフローは **AO** で実行され、処理された結果が **AMQP** キューに返送され、**PSC** が **AMQP** キューから応答を読み取り、同じものがコメントおよび **[履歴 (History)]** セクションで個別の要求に対して更新されます。

### AO 管理統合のその他の機能：

**[Mini Service Designer]**：この機能は、イメージ、プレゼンテーションの詳細、カテゴリ、ファセットの追加など、インポートされたサービスを最小 **SD** で設計するのに役立ちます

an issue workflow (AO) BROWSE IN STORE →

[Integrations](#) / [Manage Integration](#) / an issue workflow (AO)

Details SAVE

Service Name

Description

Categories  AO\_WORKFLOWS(AO)

**[再インポート ( Reimport ) ]** : ユーザが AO のワークフローを変更した場合、手動でインポートしたりポーラー トリガーを待機したりするのではなく、この機能のみを使用して再インポートできます

The screenshot shows the configuration details for an Action Orchestrator (AO) service. The configuration includes:

- Port: 34128
- User Name: (empty)
- Password: (empty)
- Categories: PSC\_SERVICE
- AMQP Connection: AO\_AMQP(AMQ)
- Enable Poller: true
- STATUS: ACTIVE

Below the configuration, there is a "Discovered" section with tabs for "SERVICES" and "WORKFLOWS". Under "Action Orchestrator Services", there is a "RE-IMPORT" button and a table of discovered services:

<input type="checkbox"/>	NAME	DESCRIPTION
<input checked="" type="checkbox"/>	an issue workflow (AO)	an issue workflow (AO)
<input type="checkbox"/>	Attach Volume (AO)	Attach Volume (AO)
<input type="checkbox"/>	Calculate date for reimport (AO)	Calculate date for reimport (AO)
<input type="checkbox"/>	CCS activity test (AO)	CCS activity test (AO)
<input type="checkbox"/>	Convert to Upper Case Latest (AO)	Convert to Upper Case Latest (AO)

**[ログの表示 ( Show Log ) ]** : インポート アクションの最新の 3 つのログが記録されます

The screenshot shows the "Show Log" dialog box for an integration. The dialog displays the following information:

- Logs
- Past Time: 05/01/2019 9:39 PM
- Log: AO Workflows sync process started at 2019-05-02 18:20:00

The background shows the "Integrations" page with a search bar and a table of integrations:

SHORT NAME	NAME
AO	SAN_AO
AMQ	AO_AMQP

**[接続のテスト ( Test Connection ) ]** : 統合の接続をテストします

**[削除 ( Remove ) ]** : PSC から接続を削除します。サービス、AFC、およびディクショナリ グループが削除されます。接続サービスの削除時に消費されたサービスがある場合は、廃止としてマークされます

SHORT NAME	NAME	SYSTEM	USERNAME	URL
AO	SAN_AO	CCS Action Orchestrator	vspancha@cisco.com	https://172.23.14.128:34128
AMQ	AO_AMQP	RabbitMQ Server	admin	http://10.78.0.248:5672

## Service Catalog のレスポンス UI

PSC は、モバイル用のレスポンス UI をサポートしています。ログイン後、ユーザはデフォルトで ServiceCatalogMobileWebsite にリダイレクトされます。

### カスタム Web サイト設定

Website for Service Catalog

Desktop



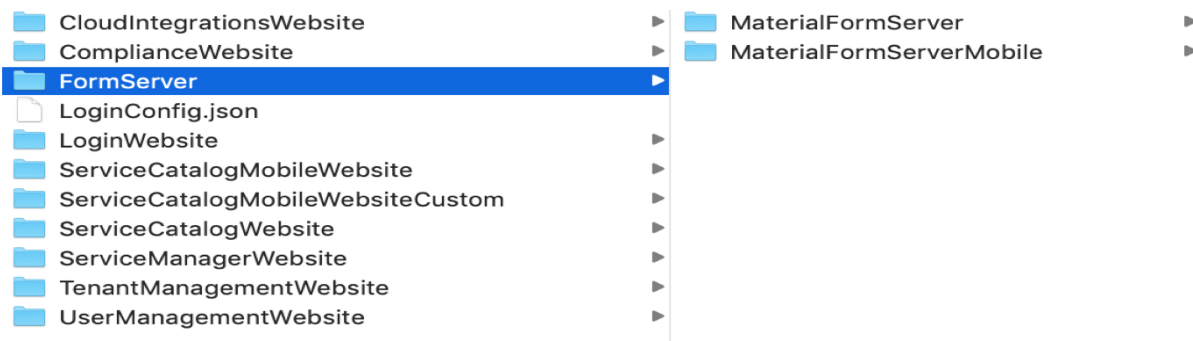

Mobile




参照元 : <http://rch-plstech-001.cisco.com/prrq-docrev/Review/ShowMarkup.aspx?documentid=286716>

### レスポンス UI のカスタマイズ

- ServiceCatalogMobileWebsite は、新しいページ レイアウトのモバイル デバイス用に特別に設計された新しい Web サイトです。
- このモバイル Web サイトは、次の URL から完全にカスタマイズできます。
  - Web サイト ベース URL : /RequestCenter/website/ServiceCatalogMobileWebsite/application
  - CSS ファイル URL : /RequestCenter/website/ServiceCatalogMobileWebsite/common/css/ngc-bootstrap-mobile.css
- このモバイル Web サイトには、次のようなサービスを注文するための独自のフォーム サーバがあります。
  - フォーム サーバの URL : /RequestCenter/website/FormServer/MaterialFormServerMobile



モバイル用の **ResponsiveUI** はマテリアル サービスをサポートします。  
 テスト済みのさまざまなデバイスの解決フォームの詳細を以下に示します。

	Android	IOS
ビューポート	360*640	375*667
解像度	1080*1920	750*1334

表示されるさまざまなプロファイルは次のとおりです。

- 情報および設定タブを表示します。
- **ResponsiveUI**にサービス項目ページが表示されません。
- ハンバーガー メニューの注文、承認、およびカテゴリが表示されます。

パッチ v10 の認可ページでは、クリックすると、承認タブに顧客、注文者、実行者に関する詳細が表示されるようになりました。

## 個人プロフィール API

### 現在のユーザ情報を取得する API

- メソッド : GET
- 要求 URL : <http://localhost:8080/RequestCenter/nsapi/directory/people/currentuser>
- Response -

```
{
  "person":{
    "name":"admin admin",
    "firstName":"admin",
    "lastName":"admin",
    "personId":1,
    "recordStatId":1,
    "email":"internal@newscale.com",
    "timeZoneId":256,
    "timeZoneName":"America/Tijuana",
    "timeZoneDisplayName":"<s ID='13576' />",
    "homeOrganizationalUnitId":1,
    "homeOrganizationalUnitName":"Site Administration",
    "login":"admin",
    "isLdapEnabled":true,
    "isLdapEnabledProperty":true,
    "localeId":1,
    "languageCode":"USEN",
    "languageName":"US English",
    "statusId":1,
  }
}
```

```

"status":"Active",
"supervisorId":0,
"supervisorName": "",
"preferences":{
  "preferenceld":1,
  "shortDateSeperator":"/",
  "shortDateFormat":"MM/dd/yyyy",
  "longDateFormat":"MMMM dd, yyyy",
  "loginModuleId":21,
  "loginModule":"Service Catalog",
  "serviceManagerView":"My Work",
  "serviceManagerStatus":"All Ongoing",
  "timeFormat":"h:mm aa",
  "viewAuthorizationsPortlet":"true",
  "viewMyServiceItemsPortlet":"true",
  "nextGenerationCatalog":"true",
  "authorizationDelegateld":0
},
"teamCount":0,
"taskCount":0,
"contacts":[
  {
    "contactId":1,
    "contactTypeld":1,
    "contactType":"Email",
    "value":"internal@newscale.com"
  }
],
"addresses":[
  {
    "addressId":1,
    "addressTypeld":1,
    "addressType":"Company Address",
    "state":"",
    "zip":"",
    "city":"",
    "country":"",
    "street1":"",
    "street2":"",
    "building":"",
    "buildingLevel":"",
    "cubicle":"",
    "office":""
  },
  . . . .
],
"includedGroups":[

],
"includedOrganizationalUnits":[
  {
    "id":1,
    "name":"Site Administration"
  },
  .....

  {
    "id":387,
    "name":["SQL Server] SG - Tenant-Information Reserved AFC"
  }

```

5

```

    }
  ],
  "personURL": "<a
href=/RequestCenter/organizationdesigner/scnavigate.do?displayRecordId=1&id=1&query=search&resetBN=true&formAction
=display&displayRec=Y&forwardPage=people&forwardTo=showGeneralSuccess&mdicontentPortlet=portlet.sc.person.general
&mdicomponentsPortlet=portlet.sc.person.mdi&selectMDI=sc.person.general&isCreate=Y&sFId=Y&layout=popup_p' on-
click='return GB_showFullScreen('Person', this.href)'\>admin admin</a>",
  "personURLOn-
ly": "/RequestCenter/organizationdesigner/scnavigate.do?displayRecordId=1&id=1&query=search&resetBN=true&formAction
=display&displayRec=Y&forwardPage=people&forwardTo=showGeneralSuccess&mdicontentPortlet=portlet.sc.person.gener
al&mdicomponentsPortlet=portlet.sc.person.mdi&selectMDI=sc.person.general&isCreate=Y&sFId=Y",
  "includedAccounts": {
    },
  "permissions": {
    "showTeams": true,
    "accessProfile": true,
    "accessMyStuff": true,
    "accessIntegration": true
  }
}
}
}

```

## 最終ログイン情報を取得する API

- メソッド : GET
- 要求 URL : [http://localhost:8080/RequestCenter/nsapi/directory/lastlogininfo?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&\\_=1586355716730](http://localhost:8080/RequestCenter/nsapi/directory/lastlogininfo?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&_=1586355716730)
- Response -

```

{
  "List": [
    {
      "lastLoginStatus": "0",
      "lastLoginHost": "127.0.0.1",
      "lastLoginDate": "Thu Apr 16 15:37:24 IST 2020"
    }
  ]
}

```

## 使用可能なタイムゾーンをリストする API

- メソッド : GET
- 要求 URL : [http://localhost:8080/RequestCenter/nsapi/directory/people/timezones?responseType=json&\\_=1586355716731](http://localhost:8080/RequestCenter/nsapi/directory/people/timezones?responseType=json&_=1586355716731)
- Response -

```

{
  "List": [
    {

```



```

    "timeZoneID":201,
    "displayName":"(GMT) Greenwich Mean Time",
    "name":"Etc/Greenwich",
    "offset":"GMT+0:00"
  },
  {
    "timeZoneID":202,
    "displayName":"(GMT) Casablanca, Monrovia",
    "name":"Africa/Casablanca",
    "offset":"GMT+0:00"
  },
  . . .
  {
    "timeZoneID":285,
    "displayName":"(GMT) Western European Time : Dublin, Edinburgh, Lisbon, London",
    "name":"Europe/Lisbon",
    "offset":"GMT+0:00"
  }
]
}

```

## ロケールをリストする API

- メソッド : GET
- 要求 URL : [http://localhost:8080/RequestCenter/nsapi/directory/people/locales?responseType=json&\\_=1586355716732](http://localhost:8080/RequestCenter/nsapi/directory/people/locales?responseType=json&_=1586355716732)
- Response -

```

{
  "List":[
    {
      "localeID":1,
      "localeCode":"USEN",
      "name":"US English",
      "active":0
    }
  ]
}

```

## カレンダー稼働時間のタイムゾーンの API

- メソッド : GET
- 要求 URL : [http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/calendar/timeschedule?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&\\_=1586355716733](http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/calendar/timeschedule?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&_=1586355716733)
- Response -

```

{
  "personCalendar":{

```

```
"timeZoneName": "(GMT-08:00) 太平洋 標準時 (米国 および カナダ) 、 Tijuana",
"localTime": "04/17/2020",
"timeScheduleEntry": [
  {
    "id": 8,
    "objectId": 2,
    "entityId": 1,
    "dayId": 1,
    "fromDate": "12:00 AM",
    "toDate": "12:00 PM",
    "tenantId": 1
  },
  {
    "id": 9,
    "objectId": 2,
    "entityId": 1,
    "dayId": 2,
    "fromDate": "09:00 AM",
    "toDate": "5:00 PM",
    "tenantId": 1
  },
  {
    "id": 10,
    "objectId": 2,
    "entityId": 1,
    "dayId": 3,
    "fromDate": "09:00 AM",
    "toDate": "5:00 PM",
    "tenantId": 1
  },
  {
    "id": 11,
    "objectId": 2,
    "entityId": 1,
    "dayId": 4,
    "fromDate": "09:00 AM",
    "toDate": "5:00 PM",
    "tenantId": 1
  },
  {
    "id": 12,
    "objectId": 2,
    "entityId": 1,
    "dayId": 5,
    "fromDate": "09:00 AM",
    "toDate": "5:00 PM",
    "tenantId": 1
  },
  {
    "id": 13,
    "objectId": 2,
    "entityId": 1,
    "dayId": 6,
    "fromDate": "09:00 AM",
    "toDate": "5:00 PM",
  }
]
```

```

    "tenantId":1
  },
  {
    "id":14,
    "objectId":2,
    "entityId":1,
    "dayId":7,
    "fromDate":"12:00 AM",
    "toDate":"12:00 PM",
    "tenantId":1
  }
]
}
}

```

## ユーザが特定のカレンダーにエントリするための API

- メソッド : GET
- 要求 URL :  
http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/calendar/entries?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&\_=1586355716734
- 応答 :

```

{
  "personCalendar":{
    "calendarEntry":[
      {
        "id":1,
        "objectId":2,
        "entityId":1,
        "name":"Labour day",
        "dayTypeId":2,
        "stringCalendarDate":"05/01/2020"
      }
    ]
  }
}

```

## 初期設定

- メソッド : GET
- 要求URL :  
http://localhost:8080/RequestCenter/nsapi/directory/people/currentuser?responseType=json&\_=1586355716742 -前述
- メソッド : GET
- 要求 URL :  
http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/information/modules?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&\_=1586355716743
- 応答 :

```

{
  "List":[

```

```

{
  "moduleId":7,
  "moduleName":"Administration",
  "logicName":"administration"
},
{
  "moduleId":15,
  "moduleName":"Advanced Reporting",
  "logicName":"advancedreporting"
},
...
{
  "moduleId":27,
  "moduleName":"User Management",
  "logicName":"usermanagement"
}
]
}

```

- メソッド : GET
- 要求 URL : [http://localhost:8080/RequestCenter/nsapi/directory/1/smviews?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&\\_=1586355716744](http://localhost:8080/RequestCenter/nsapi/directory/1/smviews?utid=0E500219AC7B988801F3D3CB813BCAAE&responseType=json&_=1586355716744)
- 応答 :

```

{
  "List":[
    {
      "display":true,
      "isActiveView":false,
      "isGlobal":true,
      "isGlobalSearch":false,
      "isPublic":false,
      "order":5,
      "viewId":"Administration",
      "viewName":"Administration",
      "datatableName":"ServiceMgrStaticViews",
      "filters":[

    ]
  },
  {
    "display":false,
    "isActiveView":false,
    "isGlobal":true,
    "isGlobalSearch":false,
    "isPublic":false,
    "order":100,
    "viewId":"DueTomorrow",
    "viewName":"Due Tomorrow",
    "datatableName":"ServiceMgrStaticViews",
    "filters":[

```

```

    ]
  },
  ...
  {
    "display":true,
    "isActiveView":false,
    "isGlobal":true,
    "isGlobalSearch":false,
    "isPublic":false,
    "order":1,
    "viewId":"AvailableWork",
    "viewName":"Available Work",
    "datatableName":"ServiceMgrStaticViews",
    "filters":[

  ]
},
{
  "display":true,
  "isActiveView":false,
  "isGlobal":true,
  "isGlobalSearch":false,
  "isPublic":false,
  "order":6,
  "viewId":"MyRequisition",
  "viewName":"Recent Requisitions",
  "datatableName":"ServiceMgrRequisitionView",
  "filters":[

  ]
}
]
}
}

```

## 権限ページを取得する API

- メソッド : GET
- 要求 URL :  
http://localhost:8080/RequestCenter/nsapi/directory/people/currentuser?responseType=json&\_=1586355716755-前述
- メソッド : GET
- 要求 URL :  
http://localhost:8080/RequestCenter/nsapi/directory/people/1/roles/capabilities?utid=0E500219AC7B988801F3D3CB813BCAAE&roleNameLike=&responseType=json&\_=1586355716756
- 応答 :

```

{
  "List":[
    {
      "0 Service Icons":[

```

```
{
  "systemCapabilityId":80,
  "moduleId":18,
  "name":"Access Service Item Instance Data",
  "inherited":false,
  "moduleName":"Service Item Manager",
  "logicName":"access_service_item_data"
},
{
  "systemCapabilityId":79,
  "moduleId":18,
  "name":"Access Service Item Definition",
  "inherited":false,
  "moduleName":"Service Item Manager",
  "logicName":"access_service_item_definition"
},
...
{
  "systemCapabilityId":66,
  "moduleId":17,
  "name":"NSAPI Access",
  "inherited":false,
  "moduleName":"Web Services",
  "logicName":"web_services_nsapi_access"
}
],
"1 Private Cloud IaaS Group":[
  {
    "systemCapabilityId":80,
    "moduleId":18,
    "name":"Access Service Item Instance Data",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_data"
  },
  {
    "systemCapabilityId":79,
    "moduleId":18,
    "name":"Access Service Item Definition",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_definition"
  },
  ...
  {
    "systemCapabilityId":66,
    "moduleId":17,
    "name":"NSAPI Access",
    "inherited":false,
    "moduleName":"Web Services",
    "logicName":"web_services_nsapi_access"
  }
],
"2 Unified Communication Group":[
  {
    "systemCapabilityId":80,
    "moduleId":18,
    "name":"Access Service Item Instance Data",
    "inherited":false,
    "moduleName":"Service Item Manager",
```

```
    "logicName":"access_service_item_data"
  },
  {
    "systemCapabilityId":79,
    "moduleId":18,
    "name":"Access Service Item Definition",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_definition"
  },
  ...

  {
    "systemCapabilityId":66,
    "moduleId":17,
    "name":"NSAPI Access",
    "inherited":false,
    "moduleName":"Web Services",
    "logicName":"web_services_nsapi_access"
  }
],
"3 Unified Workplace Demo":[
  {
    "systemCapabilityId":80,
    "moduleId":18,
    "name":"Access Service Item Instance Data",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_data"
  },
  {
    "systemCapabilityId":79,
    "moduleId":18,
    "name":"Access Service Item Definition",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_definition"
  },
  ...

  {
    "systemCapabilityId":66,
    "moduleId":17,
    "name":"NSAPI Access",
    "inherited":false,
    "moduleName":"Web Services",
    "logicName":"web_services_nsapi_access"
  }
],
"4 BYOD / Mobility Group":[
  {
    "systemCapabilityId":80,
    "moduleId":18,
    "name":"Access Service Item Instance Data",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_data"
  },
  {
    "systemCapabilityId":79,
```

```

    "moduleId":18,
    "name":"Access Service Item Definition",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_definition"
  },
  ...
  {
    "systemCapabilityId":66,
    "moduleId":17,
    "name":"NSAPI Access",
    "inherited":false,
    "moduleName":"Web Services",
    "logicName":"web_services_nsapi_access"
  }
],

```

...

```

"[SQL Server] SG - Tenant-Information Reserved AFC":[
  {
    "systemCapabilityId":80,
    "moduleId":18,
    "name":"Access Service Item Instance Data",
    "inherited":false,
    "moduleName":"Service Item Manager",
    "logicName":"access_service_item_data"
  },
  ...
  {
    "systemCapabilityId":66,
    "moduleId":17,
    "name":"NSAPI Access",
    "inherited":false,
    "moduleName":"Web Services",
    "logicName":"web_services_nsapi_access"
  }
]
}
]
}

```

- メソッド : GET
- 要求 URL :  
[http://localhost:8080/RequestCenter/nsapi/directory/people/1/roles/capabilities?utid=F699F3DBF83E7D23E126004E46246C1F&roleNameLike={roleName}&responseType=json&\\_=1586750555449](http://localhost:8080/RequestCenter/nsapi/directory/people/1/roles/capabilities?utid=F699F3DBF83E7D23E126004E46246C1F&roleNameLike={roleName}&responseType=json&_=1586750555449)
- 例 :  
[http://localhost:8080/RequestCenter/nsapi/directory/people/1/roles/capabilities?utid=83BEDED575D568A017ECC1E5E4B0BFFF&roleNameLike=Cloud%20Center%20Reserved%20Services&responseType=json&\\_=1587123121426](http://localhost:8080/RequestCenter/nsapi/directory/people/1/roles/capabilities?utid=83BEDED575D568A017ECC1E5E4B0BFFF&roleNameLike=Cloud%20Center%20Reserved%20Services&responseType=json&_=1587123121426)
- 応答



```

{
  "List":[
    {
      "Cloud Center Reserved Services":[
        {
          "systemCapabilityId":80,
          "moduleId":18,
          "name":"Access Service Item Instance Data",
          "inherited":false,
          "moduleName":"Service Item Manager",
          "logicName":"access_service_item_data"
        },
        ...
        {
          "systemCapabilityId":66,
          "moduleId":17,
          "name":"NSAPI Access",
          "inherited":false,
          "moduleName":"Web Services",
          "logicName":"web_services_nsapi_access"
        }
      ]
    }
  ]
}

```

## ユーザの更新

- 要求URL :  
<http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/information?utid=CA54557D8001B972743F195AA73568AD&responseType=json>
- メソッド : PUT
- ペイロード :

```

{
  "person":{
    "firstName":"admin",
    "lastName":"admin",
    "timeZoneId":"256",
    "homeOrganizationalUnitId":1,
    "login":"admin",
    "localeId":"1",
    "statusId":1,
    "contacts":[
      {

```

```

        "contactId":1,
        "contactTypeId":1,
        "contactType":"Email",
        "value":"internal@newscale.com"
    }
],
"addresses":[
    {
        "addressId":1,
        "addressTypeId":1,
        "addressType":"Company Address",
        "state":"",
        "zip":"",
        "city":"",
        "country":"",
        "street1":"",
        "street2":"",
        "building":"",
        "buildingLevel":"",
        "cubicle":"",
        "office":""
    },
    {
        "addressId":2,
        "addressTypeId":2,
        "addressType":"Personal Address",
        "state":"",
        "zip":"",
        "city":"",
        "country":"",
        "street1":"",
        "street2":""
    }
]
}
}

```

応答 :

```

{
  "status-message":{
    "code":"TA_116",
    "value":"Person Profile updated successfully"
  }
}

```

## カレンダーの更新

- 要求URL :  
<http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/calendar/timeschedule?utid=F699F3DBF83E7D23E126004E46246C1F&responseType=json>
-

- メソッド : PUT
- ペイロード :

```
{
  "personCalendar":{
    "timeZoneName":"(GMT-08:00) 太平洋 標準時 (米国 および カナダ) 、 Tijuana",
    "localTime":"04/12/2020",
    "timeScheduleEntry":[
      {
        "id":8,
        "fromDate":"12:00 PM",
        "toDate":"12:00 PM"
      },
      {
        "id":9,
        "fromDate":"09:00 AM",
        "toDate":"5:00 PM"
      },
      {
        "id":10,
        "fromDate":"09:00 AM",
        "toDate":"5:00 PM"
      },
      {
        "id":11,
        "fromDate":"09:00 AM",
        "toDate":"5:00 PM"
      },
      {
        "id":12,
        "fromDate":"09:00 AM",
        "toDate":"5:00 PM"
      },
      {
        "id":13,
        "fromDate":"09:00 AM",
        "toDate":"5:00 PM"
      },
      {
        "id":14,
        "fromDate":"12:00 PM",
        "toDate":"12:00 PM"
      }
    ]
  }
}
```

応答 :

```
{
```

```

"status-message":{
  "code":"TA_131",
  "value":"Time schedule updated successfully."
}
}

```

## 通常のカレンダーでは使用できない日付を追加する API

- 要求 URL :  
http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/calendar/entries?utid=F699F3DBF83E7D23E126004E46246C1F&responseType=json
- メソッド : PUT
- ペイロード :

```

{
  "personCalendar":{
    "calendarEntry":[
      {
        "name":"勤労感謝の日",
        "dayType":2,
        "stringCalendarDate":"05/01/2020"
      }
    ]
  }
}

```

応答 :

```

{
  "status-message":{
    "code":"TA_131",
    "value":"日程が正常に追加されました。"
  }
}

```

## 更新 : 基本設定

- 要求 URL :  
http://localhost:8080/RequestCenter/nsapi/directory/v2/person/1/preferences?oldShortDateFormat=MM/dd/yyyy&utid=F699F3DBF83E7D23E126004E46246C1F&responseType=json
- メソッド : PUT
- ペイロード : -

```

{
  "person":{
    "preferences":{
      "preferenceld":1,
      "shortDateSeperator":"/",
      "shortDateFormat":"MM/dd/yyyy",
      "longDateFormat":"MMMM dd, yyyy",
      "loginModuleId":21,
      "serviceManager-
View":"datatable.do?datatableFormAction=activateSavedView&viewID=MyWork&datatableID=ServiceMgrStaticViews&retu
rnToURL=%2FRequestCenter%2Fservicemanager%2Fhomepage.do%3FdatatableID%3DServiceMgrStaticViews",
      "serviceManagerStatus":"All Ongoing",
      "timeFormat":"h:mm aa",
      "viewAuthorizationsPortlet":true,
      "viewMyServiceItemsPortlet":true,
      "nextGenerationCatalog":true,
      "authorizationDelegateld":"0",
      "delegationStartDate":"","
      "delegationEndDate":""
    }
  }
}

```

[応答] : -

```

{
  "status-message":{
    "code":"TA_116",
    "value":"Person Profile updated successfully"
  }
}

```

## リソース文字列を取得する API

指定したキーと本文の JSON 形式のリソース文字列 API 値を取得します : [11789,11790]。プレースホルダを param1、param2 .....、param5 などリソース文字列にパスする場合は、JSON 形式 \`"5349 # param1 # param2 # param3 # param4 # param5"`などをパスできます。

例 : [11789,11790,\`"5349#1#100"`]

- メソッド : PUT
- URL : /RequestCenter/nsapi/v1/common/resourcestrings
- ヘッダー : Content-Type : application/json
- ペイロード : ["5349#1#100",63, 123]
- 応答 :
 

```

{
  "63" : "Logout",
  "123" : "Key Customer Project",
  "5349" : "Please enter a number between 1 and 100."
}

```

## AO ワークフローをインポートする API

- メソッド : PUT
- URL : RequestCenter/nsapi/cloud/ao/{shortName}
- 応答メッセージ : 'PSC\_SERVICE'カテゴリの15のAOワークフローが正常にインポートされました。

## AO ワークフロー サービスを再インポートする API

- メソッド : PUT
- URL : RequestCenter/nsapi/cloud/ao/reimport/{shortName}
- 要求本文 : {"List":["SRI::AO::0166CV0K6YBZJ4GufPGfyfTljLaoFFG4Lv::1218"]}
- 応答メッセージ : 再インポートが正常に完了しました

## ログ API を表示する API

### ログ反復のリストを取得する

- メソッド : GET
- URL : RequestCenter/nsapi/cloud/ao/{shortName}/logs
- 応答メッセージ :

```
{
  "Map" : {
    "4" : "4-30-19 9:51 AM",
    "5" : "4-30-19 9:30 PM",
    "6" : "5-3-19 4:17 AM",
    "Fri May 03 11:17:45 2019" : {
      "6" : [
        "AO Workflows sync process started at 2019-04-30 17:39:01",
        "AO Workflows Discovery Done in 0.324 seconds",
        "AO Workflows sync process completed at 2019-04-30 17:39:10"
      ]
    }
  }
}
```

### 反復ログを1個取得

- メソッド : GET
- URL : RequestCenter/nsapi/cloud/ao/{shortName}/logs/{iterationId}
- 応答メッセージ :

```
{
  "Map" : {
    "1" : [
      "AO Workflows sync process started at 2019-04-30 17:39:01",
      "AO Workflows Discovery Done in 0.324 seconds",
      "AO Workflows sync process completed at 2019-04-30 17:39:10"
    ]
  }
}
```

## AO ワークフローのリストを取得する API

- メソッド : GET
- URL : RequestCenter/nsapi/cloud/ao/reimport/{shortName}/workflows
- 応答メッセージ :

```
{
  "WorkFlowServiceList" : {
    "startRow" : 0,
    "totalCount" : 0,
    "recordSize" : 3,
    "workflowServices" : [
      {
        "workFlowName" : "CCS VM Actions",
        "externalId" : "0166CV0K6YBZJ4GufPGfgyfTljLaoFFG4Lv",
        "serviceName" : "CCS VM Actions (SRI)",
        "serviceId" : 1218,
        "createdOn" : 1556606350467,
        "cloudConnectionId" : 30
      },
      {
        "workFlowName" : "Create CCS Deployment for CentOS",
        "externalId" : "0166D3O1A9NTP21PMswoZSFYhIm7EAopYyC",
        "serviceName" : "Create CCS Deployment for CentOS (SRI)",
        "serviceId" : 1217,
        "createdOn" : 1556606347687,
        "cloudConnectionId" : 30
      },
      {
        "workFlowName" : "To UPPER",
        "externalId" : "016BD6EZDHLRG708Shs9arXyWYovZrS0HTe",
        "serviceName" : "To UPPER (SRI)",
        "serviceId" : 189,
        "createdOn" : 1556606345083,
        "cloudConnectionId" : 30
      }
    ]
  }
}
```

## 12.1\_Patch\_v9 で導入

### スタンドアロン Wildfly 向け PSC サービス カタログと Service Manager Web サイト UI 分離

Prime Service Catalog がインストールされているサーバとは別のサーバで、PSC は分離された Service Catalog および Service Manager Web サイトをホストすることをサポートしています。個別にホストされるサービス カタログ Web サイトは、サービス フォームを生成するために、マテリアル フォーム サーバ レンダラ (12.1\_Patch\_v8 で導入) のみを使用できます。

#### 概要

PSC では、RequestCenter が展開されているサーバ (Wildfly) とは異なる Web サーバで、ServiceCatalog および ServiceManager Web サイトのホストをサポートしています。これらの分離された Web サイトは、RequestCenter.war に対して REST API コールを行います。異なる Web サーバ (Nginx、tomcat など) でホストされている場合、ServiceCatalog と ServiceManager は、Web サーバのドメイン名と IP を使用してアクセスできます。ServiceCatalog と ServiceManager のみを使用するエンド ユーザには、分離された Web サイトの URL が提供されます。Web サイトは任意の Web サーバでホストできますが、Nginx および tomcat に対してテストおよび認定されています。

注：

1. ServiceCatalog の分離 UI で、ServiceForm は MaterialUI のみを使用してレンダリングできるため、現時点では機能が制限されています。
2. RequestCenter.war から RequestCenter が展開されるサーバでホストされる ServiceCatalog および ServiceManager の以前の動作は変更されません。

#### Tomcat Web サーバの設定

Windows に Tomcat 9 をインストールします。VM 外部から tomcat にアクセスして管理できるようになります。

IP またはホスト名で tomcat にアクセスできない場合は、次の手順を実行します

1. 以下の場所から server.xml を開きます。
  - a. <Install\_Dir>\Apache Software Foundation\Tomcat 9.0\conf
2. address = "0.0.0.0" をコネクタ文字列に追加します。  
注：0.0.0.0 の代わりに、Web サーバ VM の IP アドレスをマッピングすることもできます。  

```
<Connector port="8080" protocol="HTTP/1.1"
            connectionTimeout = "20000"
            redirectPort = "8443"
            address = "0.0.0.0" ***
```
3. localhost から VM の IP アドレスまたはホスト名で更新する必要がある defaultHost および Hostname の値を変更します。
4. http:<WebServer\_IP or HostName>:<WebServer\_PORT>/- にアクセスします
5. Web サーバのウェルカム ページがロードされます。

Nginx Web サーバに関する注意：ユーザは nginx\_conf ファイルを変更し、サーバ名を VM のホスト名にマッピングする必要があります。Nginx の展開フォルダは HTML フォルダです。

注：RequestCenter が展開されている PSC サーバ (アプリケーション サーバ) と UI 分離システム (Web サーバ) の両方が同じドメインにある必要があります。分離された Web サイトは、ServiceForm の注文または更新のいずれかをサポートします。

#### 別の Web サーバでの PSC UI の設定

1. Tomcat がインストールされている VM を選択し、Web サーバの展開フォルダ (この場合は C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps) に移動します。
2. 次のようなフォルダ構造を作成します
  - a. RequestCenter、Login、Logout フォルダを作成します。
  - b. RequestCenter 内に、website というサブフォルダを作成します。
3. RequestCenterが展開されている「wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website」の PSC サーバから ServiceCatalog、Service Manager、および FormServer フォルダをコピーし、tomcat VM の webapps/RequestCenter/website/ に配置します。
4. wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website>LoginWebsite から LoginWebsite dist フォルダの内容をコピーし、tomcat VM の webapps /またはログインに配置します。
5. wildfly10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website から LoginConfig.Json をコピーし、the webapps/RequestCenter/website に配置します。



6. wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war から help と jsbundle フォルダをコピーし、tomcat vm の webapps\RequestCenter に貼り付けます。
7. wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war\website>LoginWebsite\ssoLogout から ssologout フォルダの内容を、tomcat VM の webapps/Logout フォルダにコピーします。
8. JSON ファイルを変更します。
  - ホスト : RequestCenter が展開されている PSC サーバは、分離された Web サイトに必須です。
  - Cookie ドメイン : RequestCenter が展開されている PSC サーバはドメインに属します。例 : VM が cisco.com に属し、Cookie ドメインが cisco.com である場合。
  - serviceCatalogUrl : ホストされている tomcat サーバの IP またはホスト名であるサービス カタログ URL を指定します。
  - serviceManagerUrl : ホストされている Tomcat サーバの IP またはホスト名であるサービスカタログ URL を指定します。

ユーザが [ログイン (Log in)] ページで以下の文字列のカスタム情報を提供する場合、ここに情報をパスすることができ、カスタム情報は [ログイン (Log in)] ページに反映されます。

  - copyRightText : ログイン ページに表示する著作権メッセージ。
  - displayString : 製品名 (Cisco Service Prime Catalog がデフォルトです)。
  - MarketingVersion : 製品バージョン。

LoginConfig.json の例

```
{
  "host": "http://<Host_Name>:<Port>",
  "cookieDomain": ".cisco.com",
  "ssoUrl": "",
  "loginModule": {
    "serviceCatalogUrl": "http://<Host_Name_of_Tomcat>/RequestCenter/website/ServiceCatalogWebsite/application/index.html",
    "serviceManagerUrl": "http://<Host_Name_of_Tomcat>/RequestCenter/website/ServiceManagerWebsite/application/servicemanager.html?route=serviceManager#homepage"
  },
  "loginPageInfo": {
    "copyRightText": "",
    "displayString": "",
    "marketingVersion": ""
  }
}
```

9. Tomcat サーバを再起動します。
10. 以下のように Tomcat サーバの詳細を入力します  
[http://<IP\\_WebServer>:<WebServer\\_Port>/Login](http://<IP_WebServer>:<WebServer_Port>/Login)

URL によって PSC ログイン ページが開き、ログインすると、ユーザは ServiceCatalog と ServiceManager を確認できます。

### 分離 PSC UI での認証およびセッション管理

PSC UI を分離するための設定には 4 つの領域があります。

1. [PSC DB クレデンシャル](#)
2. [LDAP 認証](#)
3. [LDAP 外部認証](#)
4. [SAML SSO](#)

## PSC DB クレデンシャル

前提条件：RequestCenter が展開されている PSC サーバは、Wildfly アプリケーション サーバがインストールされた VM にインストールし、稼働させる必要があります。PSC DB クレデンシャルで正常にログインできるはずですが、最新パッチが PSC インスタンスにインストールされていることを確認します。

分離機能を備えたシステムを使用するには、次の手順を実行します。

1. RequestCenter が展開されている PSC サーバの **Newscale.properties** に記載されているプロパティを追加します。

```
#####
#decoupled UI domain name
decoupled.cookie.domain=.cisco.com
#####
#decoupled UI authentication mode
decoupled.authentication=true
#####
#decoupled UI landing page url for servicecatalog module
decoupled.servicecatalog.url=http://host_name:port/RequestCenter/website/ServiceCatalogWebsite/application/index.html
#####
#decoupled UI landing page url for servicemanager module
decoupled.servicemanager.url=http://host_name:port/RequestCenter/website/ServiceManagerWebsite/application/
servicemanager.html?route=servicemanager#homepage
#Only if User is using SSO with SAML#####
decoupled.saml.logout.url=http://<Host_Name>:<Port>
2. Standalone_full.xml を変更します。
3. イタリアック体のエントリを追加します。
4. CORS ヘッダーが更新されたら、UI が deployed.ui.cisco.com の FQDN に対してヘッダー値を変更します。
5. ajax 要求のための Standalone_full.xml ファイルへの RequestCenter.war に対する CORS 固有ヘッダー
<subsystem xmlns="urn : jboss:domain:undertow:3.1">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" socket-binding="http" max-post-size="104857600" redirect-socket="https" enable-
    http2="true"/>
    <https-listener name="https" socket-binding="https" security-realm="ApplicationRealm" enable-http2="true"/>
    <host name="default-host" alias="localhost">
      <location name="/" handler="welcome-content"/>
      <filter-ref name="server-header"/>
      <filter-ref name="x-powered-by-header"/>
      <filter-ref name="Access-Control-Allow-Origin"/>
      <filter-ref name="Access-Control-Allow-Headers"/>
      <filter-ref name="Access-Control-Allow-Credentials"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
    <websockets/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="{jboss.home.dir}/welcome-content"/>
  </handlers>
  <filters>
    <response-header name="server-header" header-name="Server" header-value="WildFly/10"/>
    <response-header name="x-powered-by-header" header-name="X-Powered-By" header-
    value="Undertow/1"/>
    <response-header name="Access-Control-Allow-Origin" header-name="Access-Control-Allow-Origin"
    header-value="ui.cisco.com"/>
    <response-header name="Access-Control-Allow-Headers" header-name="Access-Control-Allow-
    Headers" header-value="accept, authorization, content-type, x-requested-with, Content-Type"/>
    <response-header name="Access-Control-Allow-Credentials" header-name="Access-Control-Allow-
    Credentials" header-value="true"/>
  </filters>
</subsystem>
```

**LDAP 認証**

前提条件：LDAP がシステムで設定され、ユーザが Windows 認証で認証できる必要があります。ユーザは RequestCenter が展開されている PSC サーバでシングルサインオンを実行できる必要があります。

分離システムを使用して SSO を機能させるには、次の設定を検索します。

1. 設定については、「[PSC DB クレデンシャルの使用](#)」を参照してください。指定された設定とは別に、IIS の追加設定も行われる必要があります。
2. Web サーバ
  1. LoginConfig.json を変更します。
  2. 次のフラグを true に設定します。
  3. SSOUrl を入力します。
  4. "isWindowAuth":"true",
  5. "ssoUrl":"https://<Host Name of PSC where RequestCenter is deployed >/RequestCenter/sessioninfo.jsp?isDecoupled=true",
  6. IIS がホストされている VM に移動します。
  7. CORS モジュールを <https://www.iis.net/downloads/microsoft/iis-cors-module> からダウンロードします。
  8. CORS モジュールを取り付けます。
  9. IIS の Web.config を開き、system.webserver 内に以下のタグを追加します。

```
<snmp enabled="true">
<add origin="http://vm235.cisco.com" allowCredentials="true" >
  <allowMethods>
    <add method="PUT" />
    <add method="GET" />
    <add method="POST" />
    <add method="DELETE" />
    <add method="OPTIONS" />
  </allowMethods>
</add>
</cors>
<httpProtocol>
  <customHeaders>
    <remove name="X-Powered-By" />
    <add name="Access-Control-Allow-Headers" value="x-requested-with, Content-Type, origin, Authorization, Accept, locale, access_token, utid, Range" />
    <add name="X-Powered-By" value="ASP.NET, Undertow/1" />
  </customHeaders>
</httpProtocol>
```

次に、IIS の web.config の例を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<system.web>
<!-- Default <, >, *, %, &, :, \, ? or %u003c,%u003e,%u002a,%u0025,%u0026,%u003a,%u005c,%u003f -->
  <httpRuntime maxUrlLength="1000" requestPathInvalidCharacters="%u003c,%u003e,%u0025,%u005c,%u003f,%u252F" />
</system.web>
<system.webServer>
  <modules>
    <remove name="WebDAVModule" />
  </modules>
  <handlers>
    <remove name="WebDAV" />
  </handlers>
<security>
  <requestFiltering allowDoubleEscaping="true">
<requestLimits maxAllowedContentLength="104857600" maxQueryString="2048" />
<verbs>
  <add verb="POST" allowed="true" />
  <add verb="PUT" allowed="true" />
  <add verb="DELETE" allowed="true" />
  <add verb="GET" allowed="true" />
```

```

        <add verb="OPTIONS" allowed="true" />
    </verbs>
</requestFiltering>
</security>
<httpErrors>
    <remove statusCode="403" subStatusCode="-1" />
    <error statusCode="403" prefixLanguageFilePath="" path="https://10.78.0.251/RequestCenter" response-
Mode="Redirect" />
</httpErrors>
    <cors enabled="true">
    <add origin="http://vm235.cisco.com" allowCredentials="true" >
        <allowMethods>
            <add method="PUT" />
                <add method="GET" />
            <add method="POST" />
            <add method="DELETE" />
                <add method="OPTIONS" />
        </allowMethods>
    </add>
</cors>
<httpProtocol>
    <customHeaders>
        <remove name="X-Powered-By" />
        <add name="Access-Control-Allow-Headers" value="x-requested-with, Content-Type, origin,
Authorization, Accept, locale, access_token, utid, Range" />
        <add name="X-Powered-By" value="ASP.NET,Undertow/1" />
    </customHeaders>
</httpProtocol>
</system.webServer>
</configuration>

```

ファイルを保存し、IIS Web サーバを再起動して、[分離された URL]を確認します。

### LDAP 外部認証

前提条件：システムで LDAP を設定し、ユーザが外部認証で認証できる必要があります。ユーザは、RequestCenter が展開されている PSC サーバで外部認証を実行できる必要があります。

以下は、分離を使用した外部認証の設定です。

1. 設定については、「[PSC DB クレデンシャルの使用](#)」を参照してください。指定された設定とは別に、LoginConfig.Json での追加設定も行う必要があります。
2. Tomcat 側から
  - a. LoginConfig.json に変更します
  - b. 次のフラグを true に設定します。  
"isWindowAuth": "true",

### SAML SSO

前提条件：SAML SSO を有効にし、RequestCenter が展開されている PSC サーバで必要な設定を行う必要があります。ユーザは SAML 認証でログインできる必要があります。

分離を使用した SAML 認証による SSO 設定を次に示します。

1. 設定については、「[PSC DB クレデンシャルの使用](#)」を参照してください。指定された設定とは別に、LoginConfig.Json での追加設定も行う必要があります。
2. Tomcat 側から
  - a. LoginConfig.json に変更します。  
以下のフラグを true に設定し、SSO URL を指定します。  
"isWindowAuth": "false",  
"ssoUrl": "https://<Host Name of PSC server where RequestCenter is de-  
ployed>/RequestCenter/sessioninfo.jsp?isDecoupled=true",

**注：**すぐに使用できる ServiceCatalog または ServiceManager Web サイトの代わりに CustomWebsite を使用するには、次の手順を実行します。

1. webapps の Webサーバ、または RequestCenter または Web サイトにそれぞれのフォルダを作成します。
2. newscale.properties および LoginConfig.json ファイルにリダイレクト URL を指定します。

#### 認証およびセッション管理中のトラブルシューティング

- システム相対パスからのイメージ URL を使用するサービスは、同じイメージを tomcat サーバに配置する必要があります。
- webapps/RequestCenter で、images というフォルダを作成し、サービス内のイメージ URL に使用されるすべてのイメージを配置します。（これらのイメージは、PSC インスタンス wildfly-10.1.0.Final\ServiceCatalogServer\deployments\RequestCenter.war/images にあります。
- tomcat サーバを再起動し、ログインすると、各サービスのイメージのロードを確認できます。
- RequestCenter が展開されている PSC サーバが IIS Web サーバを指し、分離された UI が別の tomcat サーバまたは他の Web サーバ上にある場合、ユーザが誤ったクレデンシャルを入力してもエラー メッセージは表示されません。この問題を回避するには：
  1. [機能設定ページの詳細 (feature setting page details) ] ラジオ ボタンを編集し、保存します。
  2. IIS サーバを再起動します。  
詳細については、以下のリンクを参照してください。<https://stackoverflow.com/questions/2640526/detailed-500-error-message-asp-iis-7-5>
- IIS では、すべてのエラー コードを詳細に作成し、エラー メッセージを PSC インスタンスからキャプチャして、IIS またはエンドユーザに表示できるようにします。例：412、405、404、406 など

#### パスワードをリセットする API

- API - GET メソッド -
- エンドポイント -

`http://localhost:8080/RequestCenter/nsapi/authentication/login?forgotPassword=true&NSA_LOGIN_NAME=user1`

ヘッダー パラメータ

- Content-Type - application/json

**API の応答は、JSON で次のようになります。**

```
{
  "nsapi-response" : {
    "NSA_FORGOT_PASSWORD_OPERATION_SUCCESS" : "Your Request for a new password is being processed. 9An email has been sent to your email address."
  }
}
```

#### システム準拠のサンプル パスワードを生成する API

- API - GET メソッド -
- エンドポイント - `http://localhost:8080/RequestCenter/nsapi/directory/randompass`  
ヘッダー パラメータ
- コンテンツ タイプ - application/json

**API の応答は、JSON で次のようになります。**

```
{
  "List" : [
    "XJ(j@3x4"
  ]
}
```

#### パスワードを変更する API

- API - GET 方法 -
- エンドポイント -

`http://localhost:8080/RequestCenter/nsapi/authentication/login?resetPassword=true&NSA_LOGIN_NAME=user1&CURRE`

```
NT_PASSWORD=!@^_YWRtaW4=_^@!&NEW_PASSWORD=!@^_YWRtaW4=_^@!&CONFIRM_PASSWORD=!@^_YWRtaW4=_^@!
```

ヘッダー パラメータ

- コンテンツ タイプ - application/json
- 送信されるパスワードは Base64 でエンコードされます。

**API の応答は、JSON で次のようになります。**

```
{
  "nsapi-response" : {
    "NSA_PASSWORD_UPDATE_SUCCESS" : "You have successfully changed your password. 新しいパスワードでログインできます。"
  }
}
```

### ログインしてセッション ID を取得する API

- API - GET 方法 -
- エンドポイント : -http://localhost:8080/RequestCenter/nsapi/authentication/token?persistent=true  
ヘッダー パラメータ
- コンテンツ タイプ - application/json

**API の応答は JSON で次のようになります。 :**

```
{
  "sessiontoken" : {
    "utid" : "P_D7E79E1479600246A4932444172E058D",
    "sessionId" : "ylSpNsKqJUbiy0zdF8eLzHC5NOrYUxfyCmTn_QkU",
    "loginModule" : "Service Catalog",
    "loginModuleLogicName" : "servicecatalog"
  }
}
```

### システムからログアウトする API

- API - GET 方法 -
- エンドポイント - http://localhost:8080/RequestCenter/nsapi/authentication/logout?persistent=true&isDecoupled=true  
ヘッダー パラメータ
- コンテンツ タイプ - application/json

**API の応答は、JSON で次のようになります。**

```
{
  "nsapi-response" : {
    "Success" : "User is Logged out."
  }
}
```

### グローバル設定用 API

設定 API は、コンプライアンス Web サイトが有効になっている、または SAML 認証が有効になっているなど、サイト レベルのパラメータを取得するためにログイン ページで使用されます

- API - GET 方法 -  
エンドポイント : http://localhost:8080/RequestCenter/nsapi/global/configuration  
ヘッダー パラメータ
- コンテンツ タイプ - application/json

**API の応答は JSON で次のようになります。 :**

```
{
  "Map" : {
    "enableCustomLoginLogout" : false,
    "isSAMLSSO" : false,
  }
}
```

```
"gdprConsentUrl" : "http://kjcjbvcbajs.com",
"enableComplianceWebsite" : true
}
```

## ブラウザ ベース イベントの DDR ルールを実行する API

この API は、ruleId が PathParam の一部であったバージョン **12.1\_Patch\_v6 で導入** で以前導入されました。このパッチでは、queryId として ruleId を使用するように API がアップグレードされました。

- API – PUT 方法 –  
エンドポイント : エンドポイント : <http://ServerURL:Port/RequestCenter/nsapi/catalog/v1/service/Id/rule?ruleId=<encoded ruleId>> ヘッダー パラメータ
- コンテンツ タイプ - application/json
- 受付 : application/json

**API の応答は JSON で次のようになります :**

```
{
  "listOfValues" : [
    [
      {
        "key" : "<dictionary_name>.<field_name>",
        "value" : [
          {
            "label" : "<label_configured_in_rule>",
            "value" : "<value_mapped>"
          },
          {
            "label" : "3-tier Production Application SVG",
            "value" : "7"
          },
          {
            "label" : "Activity SVG",
            "value" : "8"
          },
          {
            "label" : "Apple Macbook",
            "value" : "9"
          }
        ]
      }
    ]
  ],
  "listOfTargets" : [
    "<dictionary_name>.<field_name>"
  ]
}
```

## 12.1\_Patch\_v8 で導入

注 : パーティクル フォーム サーバのサポートは廃止され、フォルダは次の場所から削除する必要があります。

```
/Requestcenter.war/website/FormServer
```

### マテリアル UI コンポーネントを使用したサービス フォーム レンダリング。

マテリアル UI コンポーネントを使用してサービス フォームをレンダリングするためのプロビジョニングがあり、**ReactJS** フレームワークで設計されています。これはすべての機能が従来のサービス フォームに似ており、カスタマイズ可能です。この新しいサービス フォームは API を使用して設計されており、バックエンドから分離されています。すぐに使用できるマテリアル UI ServiceForm は、このリリースの Grid Control と Paginated ServiceForm を除くすべてのフィールドをサポートします。

新しいサービス フォーム UI の仕組みについては、「[12.1\\_Patch\\_v7 で導入](#)」の「[新しいサービス フォーム](#)」セクションを参照してください。

### 新しいサービス フォームの有効化

デフォルトで表示されるサービス フォームは従来型で、次の場所から新しいマテリアル UI サービス フォームを有効にできます。

- 管理→設定
- [サービス デザイナー (Service Designer)] > [全般 (General)] タブ

**注：** JavaScript 関数は、マテリアル フォーム サーバと互換性があるように書き換える必要があります、既存の JavaScript 関数は新しいサービス フォームでは動作しません。

新しいサービス フォームの場所

新しいサービス フォームのソース コードは、次の URL から入手できます。

`/Requestcenter.war/website/FormServer/MaterialFormServer`

マテリアル フォーム サーバのブラウザ互換性

- ✓ サポートされるブラウザ：
  - Microsoft Edge バージョン 42、
  - Mozilla Firefox バージョン 64、
  - Google Chrome バージョン 70、
  - Safari バージョン 12.0
- **注：** Internet Explorer 11 は廃止されており、マテリアル フォーム サーバでは使用できません。

マテリアル フォーム サーバのテクノロジー スタック

新しいサービス フォーム UI は次のライブラリを使用します。

- a) **NodeJS**
  - 次のテクノロジー スタックのベース
- b) **WebPack バージョン 4.20.2**
  - プロジェクト全体をコンパイルし、実稼働構築を準備するために使用されます。
- c) **React JS バージョン 16.7.0**
  - 新しい品目フォーム サーバの開発に使用
- d) **マテリアル UI ライブラリ バージョン 3.9.0**
  - マテリアル フォーム サーバを設計するためのコアライブラリとして使用
- e) **構文的に優れたスタイルシート ( Sass )**
  - CSS スタイリングに使用

カスタム サービス フォームの変更

ここでは、新しいコンポーネントを設計したり、既存のコンポーネントを強化したりするためのサービス フォームの変更について説明します。

- 新しいライブラリを導入するには、**package.json** ファイルを更新します。  
必要に応じて、次の設定ファイルで必要な変更を行います。
  - `config/webpack.config.js`
- PSC が提供する新しいサービス フォームコードを変更して、任意のコンポーネントをカスタム変更したり、新しいコンポーネントを設計したりするには、次のファイルを正しい順序で更新します。
  - a) `Index.js`
  - b) `App.js`
  - c) `AppInit.js`
  - d) `ServiceForm.js` (サービス フォーム ヘッダー、本文、およびフッター コンポーネントを作成)
- 新しいサービス フォーム (CSS) のスタイリング フォルダのルート パスを変更するには、



### – /src/styles

すべての CSS スタイル ファイルは、上記のルートパスの親フォルダにあります。

これらすべてのファイルの拡張子は **.scss** です。**.scss** ファイルをインポートし、**shared.scss** という名前の単一のファイルにマージする必要があります。新しいファイルを含める場合は、ファイル名をアンダースコア ( \_ ) で開始します。たとえば、新しいファイルに **\_BasicForm.scss** という名前を付け、このファイルを **shared.scss** にインポートします。マテリアル UI コンポーネントのカスタム CSS は、**Shared/cssUtility.js** フォルダに配置されます。

### ➤ すべての新しいライブラリと変更されたファイルをコンパイルするには、次の手順を実行します。

#### 前提条件：

- ✓ 対応するシステムに最新の **node.js** ライブラリをインストールします。
- ✓ 「**npm install**」を使用して、新しいサービス フォームで使用される **ReactJS** およびその他の関連ライブラリをインストールします。

**注：** **npm install** コマンドを使用するには、インターネットに接続している必要があります。

### ➤ すべての変更をコンパイルし、実稼働構築を準備するには、次のコマンドを実行します。

#### – npm run build

**注：** 実稼働構築は、縮小されたファイルのセットです。

- プロダクション構築フォルダ (**/dist**) をコピーし、次の場所に置き換えます。  
Requestcenter.war/website/FormServer/MaterialFormServer/dist
- マテリアル UI のホーム URL：<https://material-ui.com/>

## サービス フォームに関するマテリアル UI の制限

### スライダ コンポーネント

マテリアル フォーム サーバは、**onFocus** および **onBlur** イベントをサポートしていません。

### 日付および日時コンポーネント

マテリアル UI の日付および日時コンポーネントは、選択したユーザの日付形式で短い方に基づいて動作します。

- マテリアル UI は次の日付形式をサポートしていません：DD/MM/YY、DD/MM/YYYY、YY/MM/DD
  - Safari ブラウザでは、DD/Month/YY および DD/Month/YYYY は YYYY/MM/DD に置き換えられます。
  - Microsoft Edge では、DD/Month/YY および DD/Month/YYYY の日付形式はサポートされていません。
- **date** および **dateTime** コンポーネントは、プロファイルの日付区切り文字としてスラッシュ (/)、ハイフン (-)、およびピリオド (.) をサポートしますが、UI にはスラッシュ (/) のみが表示されます。
- **date** および **dateTime** コンポーネントは、**onClick** および **onBlur** イベントをサポートしていません。

#### 注：

- マテリアル UI と対応するブラウザでサポートされている日付形式のみが機能します。
- 個人フィールドが条件付きルールで設定されている場合、値は **personID : personLoginName** としてパスする必要があります。

## 個人プロフィール UI

個人プロフィール UI は、BackBone JavaScript ベース UI に置き換えられ、必要なデータを提供するためのサポート API を備えています。これで、カスタマイズが可能になりました。

新しい個人プロフィール UI は、以下のモジュールの Web サイトで使用されます。

- クラウド統合 (Cloud Integrations)
- コンプライアンス
- サービス カタログ (Service Catalog)
- サービス マネージャ (Service Manager)

- テナント管理
- ユーザ管理

### サービス カタログ Web サイト

Service Catalog Web サイトは完全に API ベースの Webサイトになり、すべての struts コールが削除されました。ダッシュボード、レポート、注文、認証、およびサービス項目に関連するファイルは、Service Catalog Web サイトに移動しました。

### 新しいサービス フォームのレンダラ設定の変更

- **サイト レベル レンダラ設定**  
管理レベルでのレンダラ設定へのパス  
[管理 ( Administrations ) ] > [設定 ( Settings ) ] > [レンダラの選択 ( Select Renderer ) ] > [クラシック ( Classic ) ] または [MaterialFormServer]
- **サービス レベル レンダラの設定**  
サービス レベルでのレンダラ設定へのパス、  
[サービス デザイナー ( Service designer ) ] > 左ペインの [サービス ( Service ) ] を選択し、 > [レンダラのオーバーライド ( Override Renderer ) ] > [従来 ( Classic ) ] または [MaterialFormServer] > [保存 ( Save ) ] を選択します

### ローカライズされた JavaScript 文字列へのアクセス

サービスのカスタム スクリプトを開発する際に、サービス デザイナー モジュールからサービスのローカライズされたリソース文字列に JavaScript からアクセスするには、次の関数を使用します。

```
getLocalizedString("<js_resource_id");
```

- Cisco Prime Service Catalog 12.1 デザイナー ガイドの「[JavaScript 文字列の変換](#)」を

### JavaScript レンダラのサポート

Java スクリプト関数の定義は、サービス デザイナー→スクリプトでさまざまなレンダラに対して定義できます。

サービス フォームがレンダリングされると、サービスのレンダラに基づいて、対応する Java スクリプトが取得され実行されません。

### 認証トークン API

認証 API の応答が拡張され、新しい応答属性 **sessionId**、**loginModule**、および **loginModuleLogicName** が追加されました。

- **sessionId** : セッションに固有の sessionId。認証に使用されます。
- **Loginmodule** : ログイン後に表示されるデフォルト モジュール。これは、個人プロファイルの設定に基づいています。
- **loginModuleLogicName** : 各モジュールの一意の識別子。

### REST API

```
RequestCenter/nsapi/authentication/token?persistent=true
```

### API メソッド : GET

### サンプル ペイロード :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sessiontoken utid="P_95A5F7D47752CF96324466027606864B">
  <sessionId>EYsEWXmVgCqaWZKAK1fMU_HJJ7m_QOOP0JEhVMrW</sessionId>
  <loginModule>Service Catalog</loginModule>
```

```
<loginModuleLogicName>servicecatalog</loginModuleLogicName>
```

```
</sessiontoken>
```

## 12.1\_Patch\_v7 で導入

### 新しいサービス フォーム UI

新しいサービス フォーム UI は、ParticleUX が付属した ReactJS などの新しいテクノロジー スタックを使用して設計されています。サービス フォーム UI は、カスタム構築の ServiceForm 実装が行える Web サイトで、同じものがレンダリングされます。新しいサービス フォーム UI は、サービス フォームをレンダリングするために API を使用し、バックエンドから完全に分離されています。

パーティクル UI ServiceForm はベータ版であり、ServiceForm 関数のすべてを完全にはサポートしていません。ただし、この実装はカスタム ServiceForm UI を開発するための参照として使用できます。また、ParticleUX コンポーネントのカスタマイズのサポートは現在予定されていません。

この新しいサービス フォームは、**[管理 ( Administration ) ] -> [設定 ( Settings ) ] または [サービスデザイナー ( Service Designer ) ] -> [全般 ( General ) ] タブのいずれかから有効にできます。**

**注：**パーティクル フォーム UI/カスタム UI では、対応する JavaScript ライブラリを使用して Java スクリプトを記述する必要があります。既存の Javascript は、新しいサービス フォームでは機能しません。

パーティクル UI である UI ロジックの 1 つの実装が提供されており、3 つの主要なコンポーネントがあります。

### コンポーネント

サービスフォームのタイプ

1. **従来のサービス フォーム**
  - 旧サービス フォーム
2. **パーティクル フォーム サーバ**
  - ReactJS with ParticleUX を使用して設計された新しいサービス フォーム

新しいサービス フォームの場所

**注：**新しいサービス フォームのソース コードの場所は

```
/Requestcenter.war/website/FormServer/ParticleFormServer
```

パーティクル フォーム サーバのブラウザ互換性

- ✓ たとえば、すべてのブラウザをサポートします。
  - IE Edge (バージョン 42) 、
  - Mozilla Firefox (バージョン 64) 、
  - Google Chrome (バージョン 70)
  - Safari (12.0)
- **注：**IE11 は非推奨であり、パーティクル フォーム サーバでの使用は推奨されません。

パーティクル フォーム サーバのテクノロジー スタック

新しいサービス フォーム UI は、次のライブラリを使用して設計されています。

- f) **NodeJS**
  - 以下のテクノロジー スタックのベース
- g) **WebPack (バージョン 4.20.2)**
  - プロジェクト全体をコンパイルし、実稼働構築を準備するために必要です。
- h) **React JS (バージョン 15.6.2)**
  - 新しいパーティクル フォーム サーバの開発に使用
- i) **Cisco パーティクル UI ライブラリ (バージョン 2.9.0)**
  - パーティクル フォーム サーバを設計するコア ライブラリとして使用
- j) **Sass (構文的に素晴らしいスタイルシート)**

CSS スタイリングに使用

## フォーム ルール用の ISF ライブラリ

- サービス フォーム ルールの実行 (CR および DDR) は、ルール固有の生成された JavaScript 関数を使用して実行できます。これらの関数は、ISF ライブラリの関数を呼び出して、目的のサービス フォーム操作を動的に実行します。
- テンプレートファイル (formrules\_javascript\_tpl.ftl) は、ルール固有の JavaScript 関数を生成します。このテンプレートは、設定された DOM イベントでルールをバインドするコードも生成します。
- 例：
  - DDR ルールを実行するために、テンプレートは ISF.isf\_ddr\_call() を呼び出す関数を生成します。
  - アラート アクションを実行するために、テンプレートは ISF.isf\_cr\_alert() を呼び出す機能を生成します。
- このテンプレート ファイルは、/RequestCenter.war/WEB-INF/classes/service-form-rules-js-templates/formrules\_javascript\_tpl.ftl にあります。
- このテンプレート ファイルへのパスは設定可能です。詳細については、「[FTL ファイル パスの Newscale プロパティ設定](#)」セクションを参照してください。
- パッチには、この JavaScript コード生成を実現するために FTL 変換を使用するテンプレートが付属しています。このテンプレートは、完全なフォーム ルール実行の異なる実装を実現するために書き換えることができます。詳細については、「[フォーム ルール固有の JavaScript を生成する FTL テンプレート](#)」を参照してください。
- この生成された javascript では、属性「javascriptFunctionDefinitionsPO」の下のフォーム メタ データ API の応答で確認できます。
- これらの関数を DOM の一部にするには、この javascript を実行します。このコードを実行すると、フォーム ルールの関数マップへのイベントも初期化されます。このマップは、ISF ライブラリで宣言されます。

## Javascript フォーム メタ データ API 応答

```

8
9  {
10  "formPO": {
11    ...
12    "formSectionPO": [
13      {
14        ...
15        "id": "XX_Tenant_Information_XX",
16        "dictionaryAccess": true,
17        "formFieldPO": ...
18      }
19    ]
20    "javascriptFunctionDefinitionsPO": {
21      "content": "\n function rc_GL_CMN_Remeddy_Dictionary_Hide() {\n   try {\n     if (Moment ==
22    }
23  }
24 }

```

イベントがトリガーされると、サービス フォームはこのイベントにマッピングされたフォーム ルール JavaScript 関数を実行します。ルール マップに対する ISF イベントは、これらのフォーム ルール JavaScript 関数を取得するために使用されます。

- ISF (Interactive Service Form) ライブラリは、「/ParticleFormServer/src/addins/isf-addin.js」で定義された javascript オブジェクトです。ブラウザのウィンドウ オブジェクトの一部にします。
  - ISF ライブラリ関数の実装の詳細については、Cisco.com の「[Cisco Prime Service Catalog 12.1パッチ - パーティクルフォーム サーバ ISF ライブラリ メソッドシグネチャ](#)」を参照してください。

フォーム ルール固有の JavaScript を生成する FTL テンプレート

テンプレートは xml のフォーム ルール設定を受け取り、対応する javascript を生成します。

フォーム ルールを完全に実行するには、テンプレートに次のものがが必要です。

- 対応する ISF java スクリプト関数を呼び出す各フォーム ルールのハンドラ。これらの js 関数は、意図したルール動作を実行します。ISF 機能は、新しいレンダラでも変更できます。
- 各ハンドラは、フォーム ルールに設定されたイベントでトリガーにバインドされます。

## 外部 JavaScript

パーティクル フォーム サーバのスクリプトを記述し、従うためのガイドライン。

## カスタム スクリプト内の serviceForm データにアクセスするには

serviceForm オブジェクトは、すべてのスクリプト関数に最後の引数値としてパスされます。次の 2 つの方法でアクセスできます。

## 1. 追加の関数引数 serviceForm

```
function1 (arg1、 arg2、 ...、 serviceForm) {
    //write script here
    .....
}
```

- 「arg1、 arg2、 ...」 は、サービス デザイナー スクリプトの「Function Arguments」セクションから定義される引数です。
- 「serviceForm」は serviceForm オブジェクトを持つ引数であり、サービス デザイナー スクリプトの「Function Arguments」セクションから定義することはできません。

## 2. 関数定義内での「引数」 javascript ローカル変数の使用

最後の引数は serviceForm で、次のようにアクセスできます。

```
function2(arg1, arg2,...) {
    let serviceForm = arguments[arguments.length-1];
    //write script here
    .....
}
```

- ここで、「arg1、 arg2、 ...」は、サービス デザイナー スクリプトの「Function Arguments」セクションから定義される引数です。

## サービス フォーム フィールドの値を取得するには

```
ISF.isf_getValue(serviceForm, <Dictionary_Name.Field_Name>);
```

```
function3() {
    let serviceForm = arguments[arguments.length-1];
    let fieldName = ISF.isf_getValue(serviceForm, "Dictionary1.Name");
}
```

## サービス フォーム フィールドに値を設定するには

```
ISF.setTargetFieldValue(serviceForm, <Dictionary_Name.Field_Name>, <Field_Value>);
```

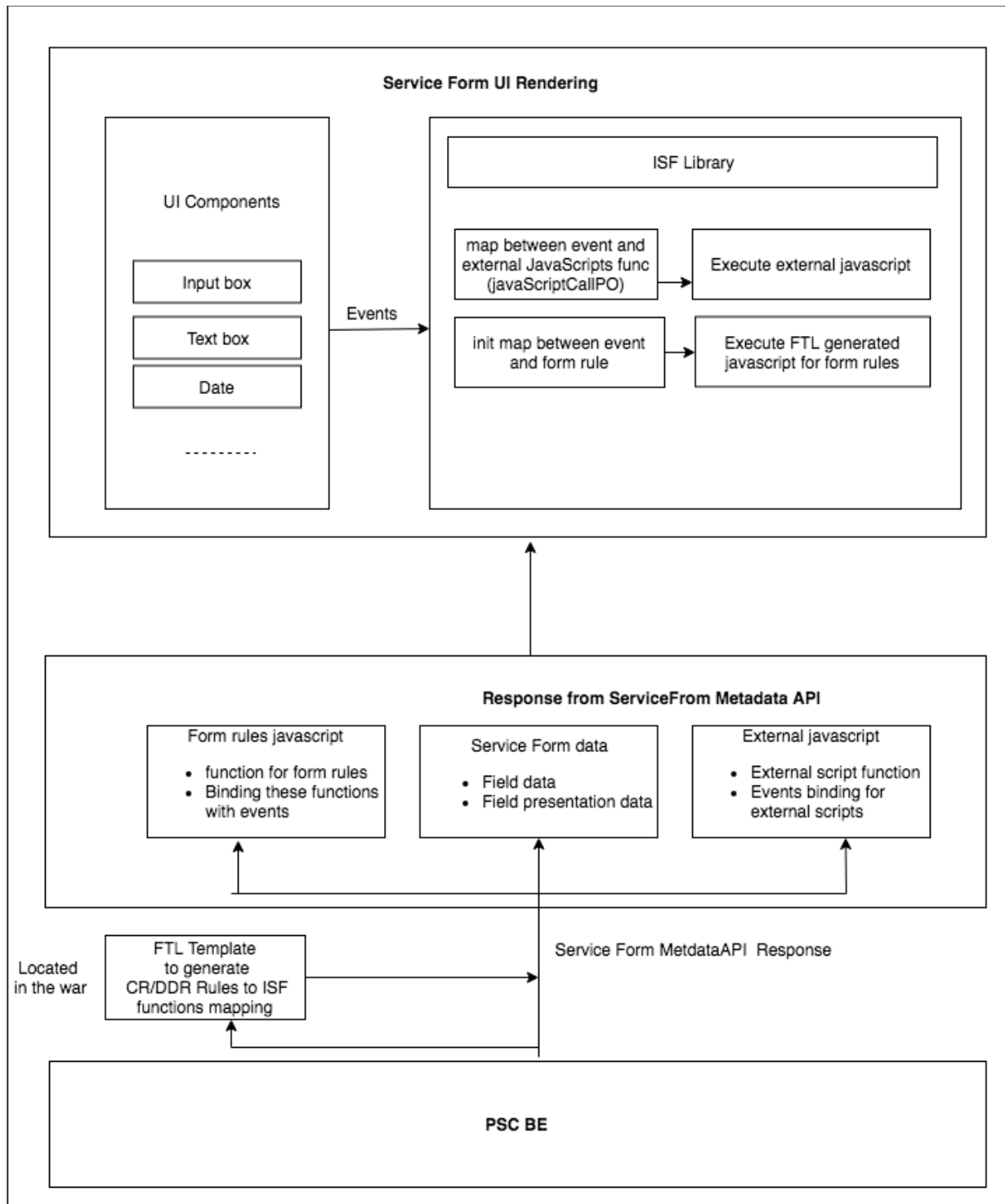
```
function4() {
    let serviceForm = arguments[arguments.length-1];
    let fieldName = ISF.isf_getValue(serviceForm, "Dictionary1.Name");

    ISF.setTargetFieldValue(serviceForm, "Dictionary2.Name", fieldName);
}
```

注： ISF ライブラリがサポートする残りの js 関数については、以下の JS ファイルを参照してください。

RequestCenter.war ▶ website ▶ FormServer ▶ ParticleFormServer ▶ src ▶ addins ▶ isf-addin.js

### サービス フォーム UI レンダリング フロー



### 新しいカスタム サービス フォーム用に変更するファイル

以下に示す既存のコンポーネントまたは新しいコンポーネントの設計におけるあらゆるタイプの拡張のために新しいサービス フォームを変更する手順

- 新しいライブラリを導入するには、**package.json** を更新する必要があります。

**注：**必要に応じて、次のような設定ファイルで必要な変更を行います。

- config/webpack.config.dev.js
- config/webpack.config.prod.js

- PSC 提供の**新しいサービス フォーム** コードを変更して任意のコンポーネントでカスタム変更を行うか、新しいコンポーネントを設計するには、以下に記載されているファイルにアクセスして、適切な順序で行う必要があります

- e) Index.js
- f) App.js
- g) Applnit.js
- h) ServiceForm.js (サービス フォーム ヘッダー、本文、およびフッターのコンポーネントがここに作成されます)

- 新しいサービス フォーム (CSS) のスタイリング フォルダのルート パスを変更するには

- /src/styles

**注：**すべての CSS スタイル ファイルは、上記のルート パスの親フォルダで使用できます。これらのファイルはすべて ( **.scss** ) 拡張子ベースのファイルであり、( shared.scss ) と呼ばれる単一のファイルにインポートおよびマージされます。新しいファイルが導入される場合、ファイル名はアンダースコアで始まる必要があります ( 例 : `_BasicForm.scss` )。そして、この新しいファイルを ( shared.scss ) にインポートする必要があります。

- すべての変更されたファイルと新しく導入されたライブラリのコンパイル

**前提条件：**

- ✓ 最新の **node.js** ライブラリを対応するシステムにインストールする必要があります
- ✓ 「npm install」を使用して、新しいサービス フォームで使用されている **ReactJS** およびその他の関連ライブラリをインストールします。

**注：**npm install (このコマンドではインターネット接続が必須)

**Compilation：**

- npm run build

**注：**上記のコマンドはすべての変更をコンパイルし、縮小されたファイルのセットである実稼働構築を準備します。

- 実稼働構築フォルダ ( / dist ) を次の場所にコピーして置き換える必要があります。

/Requestcenter.war/website/FormServer/ParticleFormServer/dist

**注：**新しいサービス フォームは UCSD および Cloud Center サービスをサポートしていません。

**新しいサービス フォーム UI の API 拡張**

サービス オーダー フォーム コンテキスト API

サービス オーダー フォーム コンテキスト API は、サービス オーダー フォームのロードに使用されるコンテキスト パラメータを提供し、ユーザが未送信の要求を持っているかどうかを確認するための追加フィールド **hasUnsubmittedRequisitions** を備えています。

/RequestCenter/nsapi/catalog/v1/service/{id}/orderformcontext?customerId={customerID}

**API メソッド：** GET

**応答ペイロード：**

```
{
  "Map": {
```

```

"siteAdminUser": true,
"EnableOOBAutoComplete": false,
"orderConfirmationDeliverToFormat": "Customer(Login)",
"allowAddAttachment": true,
"isNextGenUser": true,
"clickableHelpIcon": false,
"timeZone": "America/Tijuana",
"hideNRDicts": false,
"servicedescproperty": false,
"hideMonitorFlag": false,
"Serviceformelement.display.instructional.helpertext": false,
"isMessageInfo": "false",
"DescriptionLimit": "0",
"nsBasePath": "/RequestCenter/",
"service": {
  "serviceID": 150,
  "name": "testService1",
  "documentID": 0,
  "estimatedCost": 0.0,
  "expectedDuration": 0.0,
  "hasPortal3": true,
  "portalText3": "<p>TEST3</p>\n",
  "isEntitlement": 0,
  "isInactive": 0,
  "priceDescription": "",
  "shortPriceDescription": "",
  "priceDisplaySchemaID": 0,
  "pricingSchema": 0,
  "revisionNumber": 34,
  "allowFutureDelivery": true,
  "isBundle": false,
  "isTemplate": false,
  "tenantRelevant": true,
  "isOrderable": true,
  "orderingMode": 3,
  "computePrice": false,
  "paginationViewMode": 0,
  "isDefinedDuration": false,
  "showOrderSummary": false
},
"allowUpdateQty": false,
"sessionToken": "F85E174AB7425AD66555E5CBD517CDF4",
"ldapLookupOnOnlyForPerson": true,
"ldapLookupOn": true,
"Serviceform.label.alignment": "right",
"Serviceform.field.line.separator": false,
"hasUnsubmittedRequisitions": false
}
}

```

#### 申請エントリ API

追加フィールド **MomentLogicName** および **isBundle** が requisitionentries API に追加されます。

- サービスを注文すると、サービスは設定された複数の瞬間を通過します。
- Ordering
- 部門による承認
- 部門による確認



- サービス グループの認証
- サービス グループのレビューは
- 財務承認
- サービス提供
- 価格設定 (Pricing)
- サービスの完了

**MomentLogicName** には上記の瞬間の適切な値があり、条件評価のための条件付きルール、データ取得ルールで使用されます。

- **isBundle = true** の場合、要求エントリはバンドルされたサービス用です。

```
/RequestCenter/nsapi/transaction/requisitionentries/id/{id}?isNgcRequest=true&responseType=json
```

**API メソッド** : GET

**応答ペイロード** :

```
{
  "requisitionEntry": {
    "requisitionEntryId": 10,
    "requisitionId": 8,
    "serviceId": 142,
    "serviceDescription": "",
    "serviceName": "DDDR",
    "standardDuration": "Not Defined",
    "unitCost": "$ 0.00",
    "subTotal": "$ 0.00",
    "status": "Ongoing",
    "dueOnDate": "10/31/2018",
    "dueOnDateRaw": 1540981800000,
    "quantity": 1,
    "maxQuantity": 0,
    "serviceLevelDescription": "",
    "percentageCompleted": 0.0,
    "statusId": 1,
    "expectedDuration": 0.0,
    "isBundle": false,
    "reqEntryServiceURL": "<a href=#\" onclick=\\\"GB_showFullScreen('DDDR',/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&layout=popup_p&reqid=8&reqentryid=10&formAction=displayEntryStatus&performerID=&serviceid=142&requisitionId=8)\\\">DDDR</a>",
    "reqEntryServiceURLFor_Preparation": "<a href=#\" onclick=\\\"GB_showFullScreen('DDDR',/RequestCenter/myservices/navigate.do?query=vieworderform&layout=popup_p&serviceid=142&requisitionId=8&requisitionEntryId=10)\\\">DDDR</a>",
    "reqEntryServiceURLOnlyFor_Preparation":
"/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&reqid=8&reqentryid=10&formAction=displayEntryStatus&performerID=&serviceid=142&requisitionId=8",
    "reqEntryServiceURLOnly":
"/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&reqid=8&reqentryid=10&formAction=displayEntryStatus&performerID=&serviceid=142&requisitionId=8",
    "requisitionStatusId": 1,
    "requisitionStatus": "Preparation",
    "startedDate": "10/30/2018",
    "startedDateRaw": 1540949340000,
    "customerName": "admin admin",
    "customerId": 1,
    "customerLoginName": "admin",
    "customerOUID": 1,
  }
}
```

```

"customerEmail": "xyz@company.com",
"reviewsCount": 0,
"selfRating": 0,
"isServiceOrderable": 1,
"isServiceActive": 1,
"isCancelable": 1,
"momentLogicName": "servicedelivery",
"ownerId": 1,
"dueBy": "41 day(s) overdue",
"renderer": "Classic",
"orderOnBehalf": false
}
}

```

追加のクエリ パラメータ

新しいサービス オーダー フォームに、サービスの単位コストを示す追加のクエリ パラメータ **pricePerUnit** が追加されました。

#### カートに追加する API

```
/RequestCenter/nsapi/catalog/v1/requisition?pricePerUnit={"__DEFAULT_PRICE_KEY":"40"}
```

**API メソッド** : PUT

**要求 API を送信**

```
/RequestCenter/nsapi/catalog/v1/requisition?pricePerUnit={"__DEFAULT_PRICE_KEY":"40"}
```

**API メソッド** : POST

**要求 API を更新**

```
/RequestCenter/nsapi/catalog/v1/requisition?pricePerUnit={"__DEFAULT_PRICE_KEY":"40"}
```

**API メソッド** : PUT

更新要求のためのドラフト API として保存

このAPIを使用すると、ユーザはサービスをドラフトとして保存できます。このアクションを実行すると、チーム管理が有効になっている場合は、チーム名を除くすべてのサービス フォーム フィールドの検証がスキップされます。

```
/RequestCenter/nsapi/catalog/v1/requisitionentry/{requisitionentry}/form?responseType=json&saveAsDraft=true
```

**API メソッド** : PUT

**サンプル ペイロード** :

```

{
  "dictionaries": [
    {
      "name": "<dictionary-name1>",
      "data": {
        "<attribute1>" : "<value1>",
        "<attribute2>" : "<value2>"
      }
    },
    {
      "name": "<dictionary-name2>",
      "data": {
        "<attribute1>" : "<value1>",
        "<attribute2>" : "<value2>"
      }
    }
  ]
}

```

```

    },
    .....
  ]
}

```

#### SIBD デictionaryのサービス項目タイプ

サービス フォーム/要求エントリ メタデータ API には、サービス フォームでサービス項目の自動データ取得に使用される **dataTypeName** という追加フィールドがあります。このフィールドの値は、次に示すように SIBD Dictionaryのサービス項目タイプに設定されます。

```
/RequestCenter/nsapi/catalog/v1/service/<Service_Id>/form
```

```
/RequestCenter/nsapi/catalog/v1/requisitionentry/ <requisitionentryid>/form
```

**API メソッド : GET**

**サンプル応答 :**

```

{
  "formPO" : {
    "method" : "post",
    "timeStamp" : "",
    .....
    "serviceID" : 180,
    "formSectionPO" : [
      {
        "name" : {
          "uiresource" : {
            "key" : "siService",
            "escapeJS" : "false",
            "keyType" : "preprocess"
          }
        },
        "id": "SIDict1",
        "maxRows" : 0,
        "displayRows" : 0,
        "gridEditable" : false,
        "displayAsGrid" : false,
        "autoDataRetrieval" : false,
        "dataTypeName" : "SitestSI01",
        "templateTypeId" : "8",
        "mdrRecordID" : 0,
        "formFieldPO" : [
          {
            "value" : "",
            "id" : "SIDict.Name",
            "scale" : 0,
            "maxLength" : 128,
            "inputType" : "text",
            "fieldDataType" : "ServiceItemIdentifier",
            "hasError" : false,
            "serverEditableOnly" : false,
            "isMandatory" : false,
            "isSecure": 0,

```

```

    "formFieldLabelPO": {
      "content" : "Name"
    },
    "generateUnique" : 0,
    "showInGrid" : 1,
    "enableTypeahead" : 0,
    "typeaheadMinStart" : 0,
    "typeaheadMaxSearch" : 0,
    "presentationDataType" : {
      "textBox" : {
        "length" : "80",
        "name" : "SIDict.Name",
        "value" : "",
        "height" : "5",
        "fieldDataType" : "ServiceItemIdentifier",
        "isEditable" : "true",
        "controlName" : "nstextbox"
      }
    },
    .....
  ]
}
}
}
}
}

```

#### サービス項目 API

これは、新しいサービス オーダー フォームのサービス項目の自動データ取得に使用される既存の API です。

**/RequestCenter/nsapi/serviceitem/{dataTypeName}/name=<serviceItemName>**

#### API メソッド : GET

{dataTypeName} は、関連する SIBD ディクショナリの service/requisitionentry メタデータ API から取得されるサービス項目タイプ名です。サブスクリプションデータを取得するかどうかを決定する新しいクエリ パラメータ **fetchSubscription** が追加されました。

デフォルトでは、サービス項目のサブスクリプション データが取得されます。

#### サービス フォームの Expression API

式の値を計算するには、Expression API を使用します。

#### Rest API:

**/RequestCenter/nsapi/catalog/v1/expression**

#### API メソッド : PUT

#### ペイロード要求

```

{
  "expression" : "4.95624+5.0+29.216",
  "scale" : "2",
  "fielddatatype": "Money"
}

```

#### 応答

```

{
  "result" : "39.17"
}

```

#### 注 :

- 'scale' および 'fielddatatype' はオプションです。
- 'fielddatatype' の有効な値は 'Text'、'Money' です。
- 'scale' 値は数値で、小数点以下の桁数を示します。

- 'fielddatatype' 値が 'Money' の場合、'scale' 値が使用されます。

#### サービス項目請求の概算コスト API

これは既存の API で、新しいサービスフォームですべてのサービス項目の見積価格を取得します。

/RequestCenter/nsapi/catalog/v1/service/{id}/price

#### DDR API

DDR を実行するには、以下の API を呼び出す必要があります。この API の詳細については、パッチ v6 の readme の内容を参照してください。

/RequestCenter/nsapi/catalog/v1/service/Id/rule/<encoded ruleId>

この API の応答は、次のようにグループ化されたキーの複数の値を持つように拡張されました。

**API メソッド : GET**

#### サンプル応答

```
{
  "listOfValues" : [
    [
      {
        "key" : "DDRDictionary.service",
        "value" : [
          {
            "label" : "service1",
            "value" : "service1"
          },
          {
            "label" : "service2",
            "value" : "service2"
          },
          {
            "label" : "service3",
            "value" : "service3"
          }
        ]
      }
    ],
    [
      {
        "key" : "DDRDictionary.serviceGroup",
        "value" : [
          {
            "label" : "group1",
            "value" : "group1"
          },
          {
            "label" : "group2",
            "value" : "group2"
          },
          {
            "label" : "group3",
            "value" : "group3"
          }
        ]
      }
    ]
  ],
  "listOfTargets" : [
    "DDRDictionary.service",
    "DDRDictionary.serviceGroup"
  ]
}
```

## サービス フォーム URL の API 拡張

従来のサービス フォーム URL は、新しいサービス フォームにさらに拡張されました。サービス フォームの外観と表示方法を変更しています。

デフォルトでは、従来のレンダラ タイプは管理設定のグローバル パラメータで設定されていますが、管理者ユーザはサイト全体または個々のサービス レベルで設定するレンダラのタイプを決定でき、それ以降のサービス レベル レンダラはグローバル パラメータ設定をオーバーライドします。

新しいテクノロジー スタックでは、レンダラ タイプがパーティクル フォーム サーバに設定されている場合、新しいサービス フォームを従来のサービス フォームではなく製品全体で開くことができます。

### サービス フォームの拡張既存 API

次の API は、サイト レベルまたはサービス レベルで設定された各サービスのフィールド レンダラを含むように変更されました。

- サービスをサービス名で検索するには

```
/Requestcenter/nsapi/definition/servicedefs/?search={servicename}&searchFrom=ServiceCatalog_Home&responseType=json&recCount=30
```

- サービス項目ベースの特定のサービス ID の関連サービスを取得するには

```
/RequestCenter/nsapi/definition/servicedefs/id/{service id}/relatedservices
```

- サービス項目ベースのサービスを取得するには

```
/RequestCenter/nsapi/definition/servicedefs/serviceitemtypes
```

- サービス ID でサービスを取得するには

```
/RequestCenter/nsapi/definition/servicedefs/id/{service id}
```

- サービス名ごとにサービスを取得するには

```
/RequestCenter/nsapi/definition/servicedefs/name/{service name}
```

- ビュー名で認証を取得するには

```
/RequestCenter/nsapi/transaction/authorizations/AuthViewName={viewname}
```

- サービス項目 ID によるサービス項目サブスクリプションを取得するには

```
/RequestCenter/nsapi/serviceitems/serviceitemsubscription/PrimaryID={serviceitemids}?appendServiceData=true&appendRelatedServices=true&responseType=json&excludeServiceItemURLs=true&recordSize=20
```

- ログインしているユーザのビュー名で要求エントリーを取得するには

```
/RequestCenter/nsapi/transaction/requisitionentries/ViewName=InitiatorANDstatus=Preparation?responseType=json
```

### フィールド レンダラとその値を持つ API のうちの 1 つに対する応答。

```
/RequestCenter/nsapi/definition/servicedefs/serviceitemtypes
```

API メソッド : GET

サンプル バイロード :

```
{
  "services" : {
    "startRow" : 1,
    "totalCount" : 58,
    "recordSize" : 50,
    "service" : [
      {
        "serviceId" : 187,
```

```

    "serviceName" : "2TierContiane_rAug15(UCS)",
    "topDescription" : "",
    "middleDescription" : "",
    "bottomDescription" : "",
    "revisionNumber" : 310,
    "status" : "Active",
    "statusId" : 1,
    "expectedDuration" : 0,
    "expectedDurationUnits" : "hours",
    "canStartLater" : false,
    "isBundle" : false,
    "dateQualityId" : 4,
    "isOrderable" : true,
    "showOrderSummary" : false,
    "hideInServiceCatalog" : false,
    "hideInMyOrders" : false,
    "maxQuantity" : 0,
    "isReportable" : false,
    "serviceURL" : "<a href=/RequestCenter/myservices/navigate.do?query=serviceid&sid=187&layout=popup_p' onclick=\return GB_showFullScreen('2TierContiane_rAug15(UCS)', this.href)\>2TierContiane_rAug15(UCS)</a>",
    "serviceOrderURL" : "<a href=/RequestCenter/myservices/navigate.do?query=orderform&sid=187&layout=popup_p' onclick=\return GB_showFullScreen('2TierContiane_rAug15(UCS)', this.href)\>Order</a>",
    "serviceURLOnly" : "/RequestCenter/myservices/navigate.do?query=serviceid&sid=187",
    "serviceOrderURLOnly" : "/RequestCenter/myservices/navigate.do?query=orderform&sid=187",
    "orderingMode" : 2,
    "computePrice" : false,
    "portalText1" : "",
    "portalText2" : "",
    "portalText3" : "",
    "hasPortal1" : false,
    "hasPortal2" : false,
    "hasPortal3" : false,
    "isTemplate" : false,
    "isBaseService" : true,
    "templateType" : "null",
    "hoursPerDay" : 0,
    "templateRelevant" : false,
    "tenantRelevant" : true,
    "renderer" : "ParticleFormServer"
  }
}
}
}

```

### 新しいサービス フォームのレンダラ設定

サービス フォームは、UI コンポーネントのロードに選択されたレンダラを使用します。レンダラの設定は、次の手順に従って変更できます。

- **サイト レベル レンダラ設定**

管理レベルでのレンダラ設定へのパス

[管理 ( Administrations ) ] [設定 ( Settings ) ] [レンダラの選択 ( Select Renderer ) ] [クラシック ( Classic ) ] または [パーティクル フォーム サーバ ( Particle Form Server ) ]

- **サービス レベル レンダラの設定**

サービス レベルでのレンダラ設定へのパス、

[サービス デザイナー ( Service designer ) ] > [左ペインでサービスを選択する ( Select a Service in left pane ) ] > [レンダラのオーバーライド ( Override Renderer ) ] > [クラシック ( Classic ) ] または [パーティクル フォーム サーバ ( Particle Form Server ) ] > [保存 ( Save ) ]

### FTL ファイルパスの Newscale プロパティ設定

- プロパティ名 : formrule.js.template.path : フォーム ルール Javascript の生成に使用される FTL テンプレートへの相対パスを提供します。
- デフォルト値は service-form-rules-js-templates / formrules\_javascript\_tpl.ftl です
- カスタム FTL ファイル テンプレートに対応するパスを割り当てます。

### パーティクル フォーム UI のフィールド イベント

以下に示すように、一部のイベントは新しいパーティクル UI の異なるデータ タイプではサポートされていません。

- 日付 : OnFocus、OnBlur、OnClick、OnMouseover、OnMouseout
- 日時 : OnFocus、OnBlur、OnClick、OnMouseover、および OnMouseout
- 人物 : OnFocus、OnClick、OnBlur、OnMouseover、および OnMouseout
- テナント : OnFocus、OnClick、OnBlur、OnMouseover および OnMouseout
- ラジオ : OnFocus、OnClick、OnBlur、OnMouseover、および OnMouseout
- チェックボックス : OnFocus、OnBlur、OnMouseover、および OnMouseout
- 単一選択 : OnFocus、OnClick、OnBlur、OnMouseover および OnMouseout
- 複数選択 : OnClick、OnMouseover および OnMouseout

#### 注 :

- 新しいサービス フォームでは、Date および Datetime フィールドに複数の値が表示されません。これらのフィールドに複数の値が含まれている場合、これらのフィールドはすべてカンマ区切りで表示されます。
- DDR ルールは、日付フィールドと日時フィールドではサポートされていません

## 12.1\_Patch\_v6 で導入

次の表に、このリリースで提供される新しい API を示します。このパッチで導入されたその他の機能拡張と使用方法の詳細については、次の項を参照してください。

表 1. パッチ v6 で導入された API のリスト

nsAPI	説明	REST URL
新規および既存の注文のサービス オーダー フォーム メタデータの取得	サービス オーダー フォームをレンダリングするために必要なすべてのメタデータを提供します。	<b>新規注文 :</b> /RequestCenter/nsapi/catalog/v1/service/<Service_Id>/form  <b>既存の注文 :</b> /RequestCenter/nsapi/catalog/v1/requisitionentry/ <requisitionentryid>/form <b>DDR ルールを実行する API :</b> /RequestCenter/nsapi/catalog/v1/service/Id/rule/<encoded ruleId>
DDR ルールの実行	ブラウザ ベース イベントの DDR ルールを実行する API	<b>サンプル要求と DDR の応答</b>  <a href="/RequestCenter/nsapi/catalog/v1/service/169/rule/RJZi0%2FXysRe6RdF%2BH1NWcQ%3D%3D">/RequestCenter/nsapi/catalog/v1/service/169/rule/RJZi0%2FXysRe6RdF%2BH1NWcQ%3D%3D</a>
サービス オーダー フォーム アクション API	要求を送信し、カートに追加し、ドラフトとしてサービスのみを保存し、ドラフトのサービスを含むドラフトとしてサービスを保存します	<b>要求を送信 :</b> <a href="/RequestCenter/nsapi/catalog/v1/requisition">/RequestCenter/nsapi/catalog/v1/requisition</a>



		カートへの追加 :	
			<a href="/RequestCenter/nsapi/catalog/v1/requisition">/RequestCenter/nsapi/catalog/v1/requisition</a>
		ドラフトとしてサービスを単独で保存 :	
			<a href="/RequestCenter/nsapi/catalog/v1/requisition?saveAsDraft=true">/RequestCenter/nsapi/catalog/v1/requisition?saveAsDraft=true</a>
		ドラフトとしてサービスを保存します (ドラフト内のサービスを含む)。	
			<a href="/RequestCenter/nsapi/catalog/v1/requisition?saveAsDraft=true">/RequestCenter/nsapi/catalog/v1/requisition?saveAsDraft=true</a>
		複数の ServiceId のエクスポート ( S1 = 241、S2 = 242、S3 = 243 ) :	
			<a href="/RequestCenter/nsapi/rex/v1/definition/service?id=243&amp;id=242&amp;id=241&amp;includeContainedServices=true">/RequestCenter/nsapi/rex/v1/definition/service?id=243&amp;id=242&amp;id=241&amp;includeContainedServices=true</a>
複数のサービスのエクスポート/インポート	サービス ID のリストを取得し、サービスの詳細を xml にエクスポート/インポートします。	エクスポートされた XML をインポートします。	
			<a href="/RequestCenter/nsapi/rex/v1/definition/service">/RequestCenter/nsapi/rex/v1/definition/service</a>
サービス/ユーザーコンテキストの取得	サービス オーダー フォームのロードに使用されるコンテキスト パラメータを提供します。		<a href="/RequestCenter/nsapi/catalog/v1/service/150/orderformcontext">/RequestCenter/nsapi/catalog/v1/service/150/orderformcontext</a>
要求エントリを更新	要求に関連する要求エントリの更新に使用されます。		<a href="/RequestCenter/nsapi/catalog/v1/requisitionentry/217/form">/RequestCenter/nsapi/catalog/v1/requisitionentry/217/form</a>
サービス項目請求に基づくサービスの概算コストの計算	特定のサービス フォーム データのすべてのサービス項目の見積価格を提供します。		<a href="/RequestCenter/nsapi/catalog/v1/service/152/price">/RequestCenter/nsapi/catalog/v1/service/152/price</a>

### 新しい注文のサービス オーダー フォームのメタデータを取得する API

この API は、サービス オーダー フォームをレンダリングするために必要なすべてのメタデータを提供します。メタデータには、フォーム フィールドとその表示プロパティを持つすべてのディクショナリ、CR および DDR ルール用に生成された Java スクリプト、およびグローバルなサービス固有の JavaScript が含まれます。

**前提条件 :** 以下の API を実行する前に、パラメータを newscaler プロパティで更新する必要があります。

```
#####
##Particle form rules js template path : service-form-rules-js-templates/formrules_javascript_tpl.ftl
formrule.js.template.path=service-form-rules-js-templates/formrules_javascript_tpl.ftl
#####
```

すべてのフォームの詳細コンテンツは、以下の API から取得でき、応答は JSON 形式になります

- API - GET 方法 -

エンドポイント : `http://<ServerURL> : Port/RequestCenter/nsapi/catalog/v1/service/<Service_Id>/form`  
 ヘッダー パラメータ

- Content-Type - application/json
- Accept : application/json

**RBAC アクセス** : 注文権限を持つユーザがこの API を実行できます。

**API の応答は次のとおりです。**

```
{
  "formPO" : {
    "method" : "post",
    "timeStamp" : "",
    "pages" : "[{"description":"Tenant Information Form for when Tenant Management is enabled","name":"Tenant-Information Form","sections":["XX_Tenant_Information_XX"]},{"description":"","name":"Attach Volume","sections":["Volume_Action","VolumeInfo_RO_Action","VolumeInfo_Action"]}],
    "orderingMode" : 3,
    "serviceID" : 136,
    "javascriptFunctionDefinitionsPO" : {},
    "formSectionsPO" : []
  },
  "showCurrencySymbol" : "1",
  "langCode" : "USEN",
  "monitorPO" : [],
  "jsResourceExists" : "0",
  "monitorPOCount" : 0,
  "isFormReadOnly" : false,
  "javascriptLibraryPO" : [],
  "currencyLangCode" : "USEN",
  "formSectionPOCount" : 4,
  "hasServerError" : false,
  "javascriptLibraryPOCount" : 0,
  "errorMessagePO" : {},
  "javascriptCallPOCount" : 1,
  "javascriptCallPO" : []
}
```

応答ペイロードの説明

- **javascriptFunctionDefinitionsPO** : この本文には、サービスによって消費される Java スクリプト、DDR、CR が含まれます。
- **formPO** : ペイロードのメイン属性です。このペイロードでは、すべてのフォーム内容の詳細がメタデータ形式で表示されます。
- **メソッド** : フォームを送信する Post メソッド。
- **ページ** : サービスに関連付けられているすべてのサービス フォーム名が含まれます。
- **FormSectionPO** : [全般 (General) ] プロパティと [表示プロパティ (Display Properties) ] コンテンツがすべて含まれます。
- **キー** : サービス フォームのキャプション
- **formFieldPO** : すべてのフォーム フィールド レベルのコンテンツがこの本文の下に表示されます。
- **Id** : ディレクトリの名前
- **値** : デフォルト値が表示されます
- **maxLength** : フィールドの長さ (ディクショナリの作成時に定義)
- **fieldDataType** : フィールドのデータ型 (ディクショナリの作成時に定義)
- **inputType** : フィールドのタイプ (単一選択、複数選択、ラジオ、チェックボックスなど)

- **sliderType** : 入力タイプがスライダの場合、スライダ タイプは範囲になります。
- **スケール** : スライダで範囲が設定されている場合に値が表示されます
- **validatorPO-** この値は、範囲を **true** に設定したときに設定され、最小値と最大値は同じ本文で定義されます。
- **[GenerateUnique] : [Generate unique value]** がオフの場合はゼロに設定し、オンの場合は **1** に設定し、**1** に設定した場合は [値 (Value) ] フィールドによって、デフォルト値がシステムによって生成された一意の値に置き換えられます。
- **命令テキスト** : フィールドのヘルプ テキストを表示します。
- **ValidatorPOCount** : これはバリデータの数です。検証範囲が確認され、必須情報が確認され、最小/最大値が指定される場合の例。カウントは **3** に設定され、**1** つがオフの場合は値が **2** に設定されます。
- **formFieldLabelPO – content** - これには、ファイルのラベルと高度なフォーマット詳細が含まれます
- **associatedControlPO-** これには、ボタンの追加、ボタン テキスト、URL、送信データの詳細が含まれます
- **showInGrid** : グリッドでない場合は **0** に設定し、グリッドの場合は **1** に設定します。
- **isMandatory** – 必須の場合は **true** に設定され、必須でない場合は **false** に設定されます
- **isHidden** : 非表示の場合は **true** に設定され、そうでない場合は **false** に設定されます。
- **errorMessagePO** : これには、フォームのロード中に **DDR** が失敗した場合の汎用的なエラー メッセージが含まれます。例 : サービス フォームのロード中に、以下の **DDR** 実行メッセージがユーザに表示されます。「**webservice DDR DemoDDR** の実行中にエラーが発生しました。システム管理者に連絡してください。」同じメッセージがこの本文に保存されます
- **javascriptCallIPO** : このコンテンツは、サービスによって消費され、イベント (**onload**、**submit**) を含み、関数に引数が含まれている関数名 です。この引数は、この本文に含まれます。

## API 応答の例

```
{
  "formPO" : {
    "method" : "post",
    "timeStamp" : "",
    "pages" : "[{"description":"Tenant Information Form for when Tenant Management is enabled","name":"Tenant-Information Form","sections":["XX_Tenant_Information_XX"]},{"description":"","name":"demo_SF","sections":["demo_SF","demo_GridSF"]}",
    "serviceID" : 240,
    "orderingMode" : 3,
    "showCurrencySymbol" : "1",
    "javascriptFunctionDefinitionsPO" : {
```



```

quired';\n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.VMSnapshot.Row')) docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMSnapshot.Row').style.display = 'none';\n\n} else
{\n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.VMSnapshot.Row')) docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMSnapshot.Row').style.display = 'table-row';\n\nvar colorFont =
'#66CC33';\nvar colorFont = '#669900';\nvar vcenterAddMsg = 'VMware vCenter Element Available';\n\nif(vcenterTotal < 1){\n\nvar
colorFont = '#B40000';\n\nvar vcenterAddMsg = 'VMware vCenter Element Required';\n\n}\n\n}\n\n var vcenterMessage = '<span
style=\"color : '+colorFont+';\">'+vcenterAddMsg+'</span>';\n\nvmTempMessage.innerHTML = vcenterMes-
sage;\n\n}\n\n}\n\n}\n\n//=====
VIRTUAL OS ORDER-
ING\n\n//=====
\n\nif(document.getElementBy
Id('SIBDChangeSystemWideServiceOptions.VMOS.Message')){\n\n//=====
\n\n// NOW FOR THE MESSAGE DIS-
PLAY\n\n//=====
\n\nvar vmOSMessage = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMOS.Message');\n\nvar vmOSYesId = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMOS::1');\n\nvar vmOSNoId = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMOS::2');\n\nif(document.getElementById('vmOSMaxSnap')){\n\nocu-
ment.getElementById('vmOSMaxSnap').innerHTML = "; \n\ndocument.getElementById('vmOSMaxSnap').innerHTML = gOffer-
ing[7];\n\n}\n\n}\n\n}\n\nvar vmOSSnapVal = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot.Message_INFO').innerHTML;\n\nvar vmOSSnapYes
= document.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot::1');\n\nvar vmOSSnapNo = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot::2');\n\nvar yesSelect = 'checked';\n\nvar noSelect =
";\n\nif(vmOSSnapNo.checked){\n\n var yesSelect = "; \n\n var noSelect = 'checked';\n\n }\n\n var createChkBox =
'<br>&nbsp;&nbsp;&nbsp;<input type=\"radio\" id=\"SIBDChangeSystemWideServiceOptions.VMOSSnapshot::1\"
name=\"SIBDChangeSystemWideServiceOptions.VMOSSnapshot\" value=\"Yes\" style=\"vertical-align : middle;\"
fielddatatype=\"Text\" ishook=\"1\" onfocus=\"cc_f(this,event);\" onblur=\"cc_f(this,event);\" nalign=\"horizontal\" on-
change=\"cc_f(this,event);\" '+yesSelect+>Yes&nbsp;&nbsp;&nbsp;<input type=\"radio\"
id=\"SIBDChangeSystemWideServiceOptions.VMOSSnapshot::2\"
name=\"SIBDChangeSystemWideServiceOptions.VMOSSnapshot\" value=\"No\" style=\"vertical-align : middle;\" fielddatatype=\"Text\"
ishook=\"1\" onfocus=\"cc_f(this,event);\" onblur=\"cc_f(this,event);\" nalign=\"horizontal\" onchange=\"cc_f(this,event);\"
'+noSelect+>No';\n\n\ndocument.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot.Input').innerHTML =
";\n\n\ndocument.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot.Input').innerHTML = vmOSSnapVal
+createChkBox;\n\n\ndocument.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot.Message_INFO').style.color
= '#ffffff';\n\nif(vmOSNoId.checked){\n\nvar colorFont = '#888888';\n\nvar cspAddMsg = 'Cisco SP Element Not Required';\n\nvar
ucsAddMsg = 'UCS Manager Element Not Required';\n\nvar cspMessage = '<span style=\"color :
'+colorFont+';\">'+cspAddMsg+'</span>';\n\nvar ucsMessage = '<span style=\"color :
'+colorFont+';\">'+ucsAddMsg+'</span>';\n\n}\n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.VMSnapshot.Row')
) document.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot.Row').style.display = 'none';\n\n} else {\n\nvar
colorFont = '#66CC33';\n\nvar colorFont = '#669900';\n\nvar cspAddMsg = 'Cisco SP Element Availa-
ble';\n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.VMSnapshot.Row')) docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.VMOSSnapshot.Row').style.display = 'table-row';\n\nif(cspTotal <
1){\n\nvar colorFont = '#B40000';\n\nvar cspAddMsg = 'Cisco SP Element Required';\n\n}\n\n}\n\nvar cspMessage = '<span style=\"color :
'+colorFont+';\">'+cspAddMsg+'</span>';\n\nvar colorFont = '#66CC33';\n\nvar colorFont = '#669900';\n\nvar ucsAddMsg = 'UCS Manager
Element Available';\n\nif(ucsTotal < 1){\n\nvar colorFont = '#B40000';\n\nvar ucsAddMsg = 'UCS Manager Element Required';\n\n}\n\nvar
ucsMessage = '<span style=\"color : '+colorFont+';\">'+ucsAddMsg+'</span>';\n\n}\n\n}\n\n}\n\nvmOSMessage.innerHTML = ucsMes-
sage+cspMessage;\n\n}\n\n}\n\n}\n\n//=====
CHECK PHYSICAL SERVER ORDER-
ING\n\n//=====
\n\nif(document.getElementBy
Id('SIBDChangeSystemWideServiceOptions.PhysicalServerOrdering.Message')){\n\n//=====
\n\n// NOW FOR THE MESSAGE DIS-
PLAY\n\n//=====
\n\nvar physMessage = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.PhysicalServerOrdering.Message');\n\nvar physYesId = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.PhysicalServerOrdering::1');\n\nvar physNoId = docu-
ment.getElementById('SIBDChangeSystemWideServiceOptions.PhysicalServerOrdering::2');\n\nif(physNoId.checked){\n\nvar color-
Font = '#888888';\n\nvar cspAddMsg = 'Cisco SP Element Not Required';\n\nvar ucsAddMsg = 'UCS Manager Element Not Re-
quired';\n\nvar cspMessage = '<span style=\"color : '+colorFont+';\">'+cspAddMsg+'</span>';\n\nvar ucsMessage = '<span
style=\"color : '+colorFont+';\">'+ucsAddMsg+'</span>';\n\n} else {\n\nvar colorFont = '#66CC33';\n\nvar colorFont = '#669900';\n\nvar
cspAddMsg = 'Cisco SP Element Available';\n\nif(cspTotal < 1){\n\nvar colorFont = '#B40000';\n\nvar cspAddMsg = 'Cisco SP Element
Required';\n\n}\n\n}\n\nvar cspMessage = '<span style=\"color : '+colorFont+';\">'+cspAddMsg+'</span>';\n\nvar colorFont =
'#66CC33';\n\nvar colorFont = '#669900';\n\nvar ucsAddMsg = 'UCS Manager Element Available';\n\nif(ucsTotal < 1){\n\nvar colorFont =
'#B40000';\n\nvar ucsAddMsg = 'UCS Manager Element Required';\n\n}\n\nvar ucsMessage = '<span style=\"color :
'+colorFont+';\">'+ucsAddMsg+'</span>';\n\n}\n\n}\n\n}\n\nphysMessage.innerHTML = ucsMes-
sage+cspMessage;\n\n}\n\n}\n\n}\n\n//=====

```

## CHECK ESXi PROVISION-

```
ING\n//===== \n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.ESXiProvisioning.Message')){\n\n//=====
===== \n// NOW FOR THE MESSAGE DIS-
```

```
PLAY\n//===== \nvar esxiMessage = document.getElementById('SIBDChangeSystemWideServiceOptions.ESXiProvisioning.Message');\nvar esxiYesId = document.getElementById('SIBDChangeSystemWideServiceOptions.ESXiProvisioning::1');\nvar esxiNold= document.getElementById('SIBDChangeSystemWideServiceOptions.ESXiProvisioning::2');\n\nif(esxiNold.checked){\n\nvar colorFont = '#888888';\nvar vcenterAddMsg = 'VMware vCenter Element Not Required';\n\n} else {\n\nvar colorFont = '#66CC33';\nvar colorFont = '#669900';\n\nvar vcenterAddMsg = 'VMware vCenter Element Available';\n\nif(vcenterTotal < 1){\n\nvar colorFont = '#B40000';\n\nvar vcenterAddMsg = 'VMware vCenter Element Required';\n\n}\n\n}\n\nvar vcenterMessage = '<span style=\n"color : '+colorFont+';\n">'+vcenterAddMsg+'</span>';\n\nesxiMessage.innerHTML = vcenterMessage;\n\n}\n\n}\n\n\n//===== \n// CHECK
```

## SHARED ZONE ORDER-

```
ING\n//===== \n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.SharedZone.Message')){\n\n//=====
===== \n// NOW FOR THE MESSAGE DIS-
```

```
PLAY\n//===== \nvar sharedZoneMessage = document.getElementById('SIBDChangeSystemWideServiceOptions.SharedZone.Message');\nvar szYesId = document.getElementById('SIBDChangeSystemWideServiceOptions.SharedZone::1');\nvar szNold= document.getElementById('SIBDChangeSystemWideServiceOptions.SharedZone::2');\n\nif(szNold.checked){\n\nvar colorFont = '#888888';\nvar vcenterAddMsg = 'VMware vCenter Element Not Required';\n\n} else {\n\nvar colorFont = '#66CC33';\nvar colorFont = '#669900';\n\nvar vcenterAddMsg = 'VMware vCenter Element Available';\n\nif(vcenterTotal < 1){\n\nvar colorFont = '#B40000';\n\nvar vcenterAddMsg = 'VMware vCenter Element Required';\n\n}\n\n}\n\nvar vcenterMessage = '<span style=\n"color : '+colorFont+';\n">'+vcenterAddMsg+'</span>';\n\nsharedZoneMessage.innerHTML = vcenterMessage;\n\n}\n\n}\n\n\n//===== \n// CHECK VDC
```

## ORDER-

```
ING\n//===== \n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.VDCOrdering.Message')){\n\n//=====
===== \n// NOW FOR THE MESSAGE DIS-
```

```
PLAY\n//===== \nvar vdcMessage = document.getElementById('SIBDChangeSystemWideServiceOptions.VDCOrdering.Message');\nvar vdcYesId = document.getElementById('SIBDChangeSystemWideServiceOptions.VDCOrdering::1');\nvar vdcNold= document.getElementById('SIBDChangeSystemWideServiceOptions.VDCOrdering::2');\n\nif(vdcNold.checked){\n\nvar colorFont = '#888888';\nvar vcenterAddMsg = 'VMware vCenter Element Not Required';\n\n} else {\n\nvar colorFont = '#66CC33';\nvar colorFont = '#669900';\n\nvar vcenterAddMsg = 'VMware vCenter Element Available';\n\nif(vcenterTotal < 1){\n\nvar colorFont = '#B40000';\n\nvar vcenterAddMsg = 'VMware vCenter Element Required';\n\n}\n\n}\n\nvar vcenterMessage = '<span style=\n"color : '+colorFont+';\n">'+vcenterAddMsg+'</span>';\n\nvdcMessage.innerHTML = vcenterMessage;\n\n}\n\n}\n\n\n//===== \n// SHOW / HIDE
```

## SUBMIT AND RESET BUT-

```
TONS\n//===== \n\nif(document.getElementById('SIBDChangeSystemWideServiceOptions.VDCOrdering.Message') && document.getElementById('SIBDChangeSystemWideServiceOptions.PhysicalServerOrdering.Message')){\n\nndocument.getElementById('c
```

```
header').style.visibility=\n"visible";\n\nndocument.getElementById('cfooter').style.visibility=\n"visible";\n\n\n/alert('CSP : '+cspTotal+ ' | UCS : '+ucsTotal+ ' | vCenter : '+vcenterTotal+ ' | p : '+p);\n\n\nif((vmOSYesId.checked && (cspTotal == 0 || ucsTotal == 0 )) || (physYesId.checked && (cspTotal == 0 || ucsTotal == 0 )) || ((vmTempYesId.checked || esxiYesId.checked || szYesId.checked || vdcYesId.checked ) && vcenterTotal ==
```

```
0) ){\n\nndocument.getElementById('cheader').style.visibility=\n"hidden";\n\nndocument.getElementById('cfooter').style.visibility=\n"hidden";\n\n}\n\n}\n\n}\n\n\nfunction Allstate_Common_OnSubmit()\n\n{\n\nProcessFormOnSubmit();\n\n}\n\n\n//This is for Form ID 2 \n\n/f11112 is for rule : ShowNameField \n\nvar f11112 = function f11112(url, divName, source, eventCopy, index, arr) { var local_rule_id = encodeURIComponent("WcHmG8SJ/H1yYWnT13lmbg=="); \n\n\nvar result = true; \n\n\n// it is a conditional rule \n\n\n// loop thro the conditions list \n\n\n// loop thro the actions list \n\n\nif ( \n\n( (eventCopy && eventCopy.srcElement && eventCopy.srcElement.name == 'XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX' && getValueFromIsfCode(eventCopy.srcElement) == getProperValueForLiteral('1,true,serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX)) || (!(eventCopy && eventCopy.srcElement && eventCopy.srcElement.name == 'XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX') && serviceForm.XX_Tenant_Information_XX && serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX && getProperValueForField(getSafeValue(serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX), true, serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX) == getProperValueForLiteral('1,true,serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX')) \n\n) \n\n) \n\n{ \n\nvar actions = []; \n\nvar actionValues = []; \n\nactionValues[0] = resolveSFNamespace("\n", "\n#\n", true, true); \n\nactions[0] = function() {if (serviceForm.XX_Tenant_Information_XX && serviceForm.XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX) serviceForm.XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX.setVisible(true);}; \n\nactionValues[1] = resolveSFNamespace("\n",
```

```

\'#\', true, true); \n actions[1] = function() {if ( serviceForm.XX_Tenant_Information_XX && ser-
viceForm.XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX) ser-
viceForm.XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX.setMandatory(true);}; \n result = runActions(actions, source,
eventCopy, index, arr); \n } \n return result; \n} \n//f11113 is for rule : HideNameField \nvar f11113 = function f11113(url, divName,
source, eventCopy, index, arr) { var local_rule_id = encodeURIComponent(\'\wgfRPAIcqeZh3jpbZm7wLQ==\'); \n \n var result = true; \n
// it is a conditional rule \n // loop thro the conditions list \n // loop thro the actions list \n if ( \n ( (eventCopy && eventCopy.srcElement
&& eventCopy.srcElement.name == \'XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX\' && getProperValueFromI-
fCode(eventCopy.srcElement) == getProperValueFor-
Literal(\'0,true,serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX)) || (!(eventCopy && eventCopy.srcElement &&
eventCopy.srcElement.name == \'XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX\' && ser-
viceForm.XX_Tenant_Information_XX && serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX && getProperVal-
ueForField(getSafeValue(serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX), true, ser-
viceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX) == getProperValueFor-
Literal(\'0,true,serviceForm.XX_Tenant_Information_XX.XX_XX_TMEEnabled_XX_XX\' ) ) \n ) \n { \n var actions = []; \n var actionValues
= []; \n actionValues[0] = resolveSFNamespace(\'\', \'#\', true, true); \n actions[0] = function() {if (ser-
viceForm.XX_Tenant_Information_XX && serviceForm.XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX) ser-
viceForm.XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX.setMandatory(false);}; \n actionValues[1] = re-
solveSFNamespace(\'\', \'#\', true, true); \n actions[1] = function() {if (serviceForm.XX_Tenant_Information_XX) ser-
viceForm.XX_Tenant_Information_XX.setVisible(false);}; \n result = runActions(actions, source, eventCopy, index, arr); \n } \n return
result; \n} \n \n//type_of_method_array will hold what type of rule it is. これは DDR コールバックと、CR の後に \n//
\'checkAndFireNext\' メソッドを呼び出す必要があることを認識するために必要です。 \n// could have been done as part of the previ-
ous rule list loop, doing it here for easier understanding \nif ( !type_of_method_array ) \n type_of_method_array = new Object(); \n
type_of_method_array[f11112] = \'CR\'; \n type_of_method_array[f11113] = \'CR\'; \n \n// We know from the event to rules association
table, the following rules get fired \n// in a specific order for a specific event \nif ( onload_on_serviceForm ) \n on-
load_on_serviceForm.push(f11112); \nelse { \n var onload_on_serviceForm = new Array; \n onload_on_serviceForm.push(f11112); \n}
\nif ( onload_on_serviceForm ) \n onload_on_serviceForm.push(f11113); \nelse { \n var onload_on_serviceForm = new Array; \n on-
load_on_serviceForm.push(f11113); \n} \n \n// the following is the event to the rules map \nif ( ! event_to_rules_map ) \n{ \n alert(\'the
event_to_rules_map is not defined\'); \n event_to_rules_map = new Object(); \n} \n \nevent_to_rules_map[\'onload_serviceForm\'] =
onload_on_serviceForm; \nevent_to_rules_map[\'onload_serviceForm\'] = onload_on_serviceForm; \n
\n//////////////////////////////////// \n \nvar formEventBeenThere = {}; \n
\n//////////////////////////////////// \n \n//This is for Form ID 157 \n//f11116 is for rule :
DDRUSERNAME \nvar f11116 = function f11116(url, divName, source, eventCopy, index, arr) { var local_rule_id = encodeURICompo-
nent(\'\ZBUI1h+RqWFdYG/IAYr8AQ==\'); \n \n var result = true; \n var filters = new Array(); \n var initiatorFilter = new Object(); \n initia-
torFilter.key = \'Requisition.InitiatorID\'; \n initiatorFilter.value = ReqInitiatorID; \n filters.push(initiatorFilter); \n \n var customerFilter =
new Object(); \n customerFilter.key = \'Requisition.CustomerID\'; \n customerFilter.value = ReqCustomerID; \n fil-
ters.push(customerFilter); \n \n var serviceldFilter = new Object(); \n serviceldFilter.key = \'ServiceForm.Serviceld\'; \n serviceldFil-
ter.value = ServiceID; \n filters.push(serviceldFilter); \n \n var filter_json = \'\'; \n \n if(filters.length > 0) \n filter_json =
JSON.stringify(filters); \n \n filter_json = encodeURIComponent(filter_json); \n \n var qStr =
\'formAction=executeFormRule&ruleParameters=\'+ filter_json + \'&ruleId=\'+ local_rule_id;\t
\n \n tcommonDDRCall(localBasePath+\'myservices/executeFormRules.do\', null, source, eventCopy, index, arr, 11116, qStr); \n return
result; \n} \n//f11117 is for rule : CRPSCUSERS \nvar f11117 = function f11117(url, divName, source, eventCopy, index, arr) { var loc-
al_rule_id = encodeURIComponent(\'\xFceFrjvgvz95Mwt/nb4zw==\'); \n \n var result = true; \n // it is a conditional rule \n // loop thro the
conditions list \n // loop thro the actions list \n if ( \n true \n ) \n { \n var actions = []; \n var actionValues = []; \n actionValues[0] = re-
solveSFNamespace(\'\PSCUSER\', \'#\', true, true); \n actions[0] = function() {if (serviceForm.demo_SF && ser-
viceForm.demo_SF.pscusers) serviceForm.demo_SF.pscusers.setValueByCRCheckError(eventCopy, \'PSCUSER\', \'PSCUSER\',
false);}; \n result = runActions(actions, source, eventCopy, index, arr); \n } \n return result; \n} \n \n//type_of_method_array will hold what
type of rule it is. これは DDR コールバックと、CR の後に \n// \'checkAndFireNext\' メソッドを呼び出す必要があることを認識するた
めに必要です。 \n// could have been done as part of the previous rule list loop, doing it here for easier understanding \nif
( !type_of_method_array ) \n type_of_method_array = new Object(); \n type_of_method_array[f11116] = \'DDR\'; \n
type_of_method_array[f11117] = \'CR\'; \n \n// We know from the event to rules association table, the following rules get fired \n// in a
specific order for a specific event \nif ( onsubmit_on_serviceForm ) \n onsubmit_on_serviceForm.push(f11117); \nelse { \n var onsub-
mit_on_serviceForm = new Array; \n onsubmit_on_serviceForm.push(f11117); \n} \nif ( onload_on_serviceForm ) \n on-
load_on_serviceForm.push(f11116); \nelse { \n var onload_on_serviceForm = new Array; \n onload_on_serviceForm.push(f11116); \n}
\nif ( onload_on_serviceForm ) \n onload_on_serviceForm.push(f11117); \nelse { \n var onload_on_serviceForm = new Array; \n on-
load_on_serviceForm.push(f11117); \n} \nif ( onunload_on_serviceForm ) \n onunload_on_serviceForm.push(f11117); \nelse { \n var
onunload_on_serviceForm = new Array; \n onunload_on_serviceForm.push(f11117); \n} \n \n// the following is the event to the rules
map \nif ( ! event_to_rules_map ) \n{ \n alert(\'the event_to_rules_map is not defined\'); \n event_to_rules_map = new Object(); \n} \n
\n \nevent_to_rules_map[\'onsubmit_serviceForm\'] = onsubmit_on_serviceForm; \nevent_to_rules_map[\'onload_serviceForm\'] = on-
load_on_serviceForm; \nevent_to_rules_map[\'onload_serviceForm\'] = onload_on_serviceForm;
\n \nevent_to_rules_map[\'onunload_serviceForm\'] = onunload_on_serviceForm; \n
\n//////////////////////////////////// \n \nvar formEventBeenThere = {}; \n
\n//////////////////////////////////// \n \n \n \n// sourceURL=jsstream.js\n

```

```

    },
    "formSectionPO" : [
        {
            "name" : {
                "uiresource" : {
                    "key" : "Select team",
                    "keyType" : "preprocess",
                    "escapeJS" : "false"
                }
            },
            "id" : "XX_Tenant_Information_XX",
            "autoDataRetrieval" : false,
            "templateTypeId" : "1",
            "maxRows" : 0,
            "mdrRecordID" : 0,
            "displayRows" : 0,
            "formFieldPO" : [
                {
                    "value" : "1",
                    "id" : "XX_Tenant_Information_XX.XX_XX_TMEnabled_XX_XX",
                    "scale" : 0,
                    "maxLength" : 512,
                    "inputType" : "hidden",
                    "fieldDataType" : "Text",
                    "sliderType" : "basic",
                    "formFieldLabelPO" : {
                        "content" : "XX_XX_TMEnabled_XX_XX"
                    },
                    "generateUnique" : 0,
                    "serverEditableOnly" : false,
                    "enableTypeahead" : 0,
                    "typeaheadMaxSearch" : 0,
                    "presentationDataType" : {
                        "hidden" : {
                            "name" :
                                "XX_Tenant_Information_XX.XX_XX_TMEnabled_XX_XX",
                            "value" : "1",

```



```

        "fieldDataType" : "Text",
        "controlName" : "nshidden",
        "isEditable" : "true"
    }
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : false,
"showInGrid" : 0,
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 0,
"javascriptCallPO" : [],
"validatorPO" : [],
"isEditable" : true,
"isHidden" : true
},
{
    "value" : "TeamVspancha",
    "id" : "XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX",
    "scale" : 0,
    "maxLength" : 100,
    "inputType" : "text",
    "fieldDataType" : "Tenant",
    "sliderType" : "basic",
    "formFieldLabelPO" : {
        "content" : "TeamName"
    },
    "generateUnique" : 0,
    "serverEditableOnly" : false,
    "enableTypeahead" : 0,
    "instructionalText" : "Please select the tenant to which you want to order this
service for.",
    "typeaheadMaxSearch" : 0,

```

```

        "presentationDataType" : {
            "tenant" : {
                "length" : "80",
                "name" :
"XX_Tenant_Information_XX.XX_XX_TeamName_XX_XX",
                "value" : "TeamVspancha",
                "fieldDataType" : "Tenant",
                "controlName" : "nstenant",
                "idValue" : "",
                "isEditable" : "true"
            }
        },
        "typeaheadMinStart" : 0,
        "isSecure" : 0,
        "isMandatory" : false,
        "showInGrid" : 0,
        "hasError" : false,
        "javascriptCallPOCount" : 0,
        "errorMessagePO" : {
            "uiresourceCount" : 0,
            "uiresource" : []
        },
        "validatorPOCount" : 0,
        "javascriptCallPO" : [],
        "validatorPO" : [],
        "isEditable" : true,
        "isHidden" : false
    }
},
"gridEditable" : false,
"displayAsGrid" : false,
"formFieldPOCount" : 2,
"dictionaryAccess" : true,
"isVisible" : true
},
{
    "name" : {

```

```
    "uiresource" : {
      "key" : "demo_SF",
      "keyType" : "preprocess",
      "escapeJS" : "false"
    }
  },
  "id" : "demo_SF",
  "autoDataRetrieval" : false,
  "templateTypeId" : "1",
  "maxRows" : 0,
  "mdrRecordID" : 0,
  "displayRows" : 0,
  "formFieldPO" : [
    {
      "value" : "Please enter the user name",
      "id" : "demo_SF.Name",
      "scale" : 0,
      "maxLength" : 50,
      "inputType" : "text",
      "fieldDataType" : "Text",
      "sliderType" : "basic",
      "formFieldLabelPO" : {
        "content" : "Name"
      },
      "generateUnique" : 0,
      "serverEditableOnly" : false,
      "enableTypeahead" : 0,
      "instructionalText" : "Help Text",
      "typeaheadMaxSearch" : 0,
      "presentationDataType" : {
        "textBox" : {
          "length" : "50",
          "name" : "demo_SF.Name",
          "value" : "Please enter the user name",
          "height" : "5",
          "fieldDataType" : "Text",
          "controlName" : "nstextbox",
```

```

        "isEditable" : "true"
    }
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : true,
"showInGrid" : 0,
"hasError" : false,
"associatedControlPO" : {
    "url" :
"http://10.78.0.247:8080/RequestCenter/organizationdesigner/selectperson.do?fromPage=ABC&search_pattern=&firstName=&last
Name=&p_data=customerId&p_disp=customerName&use_ou=true&o_data=orgUnitId&o_disp=costAssignedToBU&retfunc=setMan
ager",
    "label" : "Select Person",
    "sendData" : true
},
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 3,
"javascriptCallPO" : [],
"validatorPO" : [
    {
        "type" : null,
        "param" : "true",
        "validatorMessagePO" : {
            "uiresource" : {
                "key" :
"common.alertmsgs.plsEnterAppropriateInfo",
                "keyType" : "friendly",
                "escapeJS" : "false"
            }
        }
    }
],
{
    "type" : "maximum",
    "param" : "10",

```

```

        "validatorMessagePO" : {
            "uresource" : {
                "key" :
                    "common.alertmsgs.plsEnterValueLessThan",
                "keyType" : "friendly",
                "escapeJS" : "false"
            }
        }
    },
    {
        "type" : "minimum",
        "param" : "1",
        "validatorMessagePO" : {
            "uresource" : {
                "key" :
                    "common.alertmsgs.plsEnterValueGreaterThan",
                "keyType" : "friendly",
                "escapeJS" : "false"
            }
        }
    }
],
"isEditable" : true,
"isHidden" : false
},
{
    "value" : "may16user_6, may18user_c, may20user_L1, rkholiyav_j1, rkho-
lyav_k1, soa7_x5, rkuser_B1, duserff94_y7, new_y3, soa_z3, dallaire_-9, fff_k3, NSAPIIUserNew_y5, shashi_qf, sanjajag_-3,
uhosursh_34, NSAPIIUser01_V3, mgujar_.3, dinkumar_X3, ipestel_Y3",
    "id" : "demo_SF.username",
    "scale" : 0,
    "maxLength" : 5000,
    "inputType" : "text",
    "fieldDataType" : "Text",
    "formFieldLabelPO" : {
        "content" : "username"
    },
    "generateUnique" : 0,
    "serverEditableOnly" : false,

```

```
    "enableTypeahead" : 0,
    "typeaheadMaxSearch" : 0,
    "presentationDataType" : {
      "textBox" : {
        "length" : "80",
        "name" : "demo_SF.username",
        "value" : "may16user_6, may18user_c, may20user_L1,
rkholjav_j1, rkholjav_k1, soa7_x5, rkuser_B1, duserff94_y7, new_y3, soa_z3, dallaire_-9, fff_k3, NSAPIIUserNew_y5, shashi_qf,
sanjajag_-3, uhosursh_34, NSAPIIUser01_V3, mgujar_3, dinkumar_X3, ipestel_Y3",
        "height" : "5",
        "fieldDataType" : "Text",
        "controlName" : "nstextbox",
        "isEditable" : "true"
      }
    },
    "typeaheadMinStart" : 0,
    "isSecure" : 0,
    "isMandatory" : false,
    "showInGrid" : 0,
    "hasError" : false,
    "javascriptCallPOCount" : 0,
    "errorMessagePO" : {
      "uiresourceCount" : 0,
      "uiresource" : []
    },
    "validatorPOCount" : 0,
    "javascriptCallIPO" : [],
    "validatorPO" : [],
    "isEditable" : true,
    "isHidden" : false
  },
  {
    "value" : "PSCUSER",
    "id" : "demo_SF.pscusers",
    "scale" : 0,
    "maxLength" : 500,
    "inputType" : "text",
    "fieldDataType" : "Text",
```

```
"formFieldLabelPO" : {
    "content" : "pscusers"
},
"generateUnique" : 0,
"serverEditableOnly" : false,
"enableTypeahead" : 0,
"typeaheadMaxSearch" : 0,
"presentationDataType" : {
    "textBox" : {
        "length" : "80",
        "name" : "demo_SF.pscusers",
        "value" : "PSCUSER",
        "height" : "5",
        "fieldDataType" : "Text",
        "controlName" : "nstextbox",
        "isEditable" : "true"
    }
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : false,
"showInGrid" : 0,
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 0,
"javascriptCallPO" : [],
"validatorPO" : [],
"isEditable" : true,
"isHidden" : false
},
{
    "value" : "",
    "id" : "demo_SF.Country",
```

```
"scale" : 0,
"maxLength" : 50,
"inputType" : "select",
"fieldDataType" : "Text",
"sliderType" : "basic",
"formFieldLabelPO" : {
    "content" : "Country"
},
"generateUnique" : 0,
"serverEditableOnly" : false,
"enableTypeahead" : 0,
"typeaheadMaxSearch" : 0,
"presentationDataType" : {
    "multiSelectBox" : {
        "length" : "50",
        "name" : "demo_SF.Country",
        "value" : "",
        "selectOptionsCount" : 9,
        "disabled" : "false",
        "label" : "",
        "height" : "5",
        "fieldDataType" : "Text",
        "selectOptions" : [
            {
                "name" : "USA",
                "value" : "USA",
                "isSelected" : false,
                "controlName" : "nsselectoptions",
                "isEditable" : true
            },
            {
                "name" : "UK",
                "value" : "UK",
                "isSelected" : false,
                "controlName" : "nsselectoptions",
                "isEditable" : true
            }
        ]
    }
}
```



```
{
  "name" : "IN",
  "value" : "IN",
  "isSelected" : false,
  "controlName" : "nsselectoptions",
  "isEditable" : true
},
{
  "name" : "JP",
  "value" : "JP",
  "isSelected" : false,
  "controlName" : "nsselectoptions",
  "isEditable" : true
},
{
  "name" : "FR",
  "value" : "FR",
  "isSelected" : false,
  "controlName" : "nsselectoptions",
  "isEditable" : true
},
{
  "name" : "CA",
  "value" : "CA",
  "isSelected" : false,
  "controlName" : "nsselectoptions",
  "isEditable" : true
},
{
  "name" : "SA",
  "value" : "SA",
  "isSelected" : false,
  "controlName" : "nsselectoptions",
  "isEditable" : true
},
{
  "name" : "ENG",
  "value" : "ENG",
```

```
        "isSelected" : false,
        "controlName" : "nsselectoptions",
        "isEditable" : true
    },
    {
        "name" : "EUR",
        "value" : "EUR",
        "isSelected" : false,
        "controlName" : "nsselectoptions",
        "isEditable" : true
    }
],
"controlName" : "nsmulti",
"isEditable" : "true"
}
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : false,
"showInGrid" : 0,
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 0,
"javascriptCallPO" : [],
"validatorPO" : [],
"isEditable" : true,
"isHidden" : false
},
{
    "value" : "",
    "id" : "demo_SF.ProjectName",
    "scale" : 0,
    "maxLength" : 50,
```

```
"inputType" : "text",
"fieldDataType" : "Text",
"formFieldLabelPO" : {
    "content" : "ProjectName"
},
"generateUnique" : 0,
"serverEditableOnly" : false,
"enableTypeahead" : 0,
"typeaheadMaxSearch" : 0,
"presentationDataType" : {
    "textBox" : {
        "length" : "50",
        "name" : "demo_SF.ProjectName",
        "value" : "",
        "height" : "5",
        "fieldDataType" : "Text",
        "controlName" : "nstextbox",
        "isEditable" : "true"
    }
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : false,
"showInGrid" : 0,
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 0,
"javascriptCallPO" : [],
"validatorPO" : [],
"isEditable" : true,
"isHidden" : false
},
{
```

```
"value" : "",
"id" : "demo_SF.LaptopBrand",
"scale" : 0,
"maxLength" : 50,
"inputType" : "text",
"fieldDataType" : "Text",
"formFieldLabelPO" : {
    "content" : "LaptopBrand"
},
"generateUnique" : 0,
"serverEditableOnly" : false,
"enableTypeahead" : 0,
"typeaheadMaxSearch" : 0,
"presentationDataType" : {
    "textBox" : {
        "length" : "50",
        "name" : "demo_SF.LaptopBrand",
        "value" : "",
        "height" : "5",
        "fieldDataType" : "Text",
        "controlName" : "nstextbox",
        "isEditable" : "true"
    }
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : false,
"showInGrid" : 0,
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 0,
"javascriptCallPO" : [],
"validatorPO" : [],
```

```
"isEditable" : true,
"isHidden" : false
},
{
  "value" : "",
  "id" : "demo_SF.RAM",
  "scale" : 2,
  "maxLength" : 25,
  "inputType" : "slider",
  "fieldDataType" : "Money",
  "sliderType" : "range",
  "interval" : "10",
  "formFieldLabelPO" : {
    "content" : "RAM"
  },
  "generateUnique" : 0,
  "serverEditableOnly" : false,
  "enableTypeahead" : 0,
  "typeaheadMaxSearch" : 0,
  "presentationDataType" : {
    "slider" : {
      "length" : "50",
      "name" : "demo_SF.RAM",
      "value" : "",
      "selectOptionsCount" : 0,
      "maxValue" : "32",
      "minValue" : "4",
      "label" : "",
      "height" : "5",
      "fieldDataType" : "Money",
      "sliderType" : "range",
      "interval" : "10",
      "booleanTruePrompt" : "True",
      "booleanFalsePrompt" : "False",
      "selectOptions" : [],
      "controlName" : "nslider",
      "isEditable" : "true"
    }
  }
}
```

```

    }
  },
  "typeaheadMinStart" : 0,
  "booleanTruePrompt" : "True",
  "booleanFalsePrompt" : "False",
  "isSecure" : 0,
  "isMandatory" : true,
  "showInGrid" : 0,
  "hasError" : false,
  "javascriptCallPOCount" : 0,
  "errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
  },
  "validatorPOCount" : 3,
  "javascriptCallPO" : [],
  "validatorPO" : [
    {
      "type" : null,
      "param" : "true",
      "validatorMessagePO" : {
        "uiresource" : {
          "key" :
            "common.alertmsgs.plsEnterAppropriateInfo",
          "keyType" : "friendly",
          "escapeJS" : "false"
        }
      }
    },
    {
      "type" : "maximum",
      "param" : "32",
      "validatorMessagePO" : {
        "uiresource" : {
          "key" :
            "common.alertmsgs.plsEnterValueLessThan",
          "keyType" : "friendly",
          "escapeJS" : "false"
        }
      }
    }
  ]
}

```

```

        }
    },
    {
        "type" : "minimum",
        "param" : "4",
        "validatorMessagePO" : {
            "uresource" : {
                "key" :
                    "common.alertmsgs.plsEnterValueGreaterThan",
                "keyType" : "friendly",
                "escapeJS" : "false"
            }
        }
    },
    ],
    "isEditable" : true,
    "isHidden" : false
}
],
"gridEditable" : false,
"displayAsGrid" : false,
"formFieldPOCount" : 7,
"dictionaryAccess" : true,
"isVisible" : true
},
{
    "name" : {
        "uresource" : {
            "key" : "Demo_GridSF",
            "keyType" : "preprocess",
            "escapeJS" : "false"
        }
    },
    "id" : "demo_GridSF",
    "autoDataRetrieval" : false,
    "templateTypeId" : "1",
    "maxRows" : 5,

```

```
"recordSet" : {
    "content" : ""
},
"mdrRecordID" : 0,
"displayRows" : 5,
"formFieldPO" : [
    {
        "value" : "",
        "id" : "demo_GridSF.OS",
        "defaultValue" : "",
        "scale" : 0,
        "maxLength" : 50,
        "inputType" : "text",
        "fieldDataType" : "Text",
        "formFieldLabelPO" : {
            "content" : "OS"
        },
        "generateUnique" : 0,
        "serverEditableOnly" : false,
        "enableTypeahead" : 0,
        "instructionalText" : "",
        "typeaheadMaxSearch" : 0,
        "presentationDataType" : {
            "textBox" : {
                "length" : "50",
                "name" : "demo_GridSF.OS",
                "value" : "",
                "height" : "5",
                "fieldDataType" : "Text",
                "controlName" : "nstextbox",
                "isEditable" : "true"
            }
        },
        "typeaheadMinStart" : 0,
        "isSecure" : 0,
        "isMandatory" : false,
        "showInGrid" : 1,
    }
]
```



```
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uresourceCount" : 0,
    "uresource" : []
},
"validatorPOCount" : 0,
"javascriptCallIPO" : [],
"validatorPO" : [],
"isEditable" : true,
"isHidden" : false
},
{
    "value" : "",
    "id" : "demo_GridSF.Systembit",
    "defaultValue" : "",
    "scale" : 0,
    "maxLength" : 50,
    "inputType" : "text",
    "fieldDataType" : "Text",
    "formFieldLabelIPO" : {
        "content" : "Systembit"
    },
    "generateUnique" : 0,
    "serverEditableOnly" : false,
    "enableTypeahead" : 0,
    "instructionalText" : "",
    "typeaheadMaxSearch" : 0,
    "presentationDataType" : {
        "textBox" : {
            "length" : "50",
            "name" : "demo_GridSF.Systembit",
            "value" : "",
            "height" : "5",
            "fieldDataType" : "Text",
            "controlName" : "nstextbox",
            "isEditable" : "true"
        }
    }
}
```

```
    }  
  },  
  "typeaheadMinStart" : 0,  
  "isSecure" : 0,  
  "isMandatory" : false,  
  "showInGrid" : 1,  
  "hasError" : false,  
  "javascriptCallPOCount" : 0,  
  "errorMessagePO" : {  
    "uiresourceCount" : 0,  
    "uiresource" : []  
  },  
  "validatorPOCount" : 0,  
  "javascriptCallPO" : [],  
  "validatorPO" : [],  
  "isEditable" : true,  
  "isHidden" : false  
},  
{  
  "value" : "",  
  "id" : "demo_GridSF.Version",  
  "defaultValue" : "",  
  "scale" : 0,  
  "maxLength" : 50,  
  "inputType" : "text",  
  "fieldDataType" : "Text",  
  "formFieldLabelPO" : {  
    "content" : "Version"  
  },  
  "generateUnique" : 0,  
  "serverEditableOnly" : false,  
  "enableTypeahead" : 0,  
  "instructionalText" : "",  
  "typeaheadMaxSearch" : 0,  
  "presentationDataType" : {  
    "textBox" : {  
      "length" : "50",
```

```
        "name" : "demo_GridSF.Version",
        "value" : "",
        "height" : "5",
        "fieldDataType" : "Text",
        "controlName" : "nstextbox",
        "isEditable" : "true"
    }
},
"typeaheadMinStart" : 0,
"isSecure" : 0,
"isMandatory" : false,
"showInGrid" : 1,
"hasError" : false,
"javascriptCallPOCount" : 0,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"validatorPOCount" : 0,
"javascriptCallPO" : [],
"validatorPO" : [],
"isEditable" : true,
"isHidden" : false
}
],
"gridEditable" : true,
"displayAsGrid" : true,
"formFieldPOCount" : 3,
"dictionaryAccess" : true,
"isVisible" : true
}
],
"javascriptCallPOCount" : 2,
"errorMessagePO" : {
    "uiresourceCount" : 0,
    "uiresource" : []
},
"javascriptCallPO" : [
```

```

        {
            "content" : "checkOrderingResources();",
            "event" : "submit"
        },
        {
            "content" : "Allstate_Common_OnSubmit(TestArg,TestArg1);",
            "event" : "unload"
        }
    ],
    "formSectionPOCount" : 3,
    "hasServerError" : false,
    "javascriptLibraryPOCount" : 0,
    "javascriptLibraryPO" : [],
    "jsResourceExists" : "0",
    "currencyLangCode" : "TWZH",
    "isFormReadOnly" : false,
    "monitorPOCount" : 0,
    "langCode" : "TWZH",
    "monitorPO" : []
}
}

```

### 既存の注文のサービス オーダー フォームのメタデータを取得する API

**前提条件：**以下の API を実行する前に、`newscale.properties` でパラメータを更新する必要があります。

```

#####
##Particle form rules js template path : service-form-rules-js-templates/formrules_javascript_tpl.ftl
formrule.js.template.path=service-form-rules-js-templates/formrules_javascript_tpl.ftl
#####

```

すべてのフォームの詳細は、以下の API からGETでき、応答は Json になります

- API - GETメソッド - `http://<Server_URL>:PORT/RequestCenter/nsapi/catalog/v1/requisitionentry/ <requisitionentry-id>/form`

以下は、ヘッダーで渡す必要があるパラメータです。

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

**RBAC アクセス:** サービスまたはサービス グループの注文権限を持つユーザは、この API を実行できます。

次のような API の応答:

```
{
  "formPO" : {
    "method" : "post",
    "timeStamp" : "",
    "pages" : "[{\\"description\\":\\"Tenant Information Form for when Tenant Management is enabled\\",\\"name\\":\\"Tenant-Information Form\\",\\"sections\\":[\"XX_Tenant_Information_XX\"]},{\\"description\\":\\"Attach Volume\\",\\"sections\\":[\"Volume_Action\\",\"VolumeInfo_RO_Action\\",\"VolumeInfo_Action\"]}]",
    "orderingMode" : 3,
    "servicelD" : 136,
    "javascriptFunctionDefinitionsPO" : {},
    "formSectionsPO" : []
  "showCurrencySymbol" : "1",
    "langCode" : "USEN",
    "monitorPO" : [],
    "jsResourceExists" : "0",
    "monitorPOCount" : 0,
    "isFormReadOnly" : false,
    "javascriptLibraryPO" : [],
    "currencyLangCode" : "USEN",
    "formSectionPOCount" : 4,
    "hasServerError" : false,
    "javascriptLibraryPOCount" : 0,
    "errorMessagePO" : {}
  "javascriptCallPOCount" : 1,
    "javascriptCallPO" : []
  }
}
```

応答ペイロードの説明

- **javascriptFunctionDefinitionsPO** : この本文には、サービスによって消費される Java スクリプト、DDR、CR が含まれます。
- **formPO** : ペイロードのメイン属性です。このペイロードでは、すべてのフォーム内容の詳細がメタデータ形式で表示されます。
- **メソッド** : フォームを送信する Post メソッド。
- **ページ** : サービスに関連付けられているすべてのサービス フォーム名が含まれます。
- **FormSectionPO** : [全般 (General) ] プロパティと [表示プロパティ (Display Properties) ] コンテンツがすべて含まれます。
- **キー** : サービス フォームのキャプション
- **formFieldPO** : すべてのフォーム フィールド レベルのコンテンツがこの本文の下に表示されます。
- **Id** : ディレクトリ の名前
- **値** : サービス注文中にパスされた値またはデフォルト値
- **maxLength** : フィールドの長さ (ディクショナリの作成時に定義)
- **fieldDataType** : フィールドのデータ型 (ディクショナリの作成時に定義)
- **inputType** : フィールドのタイプ (単一選択、複数選択、ラジオ、チェックボックスなど)
- **sliderType** : 入力タイプがスライダの場合、スライダ タイプは範囲になります。
- **スケール** : スライダで範囲が設定されている場合に値が表示されます
- **validatorPO**- この値は、範囲を true に設定したときに設定され、最小値と最大値は同じ本文で定義されます。

- **[GenerateUnique]** : [Generate unique value] がオフの場合はゼロに設定し、オンの場合は 1 に設定し、1 に設定した場合は [値 (Value)] フィールドによって、デフォルト値がシステムによって生成された一意の値に置き換えられます。
- **命令テキスト** : フィールドのヘルプ テキストを表示します。
- **ValidatorPOCount** : これはバリデータの数です。検証範囲が確認され、必須情報が確認され、最小/最大値が指定される場合の例。カウントは 3 に設定され、1 つがオフの場合は値が 2 に設定されます。
- **formFieldLabelPO – content** - これには、ファイルのラベルと高度なフォーマット詳細が含まれます
- **associatedControlPO-** これには、ボタンの追加、ボタン テキスト、URL、送信データの詳細が含まれます
- **showInGrid** : グリッドでない場合は 0 に設定し、グリッドの場合は 1 に設定します。
- **isMandatory** – 必須の場合は true に設定され、必須でない場合は false に設定されます
- **isHidden** : 非表示の場合は true に設定され、そうでない場合は false に設定されます。
- **errorMessagePO** : これには、フォームのロード中に DDR が失敗した場合の汎用的なエラー メッセージが含まれます。  
例：サービス フォームのロード中に、以下の DDR 実行メッセージがユーザーに表示されます。「webservice DDR DemoDDR の実行中にエラーが発生しました。システム管理者に連絡してください。」同じメッセージがこの本文に保存されます
- **javascriptCallPO** : これには、サービスによって消費される関数名と、イベント (onload、submit)、および関数に引数があるかどうかが含まれます。この引数は、この本文に含まれます。

サービス名を使用したサービス詳細 UI へのアクセス

以前は、サービス詳細 UI URL は ServiceId に基づいていました。

例：サービス ID 11。

例の URL : <http://localhost:8080/RequestCenter/website/ServiceCatalogWebsite/application/offer.html?route=offer&id=11>

ServiceName をクエリ パラメータとしてパスすることで、[サービスの詳細 (Service Details)] ページにアクセスできるようになりました。

次の URL の名前クエリ パラメータを使用したサービス名 AppleiPad の例

例の URL :

<http://localhost:8080/RequestCenter/website/ServiceCatalogWebsite/application/offer.html?route=offer&name=Apple%20iPad>

### DDR コールを実行する API

サービス フォーム メタデータ API は、サービス内のさまざまなサービス フォームに設定されている DDR ルールのリストを返します。ルール ID は暗号化形式です。暗号化されたルール ID には特殊文字が含まれている可能性があるため、Base 64 エンコーディングを使用してルール ID をエンコードする必要があります。

注：DDR コールの API 実行は、フォーム レベル (ブラウザ側) のイベントにのみ適用されます。

DDR ルールを実行する API

メソッド : PUT

REST URL : /RequestCenter/nsapi/catalog/v1/service/Id/rule/<encoded ruleId>

DDR 実行の値を含む要求ペイロード。

サンプル 要求ペイロード :

```
{
  "key": "Requisition.InitiatorID", "value": "1",
  "key": "Requisition.CustomerID", "value": "1",
  "key": "ServiceForm.ServiceId", "value": "169"
}
```

フォームのロード中イベントに関連付けられている DDR ルールの要求と応答の例

メソッド : PUT

## REST URL :

```
http://10.78.0.216:8080/RequestCenter/nsapi/catalog/v1/service/169/rule/RJZi0%2FXysRe6RdF%2BH1NWcQ%3D%3D
```

## サンプル要求ペイロード :

```
[{"key":"Requisition.InitiatorID","value":"1"},{"key":"Requisition.CustomerID","value":"1"},{"key":"ServiceForm.ServiceId","value":"169"}]
```

## サンプル応答ペイロード :

listOFValues :

key - Dictionary.Field

value : ディクショナリ フィールドの値。

listOFTargets:

DDR ルールの実行によって取得された値が入力されるディクショナリ フィールドのリスト。

応答 :

```
{
  "listOfValues" : [
    [
      {
        "key" : "Personal_Computer_Dictionary.Make"
        "value" : "Lenovo"
      },
      {
        "key" : "Personal_Computer_Dictionary.Name",
        "value" : "PC1"
      },
      {
        "key" : "Personal_Computer_Dictionary.NetName",
        "value" : "L1"
      },
      {
        "key" : "Personal_Computer_Dictionary.RAM",
        "value" : "64"
      }
    ]
  ],
  "listOfTargets" : [
    "Personal_Computer_Dictionary.Make",
    "Personal_Computer_Dictionary.Name",
    "Personal_Computer_Dictionary.NetName",
    "Personal_Computer_Dictionary.RAM"
  ]
}
```

```
    ]
}
```

### サービス オーダー フォーム アクション API

#### 要求 API を送信

要求を送信する API。すべての送信後 DDR が実行され、フィールド検証が実行され、エントリの検証に成功すると、要求が送信されます。

要求ペイロードには、要求エントリと、送信する値を含む対応するディクショナリ フィールドのリストが含まれます。

メソッド : POST

REST URL : <http://10.78.0.247:8080/RequestCenter/nsapi/catalog/v1/requisition>

サンプル要求ペイロード :

```
{
  "requisition" : {
    "customerLoginName" : "admin",
    "services" : {
      {
        "name" : "Service_Grid",
        "quantity" : "1",
        "version" : "0",
        "dictionaries" : [
          {
            "name" : "XX_Tenant_Information_XX",
            "data" : {
              "XX_XX_TeamName_XX_XX" : "TeamA"
            }
          }
        ],
      },
      {
        "name" : "Dictioanary_Grid",
        "data" : [{
          "Text":"Test1",
          "Boolean" : "Yes",
          "SelectSingle" : "GSS1"
        }
      ],
    }
  }
}
```



```
    "Text" : "Test2",
    "Boolean" : "Yes",
    "SelectSingle" : "GSS1"
  },
  {
    "Text" : "Test3",
    "Boolean" : "Yes",
    "SelectSingle" : "GSS1"
  },
  {
    "Text" : "Test4",
    "Boolean" : "Yes",
    "SelectSingle" : "GSS1"
  },
  {
    "Text" : "Test5",
    "Boolean" : "Yes",
    "SelectSingle" : "GSS1"
  }
}
]
}
}
```

カートに追加する API

メソッド : PUT

REST URL : <http://10.78.0.247:8080/RequestCenter/nsapi/catalog/v1/requisition>

サンプル要求ペイロード :

```
{
  "requisition" : {
    "customerLoginName" : "admin",
    "services" : {
      {
        "name" : "Service_Feature",
        "quantity" : "1",
        "version" : "0",
```

```
"dictionaries" : [  
  {  
    "name" : "DictionaryFeature",  
    "data" : {  
      "Field_Text1" : "hello",  
      "Field_Number1" : "3"  
    }  
  }  
]  
}  
]
```

ドラフトとしてサービスを単独で保存する  
メソッド : POST

REST URL : <http://10.78.0.247:8080/RequestCenter/nsapi/catalog/v1/requisition?saveAsDraft=true>

サンプル要求ペイロード :

```
{  
  "requisition" : {  
    "customerLoginName" : "admin",  
    "services" : {  
      {  
        "name" : "Service_Feature",  
        "quantity" : "1",  
        "version" : "0",  
        "dictionaries" : [  
          {  
            "name" : "DictionaryFeature",  
            "data" : {  
              "Field_Text1" : "hello",  
              "Field_Number1" : "3"  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```

```
}
}
```

ドラフト内のサービスを含むドラフトとしてサービスを保存する

メソッド : PUT

REST URL : <http://10.78.0.247:8080/RequestCenter/nsapi/catalog/v1/requisition?saveAsDraft=true>

サンプル要求ペイロード :

```
{
  "requisition" : {
    "customerLoginName" : "admin",
    "services" : {
      {
        "name" : "Service_Feature",
        "quantity" : "1",
        "version" : "0",
        "dictionaries" : [
          {
            "name" : "DictionaryFeature",
            "data" : {
              "Field_Text1" : "hello",
              "Field_Number1" : "3"
            }
          }
        ]
      }
    ]
  }
}
```

### サービスをエクスポート/インポートする API

#### エクスポート サービス API

エクスポート サービス API はサービス ID のリストを取得し、サービスの詳細を xml にエクスポートします。

応答は、Catalog Deployer からのサービスのエクスポートで提供される xml に似ています。

- 複数の ServiceId (S1=241、S2=242、S3=243) のエクスポートは次のようにパスされます。

EndPoints :

<http://10.78.0.247:8080/RequestCenter/nsapi/rex/v1/definition/service?id=243&id=242&id=241&includeContainedServices=true>

メソッド : GET

- サービス S2 を囲むバンドル サービスであるサービス S1 があり、別のサービス S3 をエクスポートする場合、エンドポイントは次のようになります。

EndPoints :

<http://10.78.0.247:8080/RequestCenter/nsapi/rex/v1/definition/service?id=243&id=241&includeContainedServices=true>

メソッド : GET

- この場合、サービス S2 を囲むバンドル サービスであるサービス S1 をエクスポートする必要がありますが、S2 はエクスポートされません。エンドポイントは次のようになります。

EndPoints :

<http://10.78.0.247:8080/RequestCenter/nsapi/rex/v1/definition/service?id=243&id=241&includeContainedServices=false>

メソッド : GET

インポート サービス API

- エクスポートされた XML をインポートする場合、エンドポイントは次のようになります。

エンドポイント : <http://10.78.0.251:8080/RequestCenter/nsapi/rex/v1/definition/service>

メソッド : PUT

注 :

- エクスポート ファイルは改ざんやインポートを行わないでください。
- ユーザのエクスポート/インポートには SD 機能が必要です
- ユーザには WebServices- REX API アクセス機能が必要です。
- ユーザのエクスポート/インポートには、すぐに統合管理者ロールが必要です。
- includeContainedServices フラグはデフォルトで true です。

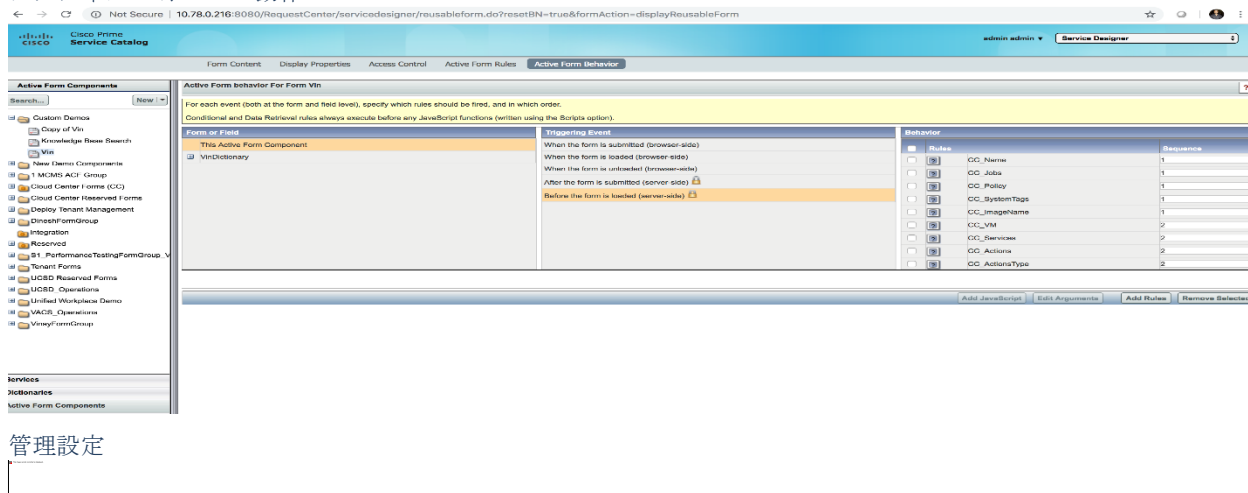
ユーザのエクスポート/インポートには次の権限があります

- 読み取り/書き込み定義 : サービス項目定義 「すべてのオブジェクト」
- 読み取り/書き込み定義-サービス項目グループ 「すべてのオブジェクト」

#### 並列スレッドでの独立した Web サービス DDR の実行

- この機能は、サービス フォームのロードに対して最適化するために、サーバ側のルールを並行して実行するプロビジョニングを提供します。
- 依存関係に基づいて、DDR ルールを順序付ける必要があります。たとえば、DDR2 が DDR1 の応答に依存する場合。DDR2 は DDR1 の後にシーケンスする必要があります。DDR2 に依存関係がない場合は、DDR1 と同じシーケンスを使用できます。これにより、両方のルールが並行して実行され、実行時間が最適化されます。以前は、DDR ルールは次々に実行されていました。
- これを行うには、**[アクティブ フォーム コンポーネント ( Active Form Components ) ] > [アクティブ フォーム動作 ( Active Form Behavior ) ] > [トリガー イベントの選択 ( フォームがロードされる前 ( サーバ側 ) ) ( Select Triggering Event ( Before the Form is Loaded(server-side)) ) ] > [ルールの追加および順序の更新 ( Add Rules and update the Sequence ) ]** から実行の順序を定義できます。
- 2 つ以上のルールが同じシーケンス番号を持つ場合、つまりグループ化されている場合、それらは並行して実行されません。このようなルールの例としては、グループ内のルールの結果に依存しないルールがあります。
- 要求ごとに生成されるスレッドの数は、管理設定から設定できます。
- [管理 ( Administration ) ]、[設定 ( Settings ) ]、[カスタム ( Customizations ) ]、[最大平行ルール ( Maximum Parallel rules ) ]** の順に選択して実行します。
- 「0」に設定すると、制限はなくなります (推奨されません)。
- デフォルトでは、値は 25 に設定されます (つまり、要求ごとに 25 個のスレッドが生成されます)。

## アクティブ フォームの動作



管理設定

## フォーム ルール javascript FTL の変更

- サービス フォーム ルールの実行 (DDR および CR) は、ISF 機能の実装に依存します。これは、設定されたイベントがトリガーされたときに実行される JavaScript コードです。この javascript は、サービス デザイナーで実行されるサービス フォーム ルール設定から生成されるサービス フォーム固有の JavaScript です。この JavaScript は、サービス フォームのメタ データ API 応答 <add hyperlink here> から取得できます。応答の javascriptFunctionDefinitionsPO プロパティに含まれています。このスクリプトを実行すると、サービスに固有のアクションを実行するようにイベント ハンドラが設定されます。
- フォーム ルール javascript は、サービス デザイナーで行われたルール設定を使用して、freemarker template (ftl) ファイルから生成されます。
- ftl ファイルの出荷バージョンは、次の場所にあります。  
/RequestCenter.war/WEB-INF/classes/service-form-rules-js-templates/formrules\_javascript\_tpl.ftl.
- 生成された JavaScript は、レンダラ フォーム ルール実行フローによって提供される ISF ライブラリを使用します。
- 新しいサービス フォーム レンダラの実装は、ftl と対応する ISF ライブラリを書き換えることで実現できます。

// isf\_methodArray ルール ID に対する保留ルールのタイプ (DDR または CR)

```
ISF.isf_methodArray();
```

// isf\_eventMap は、イベントおよびディクショナリ フィールドの詳細に対するキーとしてのルールに対して呼び出される javascript 関数の配列を保持します。

```
ISF.isf_eventMap();
```

```
/*
```

DDR 実行 API コールを作成する

フォーム : サービス フォーム メタ データおよび値を含みます

dictionaryParameters : 要求本文のディクショナリ パラメータ

encodedRuleId : エンコードされた暗号化ルール ID

```
*/
```

```
isf_ddd_call = (form, dictionaryParameters, encodedRuleId)
```

```
/*
```

フォーム：サービス フォーム メタ データおよび値を含みます

左オペランドと右オペランドを比較します。leftOperand/rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField/rightOperandsField です

\*/

```
isf_cr_equals = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

/\*

フォーム：サービス フォーム メタ データおよび値を含みます

左オペランドと右オペランドの大文字小文字を無視して比較します。leftOperand / rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です

\*/

```
isf_cr_equals_nocase = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

/\*

フォーム：サービス フォーム メタ データおよび値を含みます

左オペランドと右オペランドを比較します。leftOperand/rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField/rightOperandsField です

\*/

```
isf_cr_not_equals = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

/\*

フォーム：サービス フォーム メタ データおよび値を含みます

左オペランドと右オペランドを比較します。leftOperand/rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField/rightOperandsField です

\*/

```
isf_cr_gt = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

/\*

フォーム：サービス フォーム メタ データおよび値を含みます

左オペランドと右オペランドを比較します。leftOperand/rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField/rightOperandsField です

\*/

```
isf_cr_lt = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

/\*

フォーム：サービス フォーム メタ データおよび値を含みます

左オペランドと右オペランドを比較します。leftOperand/rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField/rightOperandsField です

\*/

```
isf_cr_gte = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

```
/*  
    フォーム：サービス フォーム メタ データおよび値を含みます  
    左オペランドと右オペランドを比較します。leftOperand/rightOperand がディクショナリ フィールド名またはリテラルの  
    場合、leftOperandsField/rightOperandsField です  
*/
```

```
isf_cr_lte = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

```
/*  
    フォーム：サービス フォーム メタ データおよび値を含みます  
    左のオペランドが右のオペランドから始まるかどうかをチェックします。leftOperand / rightOperand がディクショナリ  
    フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です  
*/
```

```
isf_cr_starts_with = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

```
/*  
    フォーム：サービス フォーム メタ データおよび値を含みます  
    左のオペランドが右のオペランドで終わるかどうかをチェックします。leftOperand/rightOperand がディクショナリ フィ  
    ルド名またはリテラルの場合、leftOperandsField/rightOperandsField です  
*/
```

```
isf_cr_ends_with = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)
```

```
/*  
    フォーム：サービス フォーム メタ データおよび値を含みます  
    leftOperand フィールドが undefined と等しくありません。leftOperand / rightOperand がディクショナリ フィールド名ま  
    たはリテラルの場合、leftOperandsField / rightOperandsField です  
*/
```

```
isf_cr_exists = (form, leftOperand, leftOperandsField)
```

```
/*  
    フォーム：サービス フォーム メタ データおよび値を含みます  
    isDisabled が左オペランド フィールドに true であるかどうか確認します。leftOperand / rightOperand がディクショナリ フ  
    イールド名またはリテラルの場合、leftOperandsField / rightOperandsField です  
*/
```

```
isf_cr_isDisabled = (form, leftOperand, leftOperandsField)
```

```
/*  
    フォーム：サービス フォーム メタ データおよび値を含みます  
    isDisabled が左のオペランドフィールドに false であるかどうかを確認します。leftOperand / rightOperand がディクショ  
    ナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です  
*/
```

```
*/
isf_cr_not_isDisabled = (form, leftOperand, leftOperandsField)

/*
    フォーム：サービス フォーム メタ データおよび値を含みます
    isHidden が左オペランドフィールドに true であるかどうか確認します。leftOperand / rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です
*/
isf_cr_isHidden = (form, leftOperand, leftOperandsField)

/*
    フォーム：サービス フォーム メタ データおよび値を含みます
    leftOperand が null に等しくないかどうかを確認します。leftOperand / rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です
*/
isf_cr_isNull = (form, leftOperand, leftOperandsField)

/*
    フォーム：サービス フォーム メタ データおよび値を含みます
    leftOperand に右オペランドが含まれているかどうかを確認します。leftOperand / rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です
*/
isf_cr_contains = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)

/*
    フォーム：サービス フォーム メタ データおよび値を含みます
    leftOperand に右オペランドが含まれていないかどうかをチェックします。leftOperand / rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です
*/
isf_cr_not_contains = (form, leftOperand, leftOperandsField, rightOperandsField, rightOperand)

/*
    フォーム：サービス フォーム メタ データおよび値を含みます
    isHidden が左オペランドフィールドに false であるかどうかをチェックします。leftOperand / rightOperand がディクショナリ フィールド名またはリテラルの場合、leftOperandsField / rightOperandsField です
*/
isf_cr_not_isHidden = (form, leftOperand, leftOperandsField)

/*
    フォーム：サービス フォーム メタ データおよび値を含みます
```



フォーム オブジェクトのディクショナリ フィールドに **isHidden** を設定します。 **elem** がフィールドのキーになります。

例 : **ele = DictionaryName.fieldName**

\*/

```
isf_cr_setHidden = (form, elem, onoff = false)
```

/\*

フォーム : サービス フォーム メタ データおよび値を含みます

フォーム オブジェクトのディクショナリ フィールドの値を設定します。 **elem**がフィールドのキーになります。 例 : **ele = DictionaryName.fieldName**

\*/

```
isf_cr_setValue = (form, elem, value = "")
```

/\*

フォーム : サービス フォーム メタ データおよび値を含みます

フォーム オブジェクトのディクショナリ フィールドの値を設定します。 **elem**がフィールドのキーになります。 例 : **ele = DictionaryName.fieldName**

\*/

```
isf_cr_setValue = (form, elem, value = "")
```

/\*

フォーム : サービス フォーム メタ データおよび値を含みます

フォーム オブジェクトのディクショナリ フィールドの **isMandatory** を設定します。 **elem**がフィールドのキーになります。 例 : **ele = DictionaryName.fieldName**

\*/

```
isf_cr_setMandatory = (form, elem, onoff = false)
```

/\*

フォーム : サービス フォーム メタ データおよび値を含みます

フォーム オブジェクトのディクショナリ フィールドの **isEditable** を設定します。 **elem**がフィールドのキーになります。 例 : **ele = DictionaryName.fieldName**

\*/

```
isf_cr_setDisabled = (form, elem, onoff = true)
```

/\*

フォーム : サービス フォーム メタ データおよび値を含みます

フォーム オブジェクトのディクショナリ フィールドにフォーカスを設定します。 **elem**がフィールドのキーになります。 例 : **ele = DictionaryName.fieldName**

\*/

```
isf_cr_setFocus = (form, elem)
```

/\*

```
    フォーム：サービス フォーム メタ データおよび値を含みます  
    メッセージとともにアラートポップアップを表示する
```

```
*/
```

```
isf_cr_alert = (form, msg)
```

```
/*
```

```
    フォーム：サービス フォーム メタ データおよび値を含みます  
    サービス フォームの送信の停止
```

```
*/
```

```
isf_cr_stopSubmission = (form)
```

```
/*
```

```
    ログインしたユーザがロールのメンバーであるかどうかを確認します
```

```
*/
```

```
isf_isMemberOfRole(roleId)
```

```
/*
```

```
    顧客がロールのメンバーであるかどうかを確認します
```

```
*/
```

```
isf_isCustomerMemberOfRole(roleId)
```

```
/*
```

```
    フィールドがグリッド フィールドかどうかを確認します
```

```
    フォーム：サービス フォーム メタ データおよび値を含みます
```

```
*/
```

```
isf_isGridField = (form, fieldKey)
```

```
/*
```

```
    デictionaryがフォームで使用可能かどうかを確認します
```

```
    フォーム：サービス フォーム メタ データおよび値を含みます
```

```
    dictionaryName
```

```
*/
```

```
isf_checkDictionaryExistence = (form, fieldKey)
```

**注：**この FTL および使用可能な ISF 関数の呼び出しは、開発中のすべてを網羅するものではなく、変更される可能性があります。

**UCSD VM プロビジョニング**

以前は、新しい VM がプロビジョニングされるか、障害が発生した VM が更新されるたびに、VDC の完全な同期が行われていました。

- ✓ **新しい動作** : 新しい VM がプロビジョニングされるか、または障害が発生した VM が更新されるときは、VM データが PSC に同期される場合のみ、完全な VDC 同期は発生しません。

**サービス/ユーザ コンテキストを取得する API**

オーダー フォーム コンテキスト API (GET API) は、サービス オーダー フォームをロードするために使用されるコンテキスト パラメータを提供します。

REST URL : <http://localhost:8080/RequestCenter/nsapi/catalog/v1/service/150/orderformcontext>

メソッド : GET

クエリ パラメータ : customerId

代理注文ユーザの場合、注文対象のユーザの顧客 ID を渡す必要があります。

<http://10.78.0.247:8080/RequestCenter/nsapi/catalog/v1/service/148/orderformcontext?customerId=15>

応答ペイロード :

応答ペイロード : -

```
{
  "Map" : {
    "siteAdminUser" : true,
    "EnableOOBAutoComplete" : false,
    "orderConfirmationDeliverToFormat" : "Customer(Login)",
    "allowAddAttachment" : true,
    "isNextGenUser" : true,
    "clickableHelpIcon" : false,
    "timeZone" : "America/Tijuana",
    "hideNRDicts" : false,
    "servicedescproperty" : false,
    "hideMonitorFlag" : false,
    "Serviceforelement.display.instructional helptext": false,
    "isMessageInfo" : "false",
    "DescriptionLimit" : "0",
    "nsBasePath" : "/RequestCenter/",
    "service" : {
      "serviceID" : 150,
      "name" : "testService1",
      "documentID" : 0,
      "estimatedCost" : 0.0,
```

```

        "expectedDuration" : 0.0,
        "hasPortal3" : true,
        "portalText3" : "<p>TEST3</p>\n",
        "isEntitlement" : 0,
        "isInactive" : 0,
        "priceDescription" : "",
        "shortPriceDescription" : "",
        "priceDisplaySchemaID" : 0,
        "pricingSchema" : 0,
        "revisionNumber" : 34,
        "allowFutureDelivery" : true,
        "isBundle" : false,
        "isTemplate" : false,
        "tenantRelevant" : true,
        "isOrderable" : true,
        "orderingMode" : 3,
        "computePrice" : false,
        "paginationViewMode" : 0,
        "isDefinedDuration" : false,
        "showOrderSummary" : false
    },
    "allowUpdateQty" : false,
    "sessionToken" : "F85E174AB7425AD66555E5CBD517CDF4",
    "ldapLookupOnOnlyForPerson" : true,
    "ldapLookupOn" : true,
    "Serviceform.label.alignment" : "right",
    "Serviceform.field.line.separator" : false
}
}

```

### 要求エントリを更新する API

**UpdateRequisitionAPI** は、要求に関連する **RequisitionEntry** を更新するために使用されます。この **RequisitionEntry** のフォームフィールドが更新されます。

- この API には、フィールドを含む複数のディクショナリがあります。フィールドのディクショナリ レベルで検証が行われます
- すべてのディクショナリ フィールド検証と送信後 DDR が実行されます。
- さまざまなデータ タイプのサンプル ペイロードを次の例に示します。

EndPoints : <http://10.78.0.247:8080/RequestCenter/nsapi/catalog/v1/requisitionentry/217/form>

メソッド : PUT

UpdateRequisitionAPI ペイロード :

```
{  
  
  "dictionaries" : [  
    {  
      "name" : "NagDictFF_NonGrid",  
      "data" : {  
        "Text" : "12",  
        "Number" : "12",  
        "Account" : "NewAccount",  
        "Boolean" : "No",  
        "Person" : "dusersd",  
        "Tenant": "TeamA",  
        "CheckBox" : ["Checkbox3", "Checkbox4", "Checkbox5"],  
        "Radio" : "Radio3",  
        "Date" : "10/29/2018",  
        "Money" : "12",  
  
        "Slider" : "30",  
        "SelectSingle" : "SS3",  
        "SelectMultiple" : ["SM1", "SM4"],  
        "DateAndTime" : "10/29/2018 10:42 PM",  
        "Phone" : "23634408",  
        "SSN" : "1234",  
        "URL" : "http://google.co.in",  
        "TextArea" : "This is the textarea",  
        "Password" : "hello"  
      }  
    }  
  ],  
  
  {  
    "name" : "NagDictFF_Grid",  
    "data" : {  
      "Text" : "Din1",  
      "Number" : "123",  
  
      "Account": "NewAccount",
```

```
        "Boolean": "No",
        "Tenant": "TeamZZ",
        "Person" : "din01",
        "Date": "10/25/2018",
        "Money": "12",
        "DateAndTime": "10/24/2018 5:20 PM",
        "Phone": "23634408",
        "SSN": "1234",
        "URL": "http://google.co.in",
        "TextArea": "This is the textarea",
        "Password": "hello",
        "SelectSingle": "GSS1"
    },
    {
        "name" : "NagDictFF_Grid",
        "data" : {
            "Text" : "Din1",
            "Number" : "123",
            "Account": "NewAccount",
            "Boolean": "No",
            "Tenant": "TeamZZ",
            "Person" : "din01",
            "Date": "10/25/2018",
            "Money": "12",
            "DateAndTime": "10/24/2018 5:20 PM",
            "Phone": "23634408",
            "SSN": "1234",
            "URL": "http://google.co.in",
            "TextArea": "This is the textarea",
            "Password": "hello",
            "SelectSingle": "GSS1"
        }
    },
},
```

```
    {
      "name" : "NagDictPrsnBsd",
      "data" : {
        "Select_Person" : "din01"
      }
    },
  {
    "name" : "NagDictSIBD",
    "data" : {
      "Name" : "din01 din01",
      "RequisitionID" : "12",
      "Age": "12",
      "Address" : "dineshk2@cisco.com",
      "Money": "132",
      "SubmittedDate" : "10/16/2018",
      "Lkong": "12",
      "DateTime": "10/31/2018 12:00 AM",
      "DTime" : "10/31/2018 12:00 AM"
    }
  }
]
}
```

#### サービス項目請求の概算コスト

この API は、指定のサービス フォーム データのすべてのサービス項目の見積価格を取得するために使用されます。

```
REST URL : http://localhost:8080/RequestCenter/nsapi/catalog/v1/service/152/price
```

メソッド : PUT

サンプル要求ペイロード :

```
{
  "dictionaries" : [{
    "name" : "AOC_SIBD_ComputePrice",
    "data" : {
      "Name" : "Jill",
      "Field_Text" : "DEF",
      "Field_Number" : "200",
      "AccountID" : "14",
      "CustomerID" : "1",
      "OrganizationalUnitID" : "1"
    }
  },
  {
    "name" : "AOC_SIBD_ComputePrice2",
    "data" : {
      "Name" : "Jack",
      "Field_Text2" : "ABC",
      "Field_Number2" : "100",
      "AccountID" : "7",
      "CustomerID" : "1",
      "OrganizationalUnitID" : "1"
    }
  },
  {
    "name" : "AOC_SIBD_ComputePriceUnit",
    "data" : {
      "Name" : "Eve",
      "Field_Number_UnitRate" : "200",
      "AccountID" : "7",
      "CustomerID" : "1",
      "OrganizationalUnitID" : "1"
    }
  }
}
```



```

    }
  },
  {
    "name" : "AOC_SIBD_ComputePriceUnit2",
    "data" : {
      "Name" : "David",
      "Field_Number_UnitRate2" : "300",
      "AccountID" : "7",
      "CustomerID" : "1",
      "OrganizationalUnitID" : "1"
    }
  }
]
}

```

応答ペイロード :

```

{
  "totalPrice" : "26400.0 Sp_Storage GB per Billing Cycle; 12200.0 Sp_Transactions; 123.0 Sp_Years",
  "list" : [
    {
      "dictionaryName" : "AOC_SIBD_ComputePriceUnit2",
      "priceList" : [
        "26400.0 Sp_Storage GB per Billing Cycle"
      ],
      "totalPrice" : ""
    },
    {
      "dictionaryName" : "AOC_SIBD_ComputePrice1",
      "priceList" : [],
      "totalPrice" : ""
    },
    {
      "dictionaryName" : "AOC_SIBD_ComputePriceUnit1",
      "priceList" : [
        "12200.0 Sp_Transactions"
      ]
    }
  ]
}

```

```

    ],
    "totalPrice" : ""
  },
  {
    "dictionaryName" : "AOC_SIBD_ComputePrice2",
    "priceList" : [
      "123.0 Sp_Years"
    ],
    "totalPrice" : ""
  }
]
}

```

## 12.1\_Patch\_v4 で導入

### オーダー フォームをドラフトとして保存する

Service Catalog オーダー フォームの新しいボタン **[ドラフトとして保存 ( Save as Draft in a Service Catalog order form ) ]** を使用すると、不完全なフォームを保存して後で編集することができます。これらの要求のステータスは、**[注文 (Orders) ]** ページに **[ドラフト (Draft) ]** として示されます。これらの注文は、注文フォームの完了後にキャンセル、カートに移動、または送信できます。**[ドラフトとして保存 (Save as Draft) ]** オプションは、サービス フォームとカートでも利用できます。

注：

1. カートへの移動操作は、カートが空の場合にのみ実行できます。
2. ドラフト要求は、ユーザごとに保存できます。
3. **[チーム管理 (Team Management) ]** が有効で、サービスが**[チーム関連 (Team Relevant) ]** の場合、サービス フォームのチーム名の検証は、**[ドラフトとして保存 (save as draft) ]** ではスキップされません (常に値が必要です)。

### 確認に戻る

承認タスクを割り当てられたユーザは、Service Catalog モジュールの **[マイ スタッフ (My Stuff) ] > [通知 (Notifications) ]** タブの未承認認可にある **[確認に戻る ( Return to Review ) ]** オプションを使用して、要求を送信者に送り返すことができます。承認タスクは、サービス グループ、部門、または財務の認可のいずれかになります。

確認のために注文が返されると、承認者は注文を返すためのコメントを追加する必要があります。このメモは、注文に必要な変更を加えて再送信する場合に役立ちます。確認のためにサービスを再調整すると、後続のサービス提供プロセスも確認状態に戻ります。要求が再送信されると、すべての認可が再トリガーされます。

**[確認に戻る (Return to Review) ]** オプションは、Prime Service Catalog の次の場所でも使用できます。

- **[Service Catalog] > [MyStuff] > [承認 (Approvals) ] > [認証 (Authorizations) ]**
- **[サービス マネージャ (Service Manager) ] > [承認タスクのビュー (Views for Approval Tasks) ]**
- **[サービス リンク (Service Link) ] > [トランザクションの表示 (View Transactions) ] > [外部タスク (External Tasks) ]**

確認に戻るための API については、アーカイブまたはページする [「アーカイブまたは消去する要求を取得する](#)

メソッド : **GET**

REST URL : `/RequestCenter/nsapi/transaction/v2/requisition?hasWorkFlowData=true`

サンプル応答 :

```
{
```

```

"requisitions" : {
  "startRow" : 1,
  "totalCount" : 1,
  "recordSize" : 1,
  "requisition" : [
    {
      "tenantId" : 0,
      "userId" : 0,
      "ownerId" : 0,
      "serviceId" : 0,
      "customerId" : 10645,
      "expectedDuration" : 0.0,
      "actualDuration" : 0.0,
      "expectedCost" : 0.0,
      "status" : "Closed",
      "requisitionId" : 535192,
      "lateFlag" : false,
      "statusId" : 2,
      "organizationalUnitId" : 0,
      "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=535192&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">535192</a>",
      "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=535192",
      "milestoneLink" : "",
      "percentageCompleted" : 0.0,
      "isCancelable" : false,
      "reviewsCount" : 0,
      "selfRating" : 0,
      "isServiceOrderable" : 0,
      "isServiceActive" : 0,
      "hasAttachment" : false,
      "teamId" : 0,
      "createdDate" : "12/09/2010 7:39 下午"
    }
  ]
}
}

```

要求をアーカイブまたは消去する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/requisitions/workflowdata?requisitioncount={recordSize}

ここで、recordSize は、[管理 (Administrations) ] > [設定 (Settings) ] の [ワークフロー データの消去/アーカイブ ( Purge / Archive Workflow Data ) ] 設定に基づいてアーカイブまたは消去する必要がある要求の数です。

サンプル応答 :

```
{
  "nsapi-error-response" : {
    "status-messages" : [
      {
        "code" : "NSAPI_ERROR_004",
        "value" : " Requisition Purged successfully"
      }
    ]
  }
}
```

確認のために戻る。

### 代理オーダーの自動サジェスト検索

[代理注文 (Order on Behalf Person) ] ポップアップの名前検索が、名前、姓、またはログイン名で検索操作を実行できる単一名検索に拡張されました。

また、代理注文ユーザ ポップアップ検索に自動提案機能を有効にできます。検索ボックスに入力すると、入力した文字に基づいてクエリが予測されます。検索自動提案機能を有効にするには、[管理 ( Administration ) ] モジュール > [設定 ( Settings ) ] に移動し、[OOB 自動提案の有効化 ( Enable OOB Auto Offer ) ] オプションを有効にします。このオプションは、デフォルトで無効です。

注：他のユーザのサービスを注文するには、ログインしたユーザに代理注文権限が必要です。

### 検索ファセットの表示順序の設定

[サービス カタログ (Service Catalog) ] モジュールの左側のペインで使用可能な検索ファセットは、選択した条件に基づいてサービスをフィルタリングまたは検索するために使用されます。サイト管理者として、[サービス デザイナー ( Service Designer ) ] > [拡張機能 ( Extensions ) ] > [ファセットの検索 ( Search Facets ) ] リスト パネルからファセットの優先順位を調整して、検索ファセットの優先順位を設定できるようになりました。上/下矢印を使用して、[Service Catalog] ページに表示するファセットの順序を指定します。

Search Facets																			
Name	Search Facets																		
<table border="1"> <thead> <tr> <th>Attribute Display Name</th> <th>up/down</th> </tr> </thead> <tbody> <tr> <td>Ratings</td> <td>↑ ↓</td> </tr> <tr> <td>City</td> <td>↑ ↓</td> </tr> <tr> <td>Color</td> <td>↑ ↓</td> </tr> <tr> <td>Apps</td> <td>↑ ↓</td> </tr> <tr> <td>A12888</td> <td>↑ ↓</td> </tr> <tr> <td>advance</td> <td>↑ ↓</td> </tr> <tr> <td>Brand Name</td> <td>↑ ↓</td> </tr> <tr> <td>76 Hardware Type</td> <td>↑ ↓</td> </tr> </tbody> </table>		Attribute Display Name	up/down	Ratings	↑ ↓	City	↑ ↓	Color	↑ ↓	Apps	↑ ↓	A12888	↑ ↓	advance	↑ ↓	Brand Name	↑ ↓	76 Hardware Type	↑ ↓
Attribute Display Name	up/down																		
Ratings	↑ ↓																		
City	↑ ↓																		
Color	↑ ↓																		
Apps	↑ ↓																		
A12888	↑ ↓																		
advance	↑ ↓																		
Brand Name	↑ ↓																		
76 Hardware Type	↑ ↓																		

ファセットの表示順序を設定するための API については、[優先的に検索ファセットを取得する](#) および [検索ファセットの優先度](#) で説明します。

### サービス カタログのカテゴリ内でのサービスのソート

管理者は、[最も注文された (Most Ordered)] または [名前 (Name)] (昇順/降順) に基づいて、カテゴリまたはサブカテゴリ内のサービスのソートを設定できます。デフォルトでは、サービス デザイナーで設定されたソートに従って、サービスが Service Catalog に表示されます。

[サービス ソートパターン (Services Sorting Pattern)] 設定オプションを使用して、[管理 (Administration)] モジュール > [設定 (Settings)] で提供されるソート パターンを変更します。

システムは、指定された期間からの注文量に基づいて最も注文されたサービスを表示します。たとえば、月を 1 月、年を 2012 年として選択した場合、選択したカテゴリに対して、2012 年 1 月から日付までに最も多く注文されたサービスが最初に表示されず。

### 承認の認可委任割り当て

[プロファイル (Profile)] > [設定 (Preferences)] で使用可能だった認証委任機能が、ハンバーガー メニューの [サービス カタログ (Service Catalog)] > [承認 (Approvals)] で使用できるようになりました。新しいオプション代理承認者では、[Delegation Start Date] と [Delegation End Date] フィールドを使用して指定する期間中、指定された人が代わりに承認を実行できます。詳細については、『[Cisco Prime Service Catalog 管理および操作ガイド](#)』の「ユーザへの権限の付与」の項を参照してください。

### サービス フォーム詳細 API のキー フィールド設定

サービス デザイナーは、承認に必要なキー フィールドのみが表示されるように、サービス データの取得 API で承認の詳細を設定できるようになりました。

サービスのディクショナリ レベルまたはアクティブ フォーム コンポーネント レベルでフィールドを設定できます。

- ディクショナリ レベル :  
各ディクショナリ フィールドについては、[サービス デザイナー (Service Designer)] > [ディクショナリ (Dictionaries)] > [is Key にチェックを入れる/チェックを外す (check/uncheck is Key)] フィールドに移動します。[is Key] とマークされたディクショナリ フィールドのみがサービス フォーム API に表示されます。
- アクティブ フォーム コンポーネント レベル :

[サービス デザイナー ( Service Designer ) ] > [アクティブ フォーム コンポーネント ( Active form Components ) ] > [プロパティの表示 ( Display properties ) ] の順に選択して、[overridesKey] をオンまたはオフにします。このオプションをオンにすると、デフォルトの各ディクショナリ フィールドのキー設定が上書きされます。

サービス デザイナーの設定に基づいて、以下の API の応答には必要なフィールドのみが表示されます。

1. 指定された要求のキー フィールドを取得します。  
http://<serverURL>/RequestCenter/nsapi/transaction/requisitions/id/{requisitionID}?requisitiondata=true&fetchKeyFieldsOnly=true
2. requisitiondata を使用して、指定された要求のキー フィールドを取得します。  
http://<serverURL>/RequestCenter/nsapi/transaction/requisitions/id/{requisitionID}/requisitiondata?fetchKeyFieldsOnly=true

クエリ パラメータ :

- requisitiondata = true / false : これは、API が要求内にある要求エントリのサービス フォーム フィールドを取得する必要があるかどうかを指定します
- fetchKeyFieldsOnly = true/false - この値は、この要求の keyFields のみを取得するかどうかを指定し、requisitiondata = true の場合に使用する必要があります

応答例については、[指定された要求のキー フィールド](#) および [要求データを使用して、指定された要求のキー フィールドを取得する](#) を参照してください。

### 要求のアーカイブと消去

サイト管理者として、[管理 ( Administrations ) ] > [設定 ( Settings ) ] の [ワークフロー データの消去/アーカイブ ( Purge / Archive Workflow Data ) ] 設定を使用して、完了した履歴要求をアーカイブまたは消去できるようになりました。次に説明する関連 API を使用して、システム内の最も古い要求から消去またはアーカイブする要求の数を設定します。

[完了 ( Completed ) ]、[キャンセル済み ( Cancelled ) ]、[拒否 ( Rejected ) ]、および [配信キャンセル済み ( Delivery Canceled ) ] 状態の要求のみがアーカイブまたは消去されます。

次の API を使用して、まだアーカイブ/消去されていない要求を取得し、アーカイブまたは消去するエントリ数を設定します。

1. まだアーカイブ/消去されていない要求を取得する API を取得します。  
/RequestCenter/nsapi/transaction/v2/requisition?hasWorkFlowData=true
2. [管理設定 ( Administration Settings ) ] に基づいて API を取得し、アーカイブ/消去します。  
/RequestCenter/nsapi/transaction/v2/tasks/requisitions/workflowdata?requisitioncount={recordSize}

2番目の API の RecordSize は、次のように *newscale.properties* ファイルの新しいフラグを使用して設定できます。

```
max.allowed.requisition.purge.count=100000
```

アーカイブまたは消去できる最大レコード数は 100000 です。

[ワークフロー データの消去/アーカイブ ( Purge / Archive Workflow Data ) ] オプション ボタンが [アーカイブ ( Archive ) ] に設定されている場合、API に記載されている要求の数がアーカイブされます。ラジオ ボタンが [消去 ( Purge ) ] に設定されている場合、API に記載されている要求の数が消去されます。

この機能の API の詳細については、セクション [アーカイブまたは消去する要求を取得](#) と [Error! Reference source not found.](#) アーカイブまたは消去する要求を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/requisition?hasWorkFlowData=true

サンプル応答 :

```
{
  "requisitions" : {
    "startRow" : 1,
    "totalCount" : 1,
```

```

"recordSize" : 1,
"requisition" : [
  {
    "tenantId" : 0,
    "userId" : 0,
    "ownerId" : 0,
    "serviceId" : 0,
    "customerId" : 10645,
    "expectedDuration" : 0.0,
    "actualDuration" : 0.0,
    "expectedCost" : 0.0,
    "status" : "Closed",
    "requisitionId" : 535192,
    "lateFlag" : false,
    "statusId" : 2,
    "organizationalUnitId" : 0,
    "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=535192&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">535192</a>",
    "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=535192",
    "milestoneLink" : "",
    "percentageCompleted" : 0.0,
    "isCancelable" : false,
    "reviewsCount" : 0,
    "selfRating" : 0,
    "isServiceOrderable" : 0,
    "isServiceActive" : 0,
    "hasAttachment" : false,
    "teamId" : 0,
    "createdDate" : "12/09/2010 7:39 下午"
  }
]
}
}

```

要求をアーカイブまたは消去する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/requisitions/workflowdata?requisitioncount={recordSize}

ここで、recordSize は、[管理 (Administrations) ] > [設定 (Settings) ] の [ワークフロー データの消去/アーカイブ ( Purge / Archive Workflow Data ) ] 設定に基づいてアーカイブまたは消去する必要がある要求の数です。

サンプル応答 :

```
{
  "nsapi-error-response" : {
    "status-messages" : [
      {
        "code" : "NSAPI_ERROR_004",
        "value" : " Requisition Purged successfully"
      }
    ]
  }
}
```

確認のために戻る。

アーカイブおよび消去される内部 DB ワークフロー テーブルは次のとおりです。

テーブルのパージ	アーカイブ テーブル
TxJoin	ArJoin
TxJoinExt	ArJoinExt
TxSubscription	ArSubscription
TxTimer	ArTimer
TxCondition	ArCondition
TxRule	ArRule
TxService	ArService

消去中は、消去テーブルのデータが削除されます。ただし、アーカイブ中は、消去テーブルのデータがそれぞれのアーカイブ テーブルに移動します。

### Service Manager の Web サイト

サービス マネージャ モジュールの新しい UI がこのパッチ リリースで導入され、Web サイト モデルを使用してカスタマイズできるようになりました。12.1 パッチ V4 にアップグレードすると、新しい Service Manager UI がデフォルトで有効になります。従来のサービス マネージャの UI によって提供されるすべての機能は、新しいサービス マネージャの UI でも同じであることに注意してください。サービス マネージャ従来の UI を使用する場合は、[管理 ( Administration ) ] > [設定 ( Settings ) ] で [サービス マネージャの Web サイト] を有効にする (Enable Service Manager Website) ] オプションを無効にします。

サービス マネージャの外観は、Service Catalog、テナント管理、クラウド統合、ユーザ管理などの他のモジュールと同様の Web サイト モデルを使用してカスタマイズできます。[管理 ( Administration ) ] モジュール > [設定 ( Settings ) ] タブ > [カスタム テーマ ( Custom Themes ) ] > [カスタム テーマのプロパティ ( Custom Theme Properties ) ] から、サービス マネージャの Web サイトのチェックボックスを有効にして、サービス マネージャのカスタム テーマを定義します。詳細については、「Cisco Prime Service Catalog 12.1 管理および操作ガイド」の「顧客が使用するモジュールのカスタマイズ」を参照してください。



このパッチ リリースでは、Service Manager に新しい API が導入されています。 [新しいサービス マネージャ API](#)の項を参照)

### CloudCenter 展開の電子メール通知

Prime Service Catalog で CloudCenter アプリケーションを作成または移行すると、電子メール通知を受信します。新しい CloudCenter アプリケーションの展開が正常に完了すると、サービス項目の詳細、サブスクリプションデータ、およびアプリケーションの個別の詳細（名前、表示名、説明、クラウド名、およびアプリケーションとそれに対応する仮想マシンのステータスなど）を含む電子メール通知が送信されます。

この電子メール通知の設定は、「Cisco Prime Service Catalog 12.1 管理および操作ガイド」の「VDC 作成時に電子メール通知を設定する」のセクションで説明されている手順と同じです。

### 展開および VM でのカスタム アクションのサポート

12.1 V3 パッチで提供される VM 機能のカスタム ライフサイクル操作のサポートが強化され、セクション [VM でのカスタム ライフサイクル操作のサポート](#)の手順が更新されました。また、カスタム アクションも展開でサポートされます。CloudCenter 展開のカスタム アクションを設定する詳細な手順については、[展開でのカスタム アクション操作のサポート](#)を参照してください。

### 拡張 API

必要なデータのみを返す認証 API

既存の取得認証 API は、新しく導入されたクエリ パラメータを使用して必要な情報のみを表示するように拡張されました。

メソッド：GET

REST URL : `/RequestCenter/nsapi/transaction/authorizations/{filters}?requiredFields= dueOn,lateFlag,isOOB,`  
 QueryParam : `requiredFields = <dueOn,lateFlag,isOOB>`

サンプル応答：

```
{
  "authorizations" : {
    "startRow" : 1,
    "totalCount" : 2,
    "recordSize" : 2,
    "authorizationtask" : [
      {
        "dueOn" : "06/22/2018 4:00 PM",
        "lateFlag" : false,
        "isOOB" : false
      },
      {
        "dueOn" : "06/22/2018 4:00 PM",
        "lateFlag" : false,
        "isOOB" : false
      }
    ]
  }
}
```

進行中の認証の要求ヘッダー

進行中のトランザクションの既存承認 API が拡張され、*includeHeader* クエリ パラメータが追加されました。この API にアクセスできるのは、タスクを承認する権限を持つユーザのみです。

メソッド : GET

REST URL :

/RequestCenter/nsapi/transaction/authorizations/ /RequestCenter/nsapi/transaction/authorizations/{filters}?includeHeader=true

クエリ パラメータ : *includeHeader* = True / false が追加され、応答で次の属性が返されます。

現在のタスク ID、ステータス、実行者 ID、実行者の名、姓、要求期限、タスクの期限、所有者 ID、所有者名、顧客 ID、顧客のログイン名、代理注文かどうか、今日からの差異（支払期日、経過日数）

サンプル応答 :

```
{
  "authorizations" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 1,
    "authorizationtask" : [
      {
        "tenantId" : 1,
        "dueOn" : "07/17/2018 8:00 PM",
        "totalPrice" : "$ 0.00",
        "status" : "Being approved",
        "requisitionId" : 2281,
        "taskName" : "AuthQ",
        "customer" : "admin admin : Site Administration",
        "performer" : "PSCQ Queue",
        "lateFlag" : false,
        "priority" : "Normal",
        "priorityIcon" : "/RequestCenter/images/prioritynormalicon.gif",
        "dueOnRaw" : 1531882800000,
        "activityId" : 4143,
        "authorizationURL" : "<a
href=/RequestCenter/servicemanager/navigate.do?query=task&taskID=4143&layout=popup_p&selectedTasks=& onclick=\"return
GB_showFullScreen('Task', this.href)\">AuthQ</a>",
        "serviceName" : "Personallaptop_service",
        "authorizationURLOnly" :
"/RequestCenter/servicemanager/navigate.do?query=task&taskID=4143",
        "scheduledStartRaw" : 1531782000000,
        "startedOnRaw" : 1531769664493,
```

```

        "flagId" : 0,
        "isPerformer" : true,
        "statusId" : "6",
        "priorityTypeId" : "2",
        "createdOn" : "07/16/2018 11:23 AM",
        "createdOnRaw" : 1531765431757,
        "customerLoginName" : "admin",
        "customerId" : 1,
        "ownerName" : "admin admin",
        "isOOB" : false,
        "ownerId" : 1,
        "performerId" : 1060,
        "performerFirstName" : "PSCQ",
        "performerLastName" : "Queue",
        "requisitionDueOn" : "07/17/2018 8:00 PM",
        "requisitionDueOnRaw" : 1531882800000,
        "daysToDueDate" : 0,
        "requisitionEntryId" : 2416
    }
}
}
}

```

#### Closed Authorization API の追加のカスタム データ

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/authorizations/ViewName=<viewName>[|AND|Status=<status>]

完了した認証のステータスは、承認済み、拒否、キャンセル、および確認済みです。

クエリ パラメータ :

includeHeader = true に設定すると、次の新しいフィールドが応答に表示されます。

createdOn、createdOnRaw、customerLoginName、customerId、ownerName、isOOB、ownerId、performerId、PerformerFirstName、PerformerLastName、requisitionDueOn、requisitionDueOnRaw、daysToDueDate

応答 :

```

{
  "authorizations" : {
    "startRow" : 1,
    "totalCount" : 2,
    "recordSize" : 2,

```

```
"authorizationtask" : [  
  {  
    "tenantId" : 1,  
    "dueOn" : "06/22/2018 4:00 PM",  
    "totalPrice" : "$ 0.00",  
    "status" : "Being approved",  
    "requisitionId" : 671170,  
    "taskName" : "UserAuth1",  
    "customer" : "admin admin : Site Administration",  
    "performer" : "shashi1 shashi1",  
    "lateFlag" : false,  
    "priority" : "Normal",  
    "priorityIcon" : "/RequestCenter/images/...",  
    "dueOnRaw" : 1529663400000,  
    "activityId" : 2125752,  
    "authorizationURL" : "<a href=/RequestCenter/servicemanager/.../UserAuth1</a>",  
    "serviceName" : "UserAuthorization",  
    "authorizationURLOnly" : "/RequestCenter/servicemanager/..." ,  
    "scheduledStartRaw" : 1529663400000,  
    "startedOnRaw" : 1529628842143,  
    "flagId" : 0,  
    "isPerformer" : true,  
    "statusId" : "6",  
    "priorityTypeId" : "2",  
    "createdOn" : "06/22/2018 6:23 AM",  
    "createdOnRaw" : 1529628825920,  
    "customerLoginName" : "admin",  
    "customerId" : 1,  
    "ownerName" : "admin admin",  
    "isOOB" : false,  
    "requisitionEntryId" : 662941  
  },  
  {  
    "tenantId" : 1,  
    "dueOn" : "06/22/2018 4:00 PM",  
    "totalPrice" : "$ 0.00",  
    "status" : "Being approved",
```

```

        "requisitionId" : 671170,
        "taskName" : "UserAuth1",
        "customer" : "admin admin : Site Administration",
        "performer" : "shashi1 shashi1",
        "lateFlag" : false,
        "priority" : "Normal",
        "priorityIcon" : "/RequestCenter/images/...",
        "dueOnRaw" : 1529663400000,
        "activityId" : 2125753,
        "authorizationURL" : "<a href='/RequestCenter/servicemanager/.../UserrAuth1</a>",
        "serviceName" : "UserAuthorization",
        "authorizationURLOnly" : "/RequestCenter/servicemanager...",
        "scheduledStartRaw" : 1529663400000,
        "startedOnRaw" : 1529628842737,
        "flagId" : 0,
        "isPerformer" : true,
        "statusId" : "6",
        "priorityTypeId" : "2",
        "createdOn" : "06/22/2018 6:23 AM",
        "createdOnRaw" : 1529628827027,
        "customerLoginName" : "admin",
        "customerId" : 1,
        "ownerName" : "admin admin",
        "isOOB" : false,
        "requisitionEntryId" : 662942
    }
}
]
}
}

```

オープン認証数を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/authorizations/count/AuthViewName=[My Authorizations]&AND|AuthStatus=[Ongoing]

応答 :

```

{
    "authorizations" : {

```

```

    "authorizationsCount" : 2
  }
}

```

#### ID によるコンシューマ サービス カテゴリの取得

指定した Id のサービス カタログ カテゴリが返されます。ネストされたエンティティ（サブカテゴリおよび組み込みサービス）が取得されます。既存のカテゴリ API に代わる新しい API が導入され、応答のパフォーマンスが向上しました。

既存の GET API は次のとおりです。

```
/RequestCenter/nsapi/definition/categories/id/{categoryID}?responseType=json&recursive=true&isNgcRequest=true
```

メソッド : GET

REST URL : RequestCenter/nsapi/definition/v2/categories/<Category\_ID>? hierarchyLevels=<No of Category Levels>&Services=<true/false>

それぞれの説明は次のとおりです。

- **HierarchyLevels : number** : ルートとしてパスされた **categoryId** を使用して、階層のレベルまでのカテゴリを返します。このパラメータが渡されない場合、カテゴリ ID のすべてのカテゴリが返されます。
- **Services : true / false** : **true** に設定すると、API は返されたカテゴリに属するサービスの **serviceID** を返します。**false** に設定すると、**serviceID** は取得されません。

Java の例 :

```
/RequestCenter/nsapi/definition/v2/categories/137?hierarchyLevels=5&services=true
```

サンプル応答 :

```

{
  "bottomDescription" : "",
  "bottomDescriptionEnabled" : false,
  "bottomDescriptionURL" : "",
  "catalogType" : "Consumer Services Catalog",
  "catalogTypeId" : 1,
  "categoryExtensions" : [
    {
      "facetPriorities" : "",
      "facetValues" : [],
      "logicName" : "CategoryID",
      "name" : "CategoryID",
      "type" : "",
      "value" : "47"
    }
  ],
}

```

```
{
    "facetPriorities" : "",
    "facetValues" : [],
    "logicName" : "imageUri",
    "name" : "ImageURL",
    "type" : "",
    "value" : "something"
}
],
"categoryId" : 47,
"categoryName" : "C1",
"description" : "",
"imageUrl" : "",
"isRoot" : false,
"maxQuantity" : 0,
"middleDescription" : "",
"middleDescriptionEnabled" : false,
"middleDescriptionURL" : "",
"serviceIds" : [2,3,4],
"showCaseCategoryLocation" : "",
"subCategories" : [
    {
        "id" : 48,
        "logicName" : "c2",
        "name" : "C2",
        "parentId" : 47,
        "serviceCount" : 1
    },
    {
        "id" : 49,
        "logicName" : "c3",
        "name" : "C3",
        "parentId" : 47,
        "serviceCount" : 1
    }
],
"topDescription" : "",
"topDescriptionEnabled" : false,
```

```

    "topDescriptionURL" : ""
  }

```

未処理の注文の追加情報を取得する

メソッド : GET

REST URL:

/RequestCenter/nsapi/transaction/requisitionentries/<ViewName>?taskDetails=true&currentActivities=true& serviceExtensions=true  
クエリ パラメータ :

- **taskDetails = true/false** - これは、API が要求エントリのタスク詳細を取得するかどうかを指定します
- **currentActivities = true/false** - true に設定すると、要求エントリの現在のタスクのみが返されます
- **serviceExtensions = true / false** : 要求エントリのサービス拡張を取得するかどうかを指定します

新しいフィールドが追加されました。

注文の現在進行中のタスク、現在のタスク ID、実行者 ID、実行者の名、姓、要求の期限、タスクの期限、所有者 ID、所有者名、顧客 ID、顧客ログイン名、代理注文の場合、送信日、期限から経過期間（残り日数、延滞日、その要求のサービス拡張）

注：グレーで強調表示されたフィールドは、以下の応答に新たに追加されます。

サンプル応答 :

```

{
  "requisitionEntries" : {
    "startRow" : 1,
    "totalCount" : 2,
    "recordSize" : 2,
    "requisitionEntry" : [
      {
        "requisitionEntryId" : 2430,
        "requisitionId" : 2293,
        "serviceId" : 1429,
        "serviceDescription" : "",
        "serviceName" : "X1 Carbon_Laptop",
        "standardDuration" : "Not Defined",
        "unitCost" : "$ 0.00",
        "subTotal" : "$ 0.00",
        "status" : "Ongoing",
        "dueOnDate" : "07/18/2018",
        "dueOnDateRaw" : 1531960200000,
        "quantity" : 1,
        "maxQuantity" : 0,

```



```

    "serviceLevelDescription" : "",
    "percentageCompleted" : 0.0,
    "statusId" : 1,
    "expectedDuration" : 0.0,
    "reqEntryServiceURL" : "<a href=\"#\" onclick=\"GB_showFullScreen('X1
Carbon_Laptop','/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&layout=popup_p&reqid=2293&reqentryid=2
430&formAction=displayEntryStatus&performerID=&serviceid=1429&requisitionId=2293')\">X1 Carbon_Laptop</a>",
    "reqEntryServiceURLOnly" :
"/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&reqid=2293&reqentryid=2430&formAction=displayEntryStat
us&performerID=&serviceid=1429&requisitionId=2293",
    "requisitionStatusId" : 1,
    "requisitionStatus" : "Ongoing",
    "startedDate" : "07/16/2018",
    "startedDateRaw" : 1531832982207,
    "submitDate" : "07/16/2018",
    "submitDateRaw" : 1531832985850,
    "customerName" : "Vinay Panchagavi",
    "customerId" : 1061,
    "customerLoginName" : "vspancha",
    "customerOUID" : 30,
    "customerEmail" : "vspancha@cisco.com",
    "reviewsCount" : 0,
    "selfRating" : 0,
    "isServiceOrderable" : 1,
    "isServiceActive" : 1,
    "isCancelable" : 1,
    "ownerName" : "Vinay Panchagavi",
    "ownerId" : 1061,
    "ownerLoginName" : "vspancha",
    "dueBy" : "2 day(s) overdue",
    "taskDetails" : [
        {
            "taskId" : 4198,
            "taskName" : "SGRSiteLevel",
            "performerId" : 26,
            "performerFirstName" : "dinesh07",
            "performerLastName" : "dinesh07",
            "taskState" : "Under review",
            "dueOn" : "07/17/2018",

```

```

        "dueOnRaw" : 1531875600000,
        "requisitionEntryId" : 2430,
        "requisitionId" : 2293,
        "dueBy" : "3 day(s) overdue",
        "office" : false
    },
    {
        "taskId" : 4199,
        "taskName" : "SGRSiteLevel_Queue",
        "performerId" : 1064,
        "performerFirstName" : "ITServices_Queue",
        "performerLastName" : "Queue",
        "taskState" : "Under review",
        "dueOn" : "07/17/2018",
        "dueOnRaw" : 1531868400000,
        "requisitionEntryId" : 2430,
        "requisitionId" : 2293,
        "dueBy" : "3 day(s) overdue",
        "office" : true
    }
],
    "orderOnBehalf" : false
}
]
}

```

クローズしたオーダーの追加情報を取得する

メソッド : GET

REST URL:

/RequestCenter/nsapi/transaction/requisitionentries/<ViewName>|AND|Status=AllCompleted? task-Details=true&currentActivities=true& serviceExtensions=true

クエリ パラメータ :

- **taskDetails = true/false** - これは、API が要求エントリのタスク詳細を取得するかどうかを指定します
- **currentActivities = true/false** - true に設定すると、要求エントリの現在のタスクのみが返されます
- **serviceExtensions = true/false** : 要求エントリのサービス拡張を取得するかどうかを指定します

新しいフィールドが追加されました。

注文の現在進行中のタスク、現在のタスク ID、実行者 ID、実行者の名、姓、要求の期限、タスクの期限、所有者 ID、所有者名、顧客 ID、顧客ログイン名、代理注文の場合、送信日、期限から経過期間（残り日数、延滞日、その要求のサービス拡張）

注：グレーで強調表示されたフィールドは、以下の応答に新たに追加されます。

サンプル応答：

```
{
  "requisitionEntries" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 1,
    "requisitionEntry" : [
      {
        "requisitionEntryId" : 41,
        "requisitionId" : 7,
        "serviceId" : 147,
        "serviceDescription" : "",
        "serviceName" : "order for others",
        "standardDuration" : "Not Defined",
        "unitCost" : "$ 0.00",
        "subTotal" : "$ 0.00",
        "status" : "Ongoing",
        "dueOnDate" : "07/02/2018",
        "dueOnDateRaw" : 1530527400000,
        "quantity" : 1,
        "maxQuantity" : 0,
        "serviceLevelDescription" : "",
        "percentageCompleted" : 0.0,
        "statusId" : 1,
        "expectedDuration" : 0.0,
        "reqEntryServiceURL" : "<a href=\"\#\" onclick=\"GB_showFullScreen('order for others','/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&layout=popup_p&reqid=7&reqentryid=41&formAction=displayEntryStatus&performerID=&serviceid=147&requisitionId=7')\">order for others</a>",
        "reqEntryServiceURLOnly" :
"/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&reqid=7&reqentryid=41&formAction=displayEntryStatus&performerID=&serviceid=147&requisitionId=7",
        "requisitionStatusId" : 1,
        "requisitionStatus" : "Ongoing",
        "startedDate" : "07/01/2018",
        "startedDateRaw" : 1530480343000,
```

```
"submitDate" : "07/01/2018",
"submitDateRaw" : 1530480542000,
"customerName" : "user2 user2",
"customerId" : 6,
"customerLoginName" : "user2",
"customerOUIId" : 21,
"customerEmail" : "user2@cisco.com",
"reviewsCount" : 0,
"selfRating" : 0,
"isServiceOrderable" : 1,
"isServiceActive" : 1,
"isCancelable" : 1,
"ownerName" : "user1 user1",
"ownerId" : 5,
"ownerLoginName" : "user1",
"daysToDueOnDate" : "1 day(s) overdue",
"taskDetails" : [
  {
    "taskId" : 5,
    "taskName" : "Monitor plan for order for others",
    "performerId" : 2,
    "performerFirstName" : "Default Service Delivery",
    "performerLastName" : "Queue",
    "taskState" : "Ongoing",
    "dueOn" : "07/02/2018",
    "dueOnRaw" : 1530527400000,
    "requisitionEntryId" : 41,
    "requisitionId" : 7,
    "daysToDueDate" : "1 day(s) overdue",
    "office" : true
  }
],
"orderOnBehalf" : true
}
]
}
```

**新しい API**

アーカイブまたは消去する要求を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/requisition?hasWorkFlowData=true

サンプル応答 :

```
{
  "requisitions" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 1,
    "requisition" : [
      {
        "tenantId" : 0,
        "userId" : 0,
        "ownerId" : 0,
        "serviceId" : 0,
        "customerId" : 10645,
        "expectedDuration" : 0.0,
        "actualDuration" : 0.0,
        "expectedCost" : 0.0,
        "status" : "Closed",
        "requisitionId" : 535192,
        "lateFlag" : false,
        "statusId" : 2,
        "organizationalUnitId" : 0,
        "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=535192&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">535192</a>",
        "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=535192",
        "milestoneLink" : "",
        "percentageCompleted" : 0.0,
        "isCancelable" : false,
        "reviewsCount" : 0,
        "selfRating" : 0,
        "isServiceOrderable" : 0,
        "isServiceActive" : 0,

```

```

        "hasAttachment" : false,
        "teamId" : 0,
        "createdDate" : "12/09/2010 7:39 下午"
      }
    ]
  }
}

```

要求をアーカイブまたは消去する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/requisitions/workflowdata?requisitioncount={recordSize}

ここで、recordSize は、[管理 (Administrations)] > [設定 (Settings)] の [ワークフロー データの消去/アーカイブ (Purge / Archive Workflow Data)] 設定に基づいてアーカイブまたは消去する必要がある要求の数です。

サンプル応答 :

```

{
  "nsapi-error-response" : {
    "status-messages" : [
      {
        "code" : "NSAPI_ERROR_004",
        "value" : " Requisition Purged successfully"
      }
    ]
  }
}

```

確認のために戻る

メソッド : POST

REST URL : /RequestCenter/nsapi/transaction/tasks/{taskID}/returnforreview

サンプル応答 :

<nsapi-response>タスクは正常に確認に戻りました。</nsapi-response>

指定された要求のキー フィールドを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/requisitions/id/{requisitionID}/requisitiondata?fetchKeyFieldsOnly=true

クエリ パラメータ

- requisitiondata = true / false : これは、API が要求内にある要求エントリのサービス フォーム フィールドを取得する必要があるかどうかを指定します
- fetchKeyFieldsOnly = true/false - この値は、この要求の keyFields のみを取得するかどうかを指定し、requisitiondata = true の場合に使用する必要があります

サンプル応答 :

```

{
  "RequisitionEntries" : [
    {
      "RequisitionEntry" : {
        "ServiceName" : "WebService1",
        "FormFields" : [
          {
            "DictionaryName" : "WebServiceDictionary1",
            "WebServiceDictioanry1.FirstName" : "admin",
            "mandatory" : false
          },
          {
            "WebServiceDictioanry1.LastName" : "kewl",
            "DictionaryName" : "WebServiceDictioanry1",
            "mandatory" : false
          }
        ],
        "RequisitonEntryID" : 663571,
        "Quantity" : 1
      }
    }
  ]
}

```

要求データを使用して、指定された要求のキー フィールドを取得する  
メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/requisitions/id/{requisitionID}?requisitiondata=true&fetchKeyFieldsOnly=true  
サンプル応答 :

```

{
  "requisition" : {
    "actualDuration" : 0,
    "attachmentImage" : "",
    "closedDate" : "",
    "closedDateRaw" : null,
    "createdDate" : "",
    "customerEmail" : "sanjaymalligere@cisco.com",
    "customerId" : 228075,
    "customerName" : "SanjayMalligere Shankar",

```

```
"customerWorkPhone" : "",
"dueDate" : "07/06/2018 1:00 PM",
"dueDateRaw" : {
    "date" : 6,
    "day" : 5,
    "hours" : 13,
    "minutes" : 0,
    "month" : 6,
    "seconds" : 0,
    "time" : 1530862200000,
    "timezoneOffset" : -330,
    "year" : 118
},
"expectedCost" : 0,
"expectedDuration" : 0,
"flagImage" : "",
"hasAttachment" : false,
"isCancelable" : true,
"isServiceActive" : 1,
"isServiceOrderable" : 0,
"lateFlag" : false,
"milestoneLink" : " ",
"organizationalUnitId" : 15272,
"organizationalUnitName" : "SanjayMalligere_OU",
"ownerId" : 228075,
"ownerName" : "SanjayMalligere Shankar",
"percentageCompleted" : 0,
"rating" : "0",
"requisitionData" : [
    {
        "RequisitionEntry" : {
            "ServiceName" : "SanjayWebService",
            "FormFields" : [
                {
                    "DictionaryName" : "SanjayWebServiceDictioanry",
                    "SanjayWebServiceDictioanry.FirstName" : "admin",
                    "mandatory" : false
                }
            ]
        }
    }
]
```



```
    }
  ],
  "RequisitionEntryID" : 663571,
  "Quantity" : 1
}
}
],
"requisitionId" : 671774,
"requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=671774&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">671774</a>",
"requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=671774",
"reviewsCount" : 0,
"selfRating" : 0,
"serviceId" : 2824,
"serviceImage" : "",
"serviceName" : "SanjayWebService",
"services" : [],
"startDate" : "07/06/2018 7:17 AM",
"startDateRaw" : {
  "date" : 6,
  "day" : 5,
  "hours" : 7,
  "minutes" : 17,
  "month" : 6,
  "seconds" : 26,
  "time" : 1530841646260,
  "timezoneOffset" : -330,
  "year" : 118
},
"status" : "Ongoing",
"statusId" : 1,
"submitDate" : "07/06/2018 7:18 AM",
"submitDateRaw" : {
  "date" : 6,
  "day" : 5,
  "hours" : 7,
  "minutes" : 18,
```

```

        "month" : 6,
        "seconds" : 34,
        "time" : 1530841714787,
        "timezoneOffset" : -330,
        "year" : 118
    },
    "teamDisplayName" : "",
    "teamId" : 0,
    "tenantId" : 1,
    "totalCost" : "0.0",
    "userId" : 228075
}
}

```

[Ordered For MySelf] および [Ordered-For-Other] のすべての要求エントリを取得する。

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/requisitionentries/ViewName=Orders with my involvement

パス パラメータ :

ViewName = 自分が関与する注文 : このビューは、ユーザが関与したすべての要求エントリを返します (例 : 自分/他のユーザに向けてユーザによって注文され、ユーザに向けて他のユーザによって注文された要求。)

サンプル応答 :

```

{
  "requisitions" : {
    "startRow" : 1,
    "totalCount" : 4,
    "recordSize" : 4,
    "lastUpdatedDateRaw" : 1529935200000,
    "lastUpdatedDate" : "06/25/2018 7:00 AM",
    "requisition" : [
      {
        "tenantId" : 1,
        "userId" : 31,
        "ownerId" : 31,
        "serviceId" : 183,
        "customerId" : 31,
        "expectedDuration" : 0.0,

```

```
"actualDuration" : 0.0,
"startedDate" : "06/21/2018 12:01 PM",
"dueDate" : "06/21/2018 4:00 PM",
"expectedCost" : 0.0,
"status" : "Ongoing",
"requisitionId" : 1279,
"flagImage" : "/RequestCenter/images/flaglate.gif",
"lateFlag" : true,
"customerName" : "OOBUser2 User2",
"organizationalUnitName" : "OOB_OU2",
"submitDate" : "06/21/2018 12:03 PM",
"statusId" : 1,
"serviceName" : "SanjayDraftService",
"ownerName" : "OOBUser2 User2",
"organizationalUnitId" : 45,
"startedDateRaw" : 1529607719487,
"dueDateRaw" : 1529622000000,
"submitDateRaw" : 1529607785807,
"requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=1279&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">1279</a>",
"requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=1279",
"milestoneLink" : " ",
"percentageCompleted" : 0.0,
"customerEmail" : "oobuser2@cisco.com",
"isCancelable" : true,
"rating" : "0",
"reviewsCount" : 0,
"selfRating" : 0,
"isServiceOrderable" : 0,
"isServiceActive" : 1,
"hasAttachment" : false,
"totalCost" : "0.0",
"teamId" : 0
]
}
}
```

優先的に検索ファセットを取得する

メソッド : GET

REST URL : RequestCenter/nsapi/definition/searchFacets

サンプル応答 :

```
{
  "SearchFacetList" : {
    "searchFacets" : [
      {
        "qualifierName" : "Search Facets",
        "attributeType" : 7,
        "facetValues" : [
          "OneAndAbove",
          "TwoAndAbove",
          "ThreeAndAbove",
          "FourAndAbove",
          "None"
        ],
        "name" : "FctAvgRating",
        "caption" : "Ratings",
        "dataTypeCompositionID" : 1220,
        "multiValue" : true,
        "priority" : 6
      },
      {
        "qualifierName" : "Search Facets",
        "attributeType" : 6,
        "facetValues" : [
          "Bangalore",
          "Sanjose",
          "Ottawo"
        ],
        "name" : "City",
        "caption" : "City",
        "dataTypeCompositionID" : 1329,
        "multiValue" : false,
        "priority" : 7
      }
    ]
  }
}
```

検索ファセットの優先度を更新する

メソッド : PUT

REST URL : /RequestCenter/nsapi/definition/searchFacets

要求本文 :

```
{
  "SearchFacetList" : {
    "searchFacets" : [
      {
        "name" : "FctAvgRating",
        "priority" : 2
      },
    ]
  }
}
```

```
{
  "name" : "test_facet_1",
  "priority" : 3
}
]
```

### 新しいサービス マネージャ API

タスクを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks

サンプル応答 :

```
{
  "TaskList" : {
    "startRow" : 1,
    "totalCount" : 13,
    "recordSize" : 13,
    "tasks" : [
      {
        "taskId" : 313,
        "taskName" : "DeptAuth",
        "taskNameSid" : "DeptAuth",
        "dueOnRaw" : 1532082600000,
        "scheduledStartRaw" : 1532082600000,
        "scheduledCompleteRaw" : 1532082600000,
        "completedOnRaw" : 1532056932000,
        "startedOnRaw" : 1532056520000,
        "dueOn" : "07/20/2018 4:00 PM",
        "requisitionId" : 51,
        "priority" : "prioritynormalicon.gif",
        "performerName" : "Sanjay Malligere",
        "performerId" : 24,
        "scheduledStart" : "07/20/2018 4:00 PM",
        "startedOn" : "07/20/2018 8:45 AM",
        "scheduledComplete" : "07/20/2018",
        "completedOn" : "07/20/2018 8:52 AM",
        "computeEarliestDate" : "07/20/2018 9:15 AM",
```

```
    "serviceId" : 0,
    "initiator" : "admin admin",
    "initiatorId" : 1,
    "customer" : "admin admin",
    "customerId" : 1,
    "customerOuld" : 1,
    "status" : "Approved",
    "effort" : 0.0,
    "taskType" : "Basic approval",
    "ouName" : "Site Administration",
    "queueId" : 0,
    "taskURL" : "<a
href=/RequestCenter/servicemanager/navigate.do?query=task&taskId=313&layout=popup_p&selectedTasks=&' onclick=\"return
GB_showFullScreen('Task', this.href)\">DeptAuth</a>",
    "taskURLOnly" : "/RequestCenter/servicemanager/navigate.do?query=task&taskId=313",
    "expectedDuration" : 0.0,
    "lateFlag" : false,
    "displayOrder" : 0,
    "parentTask" : 0,
    "stateId" : 7,
    "expanded" : true,
    "activityTypeId" : 2,
    "projectActivityID" : 0,
    "creatorObjectID" : 57,
    "treeLevel" : 0,
    "hasAttachment" : false,
    "requisitionEntryId" : 0,
    "flagId" : 0,
    "officeId" : 0,
    "priorityTypeId" : 2,
    "followUp" : "Red Flag",
    "isPerformer" : true,
    "priorityName" : "Normal"
  }
}
]
```

```

    }
}

```

objectType のタスクカウント API を取得します  
メソッド : GET

REST URL : /RequestCenter/nsapi/directory/organizationalunits?type=serviceTeam&viewSM=true

サンプル応答 :

```

{
  "OrganizationalUnitList" : {
    "startRow" : 6,
    "totalCount" : 77,
    "recordSize" : 20,
    "organizationalUnits" : [
      {
        "organizationalUnitName" : "BAT_OU",
        "organizationalUnitId" : 57,
        "parentId" : 0,
        "itemGroupId" : 0,
        "itemInstanceId" : 0,
        "organizationalUnitTypeId" : 1,
        "organizationalUnitType" : "Service Team",
        "statusId" : 1,
        "status" : "Active",
        "managerId" : 0,
        "isBillable" : "false",
        "accountId" : 0,
        "taskCount" : 2,
        "orgUnitURL" : "<a
href=/RequestCenter/organizationdesigner/scnavigate.do?displayRecordId=57&id=57&query=search&resetBN=true&formAction=display&displayRec=Y&forwardPage=organizationalunits&forwardTo=showSearchGeneralSuccess&mdicontentPortlet=portlet.sc.ou.general&mdicomponentsPortlet=portlet.sc.ou.mdi&selectMDI=sc.ou.general&isCreate=Y&sFId=Y&layout=popup_p' onclick=\"return GB_showFullScreen('Organizational Unit', this.href)\">BAT_OU</a>",
        "orgUnitURLOnly" :
"/RequestCenter/organizationdesigner/scnavigate.do?displayRecordId=57&id=57&query=search&resetBN=true&formAction=display&displayRec=Y&forwardPage=organizationalunits&forwardTo=showSearchGeneralSuccess&mdicontentPortlet=portlet.sc.ou.general&mdicomponentsPortlet=portlet.sc.ou.mdi&selectMDI=sc.ou.general&isCreate=Y&sFId=Y"
      }
    ]
  }
}

```

キューの詳細を取得する  
メソッド : GET

REST URL : /RequestCenter/nsapi/directory/v2/queue

サンプル応答 :

```
{
  "QueueList" : {
    "startRow" : 1,
    "totalCount" : 18,
    "recordSize" : 18,
    "queue" : [
      {
        "id" : 2,
        "name" : "Default Service Delivery",
        "homeOrganizationalUnitId" : 1,
        "notes" : "Queue for unassigned work. このキューは、サービス デザイナーが \\標準期間を使用して
概算\\ オプションを選択した場合、サービスの期日を計算するためにも使用されます。",
        "placeId" : 0,
        "recordStateId" : 1,
        "status" : 0,
        "tenantId" : 1,
        "timeZoneId" : 256,
        "timeZoneDisplayName" : "(GMT-08:00) Pacific Time (US and Canada), Tijuana",
        "taskCount" : 58,
        "shared" : false
      }
    ]
  }
}
```

カレンダーの詳細を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/calendar

サンプル応答 :

```
{
  "CalendarList" : {
    "startRow" : 1,
    "totalCount" : 13,
    "recordSize" : 1,
  }
}
```



```
"calendar" : [  
  {  
    "calendarEntryID" : 122,  
    "subject" : "Calendar Task",  
    "calendarDate" : 1517443200000,  
    "startTime" : 1517484600000,  
    "endTime" : 1517513400000,  
    "sequence" : 0,  
    "scheduledObjectID" : 2,  
    "scheduledObjectInstID" : 1,  
    "eventObjectID" : 23,  
    "eventObjectInstID" : 483,  
    "isRead" : 0,  
    "orgCalendarDate" : "02/01/2018 5:30",  
    "orgStartDate" : "02/01/2018 17:00",  
    "orgEndDate" : "02/02/2018 1:00",  
    "requisitionId" : 133,  
    "guid" : "586AAC4C-59B4-422A-BDF7-AB93AFA01C68",  
    "priority" : 2  
  },  
  {  
    "calendarEntryID" : 117,  
    "subject" : "Calendar Task",  
    "calendarDate" : 1517443200000,  
    "startTime" : 1517522400000,  
    "endTime" : 1517549250000,  
    "sequence" : 0,  
    "scheduledObjectID" : 2,  
    "scheduledObjectInstID" : 1,  
    "eventObjectID" : 23,  
    "eventObjectInstID" : 475,  
    "isRead" : 0,  
    "orgCalendarDate" : "02/01/2018 5:30",  
    "orgStartDate" : "02/02/2018 3:30",  
    "orgEndDate" : "02/02/2018 10:57",  
    "requisitionId" : 129,  
    "guid" : "32DC97C9-B8D1-434F-BDDF-F25F0D03CF6D",
```

```
    "priority" : 2
  },
  {
    "calendarEntryID" : 116,
    "subject" : "Calendar Task",
    "calendarDate" : 1517443200000,
    "startTime" : 1517455650000,
    "endTime" : 1517464800000,
    "sequence" : 0,
    "scheduledObjectID" : 2,
    "scheduledObjectInstID" : 1,
    "eventObjectID" : 23,
    "eventObjectInstID" : 475,
    "isRead" : 0,
    "orgCalendarDate" : "02/01/2018 5:30",
    "orgStartDate" : "02/01/2018 8:57",
    "orgEndDate" : "02/01/2018 11:30",
    "requisitionId" : 129,
    "guid" : "7DE22DA3-C426-498B-BA82-BB30096B036A",
    "priority" : 2
  },
  {
    "calendarEntryID" : 120,
    "subject" : "Calendar Task",
    "calendarDate" : 1517443200000,
    "startTime" : 1517484600000,
    "endTime" : 1517513400000,
    "sequence" : 0,
    "scheduledObjectID" : 2,
    "scheduledObjectInstID" : 1,
    "eventObjectID" : 23,
    "eventObjectInstID" : 479,
    "isRead" : 0,
    "orgCalendarDate" : "02/01/2018 5:30",
    "orgStartDate" : "02/01/2018 17:00",
    "orgEndDate" : "02/02/2018 1:00",
    "requisitionId" : 131,
```

```

        "guid" : "7E81885A-0394-4E88-A7FA-A33BF3A156F6",
        "priority" : 2
    },
    {
        "calendarEntryID" : 123,
        "subject" : "Calendar Task",
        "calendarDate" : 1517443200000,
        "startTime" : 1517460722000,
        "endTime" : 1517484600000,
        "sequence" : 0,
        "scheduledObjectID" : 2,
        "scheduledObjectInstID" : 1,
        "eventObjectID" : 23,
        "eventObjectInstID" : 485,
        "isRead" : 0,
        "orgCalendarDate" : "02/01/2018 5:30",
        "orgStartDate" : "02/01/2018 10:22",
        "orgEndDate" : "02/01/2018 17:00",
        "requisitionId" : 134,
        "guid" : "09365C3D-0647-40DA-AA3D-C1E8D8A8FB2C",
        "priority" : 2
    }
]
}
}

```

カレンダーの詳細をエクスポートする  
メソッド : GET

REST URL : RequestCenter/nsapi/transaction/v2/tasks/calendar/export/{calendarEntryId}

サンプル ペイロード :

```

BEGIN : VCALENDAR
PRODID : Microsoft Exchange for Microsoft Exchange
VERSION : 2.0
METHOD:PUBLISH
BEGIN : VEVENT
DTSTAMP : 20180201T053000

```

DTSTART : 20180202T033000

DTEND : 20180202T105730

UID : {32DC97C9-B8D1-434F-BDDF-F25F0D03CF6D}

Summary : Calendar Task

説明 : カレンダー タスクで作業を実行

SEQUENCE : 0

PRIORITY : 2

CLASS : Personal

STATUS : CONFIRMED

TRANSP : OPAQUE

X-MICROSOFT-CDO-BUSYSTATUS : BUSY

X-MICROSOFT-CDO-INSTTYPE : 0

END : VEVENT

END : VCALENDAR

複数のタスクの手順を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/instructions?taskIds= {taskIDs}

Java の例 :

http://<ServerURL>/RequestCenter/nsapi/transaction/v2/tasks/instructions?taskIds=203,204

応答

```
{
  "Map" : {
    "203" : {},
    "204" : {
      "instruction" : "test instructions",
      "instructionURL" : "http://cisco.com",
      "instructionURLDescription" : "test desc"
    }
  }
}
```

複数のタスクの情報を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/info?taskIds={taskIDs}

Java の例 :

http://localhost:8080/RequestCenter/nsapi/transaction/v2/tasks/info?taskIds=201,202

サンプル応答 :

```
{
  "Map" : {
    "201" : {
      "taskTypeId" : 1,
      "taskAgentId" : 0,
      "projectTaskId" : 199,
      "parentTaskId" : 0,
      "ticketObjectId" : 37,
      "creatorObjectId" : 46,
      "creatorObjectInstId" : 289,
      "showEffort" : true,
      "hasChecklist" : true,
      "hasInstructions" : true,
      "initTab" : "Tab2",
      "availableActions" : [
        {
          "actionID" : 2,
          "actionName" : "Done",
          "isDefault" : true
        },
        {
          "actionID" : 8,
          "actionName" : "Cancel",
          "isDefault" : false
        },
        {
          "actionID" : -4,
          "actionName" : "Assign",
          "isDefault" : false
        }
      ],
      "readOnlyReq" : false,
      "priceTask" : false
    },
    "202" : {
      "taskTypeId" : 4,
      "taskAgentId" : 0,
      "projectTaskId" : 0,

```

```
"parentTaskId" : 0,
"ticketObjectId" : 37,
"creatorObjectId" : 21,
"creatorObjectInstId" : 161,
"showEffort" : true,
"hasChecklist" : false,
"hasInstructions" : false,
"initTab" : "Tab2",
"availableActions" : [
  {
    "actionID" : 17,
    "actionName" : "Done",
    "isDefault" : true
  },
  {
    "actionID" : 109,
    "actionName" : "Cancel Plan",
    "isDefault" : false
  },
  {
    "actionID" : -1,
    "actionName" : "Check out",
    "isDefault" : false
  },
  {
    "actionID" : -4,
    "actionName" : "Assign",
    "isDefault" : false
  }
],
"readOnlyReq" : false,
"priceTask" : false
}
}
}
```

複数のタスクのチェックリスト情報を取得する  
メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/checklist?taskIds={taskIDs}

Java の例 :

http://<ServerURL>/RequestCenter/nsapi/transaction/v2/tasks/deliveryprocess?taskIds =176,177

サンプル応答 :

```
{
  "Map" : {
    "201" : [
      {
        "selected" : true,
        "stepID" : "1",
        "stepDescription" : "check1",
        "lastPerson" : "admin admin",
        "lastDate" : "01/21/2018",
        "ismandatory" : true
      },
      {
        "selected" : true,
        "stepID" : "2",
        "stepDescription" : "check2",
        "lastPerson" : "soauser5 soauser5",
        "lastDate" : "01/23/2018",
        "ismandatory" : true
      }
    ],
    "202" : []
  }
}
```

提供プロセス

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/deliveryprocess?taskIds={taskIDs}

応答 :

```
{
  "Map" : {
    "176" : [
      {
        "activityId" : 177,
```

```
"activity" : {
  "map" : {
    "ActivityTypeID" : "6",
    "AckOn" : null,
    "AgentID" : "4",
    "ActivityFormID" : "4"
  }
},
"customerName" : : デフォルトのサービス提供キュー
"performerName" : "admin admin",
"stateName" : "Completed",
"icon" : "task-completed.gif",
"subject" : "task1",
"dueOn" : "01/17/2018 07:00 PM",
"completedOn" : "01/16/2018 02:18 PM",
"parentActivityID" : 0,
"startDate" : "01/16/2018 01:59 PM",
"scheduledStart" : "01/16/2018 05:00 PM",
"depth" : 0,
"hasChildren" : false,
"isLast" : "false",
"group" : 0,
"waiting" : false
}
],
"177" : [
{
  "activityId" : 177,
  "activity" : {
    "map" : {
      "ActivityTypeID" : "6",
      "AckOn" : null,
      "AgentID" : "4",
      "ActivityFormID" : "4"
    }
  }
},
"customerName" : : デフォルトのサービス提供キュー
```



```

    "performerName" : "admin admin",
    "stateName" : "Completed",
    "icon" : "task-completed.gif",
    "subject" : "task1",
    "dueOn" : "01/17/2018 07:00 PM",
    "completedOn" : "01/16/2018 02:18 PM",
    "parentActivityID" : 0,
    "startDate" : "01/16/2018 01:59 PM",
    "scheduledStart" : "01/16/2018 05:00 PM",
    "depth" : 0,
    "hasChildren" : false,
    "isLast" : "false",
    "group" : 0,
    "waiting" : false
  }
]
}
}

```

**SM : タスク操作 API**

タスク利用可能なアクションの取得

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}

```

{
  "TaskData" : {
    "taskTypeId" : 3,
    "taskAgentId" : 0,
    "projectTaskId" : 0,
    "parentTaskId" : 0,
    "ticketObjectId" : 37,
    "creatorObjectId" : 57,
    "creatorObjectInstId" : 58,
    "showEffort" : true,

```

```
"hasChecklist" : false,

"hasInstructions" : false,

"initTab" : "Tab2",

"availableActions" : [

    {

        "actionID" : 119,

        "actionName" : "OK",

        "isDefault" : true

    },

    {

        "actionID" : -1,

        "actionName" : "チェックアウト",

        "isDefault" : false

    },

    {

        "actionID" : -4,

        "actionName" : "割り当て",

        "isDefault" : false

    }

],

"readOnlyReq" : false,

"priceTask" : false

}

}
```

指定タスクのスタッフの詳細を取得する

メソッド：GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/staffing

```
{
  "List" : [
    {
      "roleID" : 54520,
      "roleName" : "CSCC",
      "participant" : 1,
      "taskList" : [
        {
          "performer" : "Customer Service Call Center Queue",
          "performerID" : 12,
          "taskID" : 55601,
          "taskSubject" : "SAP Power Tools Purchasing - Change",
          "customerID" : 0
        }
      ]
    }
  ]
}
```

複数のタスクのスタッフ情報を取得する

メソッド：GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/staffing?taskIds={taskIDs}

Java の例 : http://<ServerURL>/RequestCenter/nsapi/transaction/v2/tasks/staffing?taskIds=161

サンプル応答 :

```
{
  "Map" : {
    "50" : [],
    "161" : [
      {
        "roleID" : 169,
        "roleName" : "perf1",
        "participant" : 1,
        "taskList" : [
          {
```

```
"performer" : デフォルトのサービス提供キュー
"performerID" : 2,
"taskID" : 162,
"taskSubject" : "siTask1",
"customerID" : 0
}
]
},
{
"roleID" : 170,
"roleName" : "perf2",
"participant" : 0,
"taskList" : [
{
"performerID" : 0,
"taskID" : 162,
"taskSubject" : "siTask1",
"customerID" : 11,
"customerName" : "soa user4"
}
]
},
{
"roleID" : 171,
"roleName" : "perf3",
"participant" : 1,
"taskList" : [
{
"performer" : "Vivek Verma",
"performerID" : 8,
"taskID" : 163,
"taskSubject" : "task2",
"customerID" : 0
}
]
},
{
```

```

    "roleID" : 172,
    "roleName" : "perf4",
    "participant" : 0,
    "taskList" : [
      {
        "performerID" : 0,
        "taskID" : 163,
        "taskSubject" : "task2",
        "customerID" : 2,
        "customerName" : : デフォルトのサービス提供キュー
      }
    ]
  }
]
}
}
}

```

実行者を更新する

メソッド : PUT

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/staffing

サンプル応答 :

```

{
  "staffing":{
    "participant" : 1,
    "taskList" : [
      {
        "performerID" : 2,
        "taskID" : 162
      }
    ]
  }
}

```

実行者リストを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/performerlist?name=\*&startRow=1&recordSize=10&responseType=json

```

{
  "performerList" : {
    "startRow" : 1,

```

```
"totalCount" : 5,
"recordSize" : 10,
"performers" : [
  {
    "id" : 4,
    "firstName" : "ProviderBusinessTenant_admin",
    "lastName" : "Queue",
    "homeOrganizationalUnitName" : "ProviderBusinessTenant_admin"
  },
  {
    "id" : 32,
    "firstName" : "Team2_admin",
    "lastName" : "Queue",
    "homeOrganizationalUnitName" : "Team2_admin"
  },
  {
    "id" : 1,
    "firstName" : "admin",
    "lastName" : "admin",
    "homeOrganizationalUnitName" : "Site Administration"
  },
  {
    "id" : 22,
    "firstName" : "nagadmin",
    "lastName" : "nagadmin",
    "homeOrganizationalUnitName" : "NagTestOU"
  },
  {
    "id" : 77,
    "firstName" : "shashi3",
    "lastName" : "admin2",
    "homeOrganizationalUnitName" : "Site Administration"
  }
]
}
```

スーパーバイザの更新

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/staffing

```
{
  "staffing":{
    "participant" : 0,
    "taskList" : [
      {
        "customerD" : 2,
        "taskID" : 162
      }
    ]
  }
}
```

要求データを取得する

メソッド : GET

REST URL :

/RequestCenter/nsapi/transaction/v2/requisitions/requisitiondata?requisitionIds=2&fetchDictionaryCaption=true&utid=18ffaec697217d06ff7e519f838805c6&responseType=json

サンプル応答 :

```
{
  "Requisitions" : [
    {
      "RequisitionId" : 2,
      "RequisitionEntries" : [
        {
          "RequisitionEntry" : {
            "Dictionaries" : [],
            "ServiceName" : "testSTService1",
            "FormFields" : [],
            "RequisitionEntryID" : 2,
            "Quantity" : 1
          }
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

タスク チェックリスト  
メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/task/{taskID}/checklist

サンプル応答

```

{
  "List":[
    {
      "selected":false,
      "stepID":"1",
      "stepDescription":"new item",
      "ismandatory":true
    },
    {
      "selected":false,
      "stepID":"2",
      "stepDescription":"new item",
      "ismandatory":true
    }
  ]
}

```

提供プロセス API

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/deliveryprocess

サンプル応答 :

```

{
  "deliveryProcessList" : {
    "startRow" : 1,
    "totalCount" : 2,
    "recordSize" : 20,
    "deliveryProcesses" : [
      {

```



```
"activityId" : 5830,
"activity" : {
  "map" : {
    "ActivityTypeID" : "6",
    "AckOn" : null,
    "AgentID" : "-1",
    "ActivityFormID" : "4"
  }
},
"customerName" : "デフォルトのサービス提供キュー",
"performer" : "デフォルトのサービス提供キュー",
"stateId" : "3",
"stateName" : "Completed",
"icon" : "task-completed.gif",
"subject" : "Create Si",
"dueOn" : "07-09-2018 07:57 PM",
"completedOn" : "07-06-2018 05:57 PM",
"parentActivityID" : 0,
"startDate" : "07-06-2018 05:57 PM",
"scheduledStart" : "07-06-2018 05:57 PM",
"depth" : 0,
"hasChildren" : false,
"isLast" : "false",
"group" : 0,
"waiting" : false
},
```

```
{  
  
  "activityId" : 5831,  
  
  "activity" : {  
  
    "map" : {  
  
      "ActivityTypeID" : "1",  
  
      "AckOn" : null,  
  
      "AgentID" : "0",  
  
      "ActivityFormID" : "4"  
  
    }  
  
  },  
  
  "customerName" : "デフォルトのサービス提供キュー" ,  
  
  "performerName" : "admin admin",  
  
  "performerRoleName" : "DeliveryBoy",  
  
  "stateId" : "2",  
  
  "stateName" : "Ongoing",  
  
  "icon" : "task-ongoing.gif",  
  
  "subject" : "DeliveryPlanTask",  
  
  "dueOn" : "07-11-2018 05:30 AM",  
  
  "completedOn" : "",  
  
  "parentActivityID" : 0,  
  
  "startDate" : "07-06-2018 05:57 PM",  
  
  "scheduledStart" : "07-10-2018 03:30 AM",  
  
  "depth" : 0,  
  
  "hasChildren" : false,  
  
  "isLast" : "false",
```

```

        "group" : 0,
        "waiting" : false
    }
]
}
}

```

省略されたタスクを表示する

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/deliveryprocess?showSkippedTasks=true

```

{
  "deliveryProcessList" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 20,
    "deliveryProcesses" : [
      {
        "activityId" : 6153,
        "activity" : {
          "map" : {
            "ActivityTypeID" : "1",
            "AckOn" : null,
            "AgentID" : "0",
            "ActivityFormID" : "4"
          }
        },
        "customerName" : "デフォルトのサービス提供キュー",
        "performer" : "デフォルトのサービス提供キュー",

```

```
        "stateId" : "4",
        "stateName" : "Skipped",
        "icon" : "task-skipped.gif",
        "subject" : "Test1",
        "dueOn" : "07-09-2018 04:00 PM",
        "completedOn" : "07-09-2018 06:57 AM",
        "parentActivityID" : 0,
        "startDate" : "07-09-2018 06:57 AM",
        "scheduledStart" : "07-09-2018 04:00 PM",
        "depth" : 0,
        "hasChildren" : false,
        "isLast" : "false",
        "group" : 0,
        "waiting" : false
    }
}
}
}
```

指定されたタスクの価格詳細を取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/pricing

サンプル応答 :

```
{
  "priceTask" : {
    "estimatedPrice" : 13000.0,
    "actualPrice" : 40700.0,
    "requisitionNumber" : 1322,
    "reqEntryNumber" : 1400
  }
}
```

指定したタスクの価格設定の詳細を更新する

メソッド : PUT

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/pricing

サンプルペイロード :

```
{ "priceTask" : { "estimatedPrice" : 40000, "actualPrice" : 40700, "requisitionNumber" : 1322, "reqEntryNumber" : 1400 } }
```

サンプル応答 :

```
{
  "status-message" : {
    "code" : "Success",
    "value" : "Price is successfully updated."
  }
}
```

要求 ID による要求エントリを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/requisitions/requisitionentries/id/{RequisitionID}?responseType=json

```
{
  "requisitionEntries" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 1,
    "requisitionEntry" : [
      {
        "requisitionEntryId" : 663219,
        "requisitionId" : 671445,
        "serviceId" : 2793,
        "serviceDescription" : "CD Performace Test Service",
        "serviceName" : "S1_CD Performace Test Service_V1",
        "standardDuration" : "Not Defined",
        "unitCost" : "$ 0.00",
        "subTotal" : "$ 0.00",
        "status" : "Rejected",
        "dueOnDate" : "06/29/2018",
        "dueOnDateRaw" : 1530264600000,
        "quantity" : 1,
        "maxQuantity" : 0,
        "serviceLevelDescription" : "",
        "percentageCompleted" : 0.0,
        "statusId" : 4,

```

```

        "expectedDuration" : 0.0,
        "reqEntryServiceURL" : "<a href=#\" onclick=\\\"GB_showFullScreen('S1_CD Performace
Test Service_V1','/RequestCenter/myservices/navigate.do?query=vieworderform&layout=popup_p&serviceid=2793&
requisitionId=671445&requisitionEntryId=663219')\\\">S1_CD Performace Test Service_V1</a>",
        "reqEntryServiceURLOnly" :
"/RequestCenter/myservices/navigate.do?query=requisitionentrystatus&reqid=7&reqentryid=41&formAction=displayEntryStatus&pe
rformerID=&serviceid=147&requisitionId=7",
        "requisitionStatusId" : 0,
        "requisitionStatus" : "Preparation",
        "customerName" : " ",
        "customerId" : 0,
        "customerOUId" : 0,
        "reviewsCount" : 0,
        "selfRating" : 0,
        "isServiceOrderable" : 1,
        "isServiceActive" : 1,
        "isCancelable" : 0,
        "ownerId" : 0,
        "daysToDueOnDate" : "Invalid short date format",
        "orderOnBehalf" : false
    }
}
]
}
}
}

```

**SM : エフォート API**

ユニット タイプのリストを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/efforts/unittypes?responseType=json

サンプル応答 :

```

{
  "entities": {
    "totalCount": 38,
    "hasMore": false,
    "units": [
      {
        "id": 1,
        "name": "Users",
        "logicName": "unitusers"
      },
      {
        "id": 2,
        "name": "Keyboards",

```

```
    "logicName": "unitkeyboards"
  },
  {
    "id": 3,
    "name": "Hours",
    "logicName": "unithours"
  },
  {
    "id": 4,
    "name": "Megabytes",
    "logicName": "unitmb"
  },
  {
    "id": 5,
    "name": "Servers",
    "logicName": "unitserver"
  },
  {
    "id": 6,
    "name": "CPUs",
    "logicName": "unitcpu"
  },
  {
    "id": 7,
    "name": "Departments",
    "logicName": "unitdepartment"
  },
  {
    "id": 8,
    "name": "Divisions",
    "logicName": "unitdivision"
  },
  {
    "id": 9,
    "name": "People",
    "logicName": "unitperson"
  },
  {
    "id": 10,
    "name": "Accounts",
    "logicName": "unitcustomer"
  },
  {
    "id": 11,
    "name": "Vendors",
    "logicName": "unitvendor"
  },
  {
    "id": 12,
    "name": "Desktops",
    "logicName": "unitdesktop"
  },
  {
    "id": 13,
    "name": "Devices",
    "logicName": "unitdevice"
  }
}
```

```
  },
  {
    "id": 14,
    "name": "Minutes",
    "logicName": "unitminute"
  },
  {
    "id": 15,
    "name": "Days",
    "logicName": "unitday"
  },
  {
    "id": 16,
    "name": "Weeks",
    "logicName": "unitweek"
  },
  {
    "id": 17,
    "name": "Months",
    "logicName": "unitmonth"
  },
  {
    "id": 18,
    "name": "Years",
    "logicName": "unityear"
  },
  {
    "id": 19,
    "name": "Applications",
    "logicName": "unitapplication"
  },
  {
    "id": 20,
    "name": "Databases",
    "logicName": "unitdatabase"
  },
  {
    "id": 21,
    "name": "Webservers",
    "logicName": "unitwebserver"
  },
  {
    "id": 22,
    "name": "Conference calls",
    "logicName": "unitconfcall"
  },
  {
    "id": 23,
    "name": "Message hours",
    "logicName": "unitmessagehour"
  },
  {
    "id": 24,
    "name": "Telco ports",
    "logicName": "unittelcoport"
  },
}
```



```
{
  "id": 25,
  "name": "Telco lines",
  "logicName": "unittelcoline"
},
{
  "id": 26,
  "name": "Calls",
  "logicName": "unitcall"
},
{
  "id": 27,
  "name": "Access lines",
  "logicName": "unitaccessline"
},
{
  "id": 28,
  "name": "Kilobytes",
  "logicName": "unitkilobyte"
},
{
  "id": 29,
  "name": "Gigabytes",
  "logicName": "unitgigabyte"
},
{
  "id": 30,
  "name": "Networks",
  "logicName": "unitnetwork"
},
{
  "id": 31,
  "name": "Domains",
  "logicName": "unitdomain"
},
{
  "id": 32,
  "name": "Cubicles",
  "logicName": "unitcubicle"
},
{
  "id": 33,
  "name": "Buildings",
  "logicName": "unitbuilding"
},
{
  "id": 34,
  "name": "SqFeet",
  "logicName": "unitsqfeet"
},
{
  "id": 35,
  "name": "Locations",
  "logicName": "unitlocation"
},
{
```

```

    "id": 36,
    "name": "Offices",
    "logicName": "unitoffice"
  },
  {
    "id": 37,
    "name": "BTUs",
    "logicName": "unitbtu"
  },
  {
    "id": 38,
    "name": "Other",
    "logicName": "unitother"
  }
]
}
}

```

カテゴリのリストを取得する

メソッド：GET

REST URL : RequestCenter/nsapi/transaction/v2/tasks/efforts/categories?responseType=json

サンプル応答：

```

{
  "entities": {
    "totalCount": 3,
    "hasMore": false,
    "categories": [
      {
        "billable": false,
        "id": 1,
        "name": "Labor"
      },
      {
        "billable": false,
        "id": 2,
        "name": "Materials"
      },
      {
        "billable": false,
        "id": 3,
        "name": "Other"
      }
    ]
  }
}

```

タスク ID によるエフォート エントリの取得

REST URL : RequestCenter/nsapi/transaction/v2/tasks/{taskID}/efforts?responseType=json

サンプル応答：

```
{
  "entities": {
    "totalCount": 4,
    "strReportDate": "01/31/2018",
    "billingID": 0,
    "totalExpense": "53.00",
    "items": [
      {
        "billingID": 29,
        "refObjectID": 0,
        "refObjectInstID": 0,
        "unitID": 12,
        "quantity": "1.0",
        "pricePerUnit": "50.0",
        "contextObjectID": 0,
        "contextObjectInstID": 0,
        "categoryID": 1,
        "serviceID": 0,
        "personID": 0,
        "arealID": 0,
        "reportDate": 1517404759000,
        "description": "from sm user1",
        "userID": 0,
        "statusID": 0,
        "rateID": 0,
        "rate": 0,
        "timeSpentMin": 0,
        "strReportDate": "01/31/2018",
        "expense": "50.00",
        "uirow": 0
      },
      {
        "billingID": 35,
        "refObjectID": 0,
        "refObjectInstID": 0,
        "unitID": 3,
        "quantity": "1.0",
        "pricePerUnit": "1.0",
        "contextObjectID": 0,
        "contextObjectInstID": 0,
        "categoryID": 1,
        "serviceID": 0,
        "personID": 0,
        "arealID": 0,
        "reportDate": 1517417940000,
        "description": "ee",
        "userID": 0,
        "statusID": 0,
        "rateID": 0,
        "rate": 0,
        "timeSpentMin": 0,
        "strReportDate": "01/31/2018",
        "expense": "1.00",
        "uirow": 0
      },
      {

```

```
"billingID": 36,
"refObjectID": 0,
"refObjectInstID": 0,
"unitID": 3,
"quantity": "1.0",
"pricePerUnit": "1.0",
"contextObjectID": 0,
"contextObjectInstID": 0,
"categoryID": 1,
"serviceID": 0,
"personID": 0,
"areaID": 0,
"reportDate": 1517418000000,
"description": "dffffdffd",
"userID": 0,
"statusID": 0,
"rateID": 0,
"rate": 0,
"timeSpentMin": 0,
"strReportDate": "01/31/2018",
"expense": "1.00",
"uirow": 0
},
{
  "billingID": 37,
  "refObjectID": 0,
  "refObjectInstID": 0,
  "unitID": 3,
  "quantity": "1.0",
  "pricePerUnit": "1.0",
  "contextObjectID": 0,
  "contextObjectInstID": 0,
  "categoryID": 1,
  "serviceID": 0,
  "personID": 0,
  "areaID": 0,
  "reportDate": 1517418060000,
  "description": "ssss",
  "userID": 0,
  "statusID": 0,
  "rateID": 0,
  "rate": 0,
  "timeSpentMin": 0,
  "strReportDate": "01/31/2018",
  "expense": "1.00",
  "uirow": 0
}
]
}
```

エフォート エントリを作成する

メソッド : POST

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/efforts?responseType=json

サンプル ペイロード :

```
{
  "billing": {
    "unitID": 12,
    "categoryID": 1,
    "description": "from sm user1",
    "pricePerUnit": "50.0",
    "quantity": "1.0"
  }
}
```

サンプル応答 :

```
{
  "status-message": {
    "code": "Success",
    "value": "The new Task Effort Entry is successfully added."
  }
}
```

エフォート エントリを更新する

メソッド : PUT

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/efforts/{billingID}

サンプル ペイロード :

```
{
  "billing": {
    "unitID": 17,
    "categoryID": 3,
    "description": "update description",
    "pricePerUnit": "50.0",
    "quantity": "3"
  }
}
```

サンプル応答 :

```
{
  "status-message": {
    "code": "Success",
    "value": "Task Effort Entry is successfully updated."
  }
}
```

エフォート エントリを削除する

メソッド : Delete

REST URL : /RequestCenter/nsapi/transaction/v2/tasks/{taskID}/efforts/{billingID}

サンプル応答 :

```
{
```

```
"status-message": {
  "code": "Success",
  "value": "Task Effort Entry is successfully deleted."
}
}
```

### SI一括更新のクォータとポリシー

メソッド : GET

REST URL : /RequestCenter/nsapi/serviceitem/v2/process?isBatch=true&commitType=2&batchSize=50)

isBatch が指定されていないか、または false として送信されている場合、API の動作は次のとおりです（非一括操作）。

1. ポリシーとクォータがサービス項目タイプに対して有効になっている場合、これらは作成、更新操作のサービス項目に対して実行されます。
2. コミットタイプと batchSize の設定は使用されません。操作は常に部分的であり、すべての適切なサービス項目が操作され、その他は失敗として報告されます。
3. ただし、更新操作と作成操作では、特定のサービス項目（si9 など）に違反している場合、si9 と si9 を超えるサービス項目に対しては操作は行われません。

作成操作の commitType が 2 の場合、サービス項目 s1 のいずれかに違反すると、s1 より前のすべてのサービス項目のみが作成され、サービス項目 s1 以降は作成されません。

注：削除のポリシー チェックは実行されませんが、ポリシーが合計クォータ ポリシーの場合は対応する属性カウントが、ポリシーがカウントの場合はサービス項目カウントが更新されます。これは正常な予期された動作です。

サンプル応答：

```
<serviceitem>
<name>testSIType01</name>
<serviceItemData>
  <serviceItemAttribute name="Name">D_1</serviceItemAttribute>
  <serviceItemAttribute name="Age">1</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">2451</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max11</serviceItemAttribute>
</serviceItemData>
<subscription>
  <loginID>testuser01</loginID>
  <ouname>testOU1</ouname>
  <accountName>testTenant1</accountName>
  <requisitionEntryID>1</requisitionEntryID>
```

```
</subscription>
</serviceItemData>
<serviceItemData>
  <serviceItemAttribute name="Name">D_2</serviceItemAttribute>
  <serviceItemAttribute name="Age">2</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">245</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max12</serviceItemAttribute>
  <subscription>
    <loginID>testuser01</loginID>
    <ouname>testOU1</ouname>
    <accountName>testTenant1</accountName>
    <requisitionEntryID>1</requisitionEntryID>
  </subscription>
</serviceItemData>
<serviceItemData>
  <serviceItemAttribute name="Name">D_3</serviceItemAttribute>
  <serviceItemAttribute name="Age">1</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">245</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max13</serviceItemAttribute>
  <subscription>
    <loginID>testuser01</loginID>
    <ouname>testOU1</ouname>
    <accountName>testTenant1</accountName>
    <requisitionEntryID>1</requisitionEntryID>
  </subscription>
</serviceItemData>
<serviceItemData>
  <serviceItemAttribute name="Name">D_4</serviceItemAttribute>
  <serviceItemAttribute name="Age">1</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">245</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max14</serviceItemAttribute>
```

```

<subscription>
  <loginID>testuser01</loginID>
  <ouname>testOU1</ouname>
  <accountName>testTenant1</accountName>
  <requisitionEntryID>1</requisitionEntryID>
</subscription>
</serviceItemData>
<serviceItemData>
  <serviceItemAttribute name="Name">D_5</serviceItemAttribute>
  <serviceItemAttribute name="Age">0</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">245</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max15</serviceItemAttribute>
<subscription>
  <loginID>test01</loginID>
  <ouname>testOU1</ouname>
  <accountName>testTenant1</accountName>
  <requisitionEntryID>1</requisitionEntryID>
</subscription>
</serviceItemData>
</serviceItem>

```

## 12.1\_Patch\_v3 で導入

### UCSD 6.6 認定

Prime Service Catalog 12.1 パッチ V3 は、Cisco UCSD 6.6 で検証および認定されています。

### CloudCenter の機能拡張

このパッチ リリースから、Prime Service Catalog は CloudCenter の次の機能をサポートするようになりました。

### サービス アカウント ベースの CloudCenter 統合

CloudCenter 統合は次のように区別されます。

- ユーザ アカウント ベースの統合 :  
CloudCenter 接続は、作成したユーザに関連付けられます。ユーザがアプリケーションを展開すると、このユーザのみがアプリケーションの所有者となり、アプリケーションでアクションを実行できます。ユーザ ベースの統合では、CloudCenter に存在しない場合は、すべてのユーザがCloudCenter にプッシュされてユーザ レコードが作成されます。  
**注 :** このパッチ リリースにアップグレードする場合、アップグレード前の既存のクラウド センター接続は、デフォルトでユーザ アカウント ベースの統合と見なされます。
- サービス アカウント ベースの統合 :



CloudCenter 接続は、CloudCenter に送信されるすべての要求に使用される共通サービス アカウントに関連付けられます。これには、作成者に関係なくアプリケーション展開、操作、およびアクションが含まれます。この場合、ユーザは CloudCenter にプッシュされません。

アプリケーションを展開する権限を持つユーザがこれらのアプリケーションの所有者になり、これらの展開または VM で読み取り/書き込みアクセス権を持つすべてのユーザは、それらのアプリケーションでアクションを実行できます。

これは、サイト管理者または統合管理者が新しい CloudCenter 統合を作成するときの接続ごとのワнтаイム設定です。**[統合モジュール (Integrations module)] > [新規統合 (New Integration)] > [Cisco CloudCenter]** に移動します。**[統合 (Integration)]** タイプとして、CloudCenter 統合に基づいて **[ユーザ アカウント (User Account)]** または **[サービス アカウント (Cloud Account)]** を選択します。

**重要：** この設定は、統合の作成後に変更することはできません。

### ユーザ アカウント ベースの統合

以下の機能は、ユーザ ベースの CloudCenter 統合に固有です。

#### Brownfield 展開/VM のインポート

Prime Service Catalog は、Prime Service Catalog の外部で作成された展開および VM のインポートをサポートしています。これら展開と VM は、CloudCenter で作成された Prime Service Catalog によって検出され、CloudCenter との統合が作成されると、**[サービス項目 (Service Items)]** ページに表示されます。

**[ブラウнフィールド導入のインポート (Import Brownfield Deployments)]** と **[ブラウнフィールド VM のインポート (Import Brownfield VMs)]** の 2 つの新しいオプションが **[CloudCenter 統合の作成 (Create CloudCenter Integration)]** 画面に表示されます。作成時に、その接続の展開や VM をインポートすることを選択できます。この設定は、接続の作成後いつでも編集できます。

**注：** サービス アカウント ベースの統合では、ブラウнフィールド VM と展開のインポートを無効にすることをお勧めします。サービス アカウント ベースの統合では、Prime Service Catalog へのブラウнフィールド VM インポートの動作に一貫性がない場合があります。CloudCenter に展開と VM がある場合、サービス アカウントによって所有されているかどうかに関係なく、Prime Service Catalog にインポートされると管理者に割り当てられます。CloudCenter の VM でアクションを実行する権限には所有権が不可欠なため、これは一貫性のない動作につながります。

ただし、ブラウнフィールド VM と展開を Prime Service Catalog にインポートする必要がある場合は、それらがすべて CloudCenter のサービス アカウントによって所有されていることを確認します。

#### ブラウнフィールド展開と VM のユーザの関連付け

**[Cloud Center 接続 (Cloud Center Connection)]** ドロップダウン メニューに新しく追加された **[ユーザのマッピング (Map Users)]** アクションでは、インポートされたオブジェクトにユーザをマッピングします。ユーザは、接続とインポートの直後に統合にマッピングする必要があります。ユーザが統合にマッピングされていない場合、**[管理対象 VM の同期 (Sync Managed VMs)]** 操作は失敗します。

ブラウнフィールド展開と VM は CloudCenter から Prime Service Catalog にインポートされるため、これらの展開と VM のユーザ/所有者も Prime Service Catalog に関連付ける必要があります。次のシナリオでは、ユーザの扱いが異なります。

- Prime Service Catalog にユーザがすでに存在する場合：  
ユーザは、**[サービス項目 (Service Items)]** ページで、ユーザが所有するインポートされたすべての展開にアクセスできます。展開と VM ですべての操作を実行できるようになります。
- ユーザが Prime Service Catalog に存在しない場合：  
展開のインポートで、サービス項目の顧客が null です。その後、ユーザが LDAP / IDP (SAML) / Prime Service Catalog データベースを介して Prime Service Catalog で手動で作成された場合、インポートされた展開と VM がユーザに関連付けられます。

**重要：** ユーザの作成は、ユーザの詳細が外部ユーザ ID にマッピングされるように行う必要があります。LDAP / SAML の適切なフィールドをマッピングするのは、管理者の責任です。

### 管理対象 VM の同期

[サービス カタログ (Service Catalog) ] > [サービス項目 (Service Items) ] ページで、すべてのユーザが新しいオプション **[同期管理対象VM ( Synch Managed VMs ) ]** を使用できるようになりました。このオプションは、ログインしているユーザが所有するすべての VM をリロードして表示します。CloudCenter のユーザが所有する管理対象 VM のみが同期されます。

### 選択したアプリケーション プロファイルのインポート

CloudCenter 接続の選択したアプリケーション プロファイルのみを更新および再インポートするための新しいオプション **[再インポート (Re-Import) ]** が導入されました。CloudCenter でアプリケーション プロファイルに変更が加えられた場合、選択したアプリケーション プロファイルの **[再インポート (Re-Import) ]** をクリックすると、変更が Prime Service Catalog に反映され、サービスが再生成されます。

このオプションは、**[統合 (Integrations) ] > [CloudCenter 統合 ( CloudCenter Integration ) ] > [統合の管理 ( Manage Integrations ) ] > [アプリケーション プロファイル ( Application profiles ) ]** で使用できます。

### 展開の機能拡張

展開されたアプリケーションでの完全同期の実行

Prime Service Catalog は、アプリケーションと VM の両方の完全同期を実行します。ステータスは、CloudCenter 上のアプリケーションまたは VM のステータスに応じて取得され、VM の詳細 (CPU、RAM、HD、パブリックおよびプライベート IP アドレスなど)、VM のステータスが **[サマリーの詳細 (Summary Details) ]** ページに表示されます。

#### Details

Name	42356edf-9dd1-c9a2-e1c0-71bf46e7861f
Display Name	CentOS_2
External ID	42356edf-9dd1-c9a2-e1c0-71bf46e7861f
Status	Active
Host Name	cqjw-51dd6591a
Public IP	172.23.14.76
Private IP	172.23.14.76
Start Time	2018-05-22 06:02:37.403
Cloud Connection ID	1
Cloud VM Type	DEPLOYMENT_VM
Cloud ID	1
Cloud Account ID	1
Cloud Region ID	1
External VM ID	940
External User ID	2
Termination Protection	0
Job ID	1822
Virtual CPUs	1
Memory (MB)	1024
Storage (GB)	10

### グループ用の CloudCenter サービスの注文

CloudCenter 展開をオーダーする権限を持つユーザは、ユーザが属するグループの CloudCenter サービスをオーダーできるようになりました。注文したサービス項目に対する読み取り/書き込み権限が選択したグループに提供され、グループのすべてのユーザが展開に対してアクションを実行できます。

**[管理 ( Administration ) ] > [設定 ( Settings ) ]** に [サービス フォームにグループ リストを表示 ( Display Group List in Service Form ) ] という新しいオプションが導入されました。有効にすると、注文する CloudCenter サービスのいずれかのサービス フォームに [グループ ( Group ) ] フィールドが表示されます。リストには、オーダーのユーザが属するすべてのグループが含まれます。管理設定でこのオプションを有効にできるのは、サイト管理者のみです。

#### アプリケーション プロファイル メタデータ タグのインポート

アプリケーション プロファイルにカスタム プロパティのメタデータが設定されている場合、CloudCenter 接続のインポート中にメタデータも Prime Service Catalog にインポートされます。次に、メタデータ情報を使用して新しいディクショナリと新しいサービス フォームを作成し、このサービス フォームを CloudCenter アプリケーション サービスに関連付けます。このようなサービスを注文する際、ユーザはメタデータ フィールドの詳細を表示したり、追加のメタデータをその場で追加したりできます。CloudCenter のアプリケーション プロファイルのメタデータの詳細については、「[CloudCenter のドキュメント](#)」の「[カスタム プロパティのメタデータ](#)」を参照してください。



**注：**システムによって作成されたディクショナリおよびサービス フォームは改ざんしないでください。改ざんされている場合、CloudCenter の展開は失敗します。

**注：**CloudCenter サービスの注文時に、メタデータ フォームに特殊文字を追加できます。ただし、“ ” 特殊文字の場合は、次のエスケープ シーケンスを追加する必要があります。

\\ “System is having 32 GB\\”

#### CloudCenter アプリケーション展開中のネットワークのオーバーライド

これで、CloudCenter アプリケーションが展開されているネットワークをオーバーライドできます。

[ネットワーク ( Network ) ] ディクショナリ フィールドは、デフォルトでは表示されません。サービス デザイナーまたはサービスの読み取り/書き込み権限を持つユーザは、次の手順に従って、特定のサービスでこのオプションを使用できるようになります。

1. **[サービス デザイナー ( Service Designer ) ] > [アクティブ フォーム コンポーネント ( Active Form Component ) ]** を選択します。
2. 設定するフォームを選択し、フォームにすでに追加されているディクショナリがあると想定して、**[表示プロパティ ( Display Properties ) ]** タブをクリックします。
3. **[このフォームで使用するディクショナリ ( Dictionaries Used in This Form ) ]** セクションから、フィールドを編集するディクショナリを展開します。
4. **[ネットワーク ( Network ) ]** ディクショナリの名前をクリックし、右側にプロパティを表示します。
5. **入力タイプ** を非表示からテキストに変更します。
6. **保存** します。

**注：**ネットワーク オプションが有効で、このフィールドが空白の場合、設定されたクラウドごとのデフォルト ネットワークが考慮されます。

#### スケーリング ポリシーのサポート

**Prime Service Catalog** は、**CloudCenter** で定義されたスケーリング ポリシーをサポートしています。サービス フォームで利用可能な **CloudCenter** サービスを注文する際に、適切なシステム タグを選択することで、**CloudCenter** のスケーリング ポリシーにアプリケーションの展開を関連付けることができます。次のスクリーンショットでは、スケーリング タグを選択すると、関連付けられたポリシーが自動的に入力されます。

**注：**複数のタグを選択した場合は、優先順位の高いスケーリング ポリシーが自動的に適用されます。

▼ CentOS\_2

Instance Type \*

Tags

- DiskScale
- MemoryScale
- SreeCPUScale
- QACPU SCALE**
- QADISKSCALE

Number of nodes (min:2, max:4) \*

Scaling Policy

#### 展開時の新しいアクション

- 同期展開

[展開の同期 (Sync Deployment) ] アクションでは、各展開を **CloudCenter** と同期し、特定の展開の現在のステータスを取得できます。たとえば、**CloudCenter** からの展開に変更が加えられた場合（展開が一時停止された、または新しいノードが追加された場合など）、指定されたポーリング間隔や手動インポートを待機する代わりに、このアクションを使用してそのような変更をすぐに **Prime Service Catalog** に同期できます。

- 展開の移行

展開の移行操作は、**Cloud Center** での展開操作の移行をトリガーします。

デフォルトでは、展開のコピーがターゲット クラウド上に作成され、展開は元のクラウド上で削除されます。ただし、展開サービス フォームの **keepExistingDeployment** オプションで [はい (Yes) ] を選択することで、元のクラウドで展開を維持できます。

## CentOS 6.x (v2.0) (CC1)

## ▼ General Information

\* Name

Description

\* App Version 2.0

\* Tag

- Development
- Staging
- Hour
- Day

\* Deployment Environment \* Cloud \* Cloud Account Aging Policy keepExistingDeployment  Yes  No

## ▼ CentOS\_2

\* Instance Type 

Tags

- Development
- Staging

SUBMIT

CANCEL

**VM の機能拡張**

管理対象 VM の新しいアクション

このパッチ リリースは、管理対象 VM で次の新しい操作をサポートします。

**注：**これらの操作は、サービス アカウント ベースの統合の展開または VM に対する読み取り/書き込み権限を持つユーザのみが使用でき、ユーザはユーザ ベースの統合の VM の顧客である必要があります。

- アプリケーション VM の起動/停止/再起動：  
VM のステータスに応じて、次の操作を実行できます。
  - 開始： VM の電源オンと開始
  - 停止： VM の停止と電源オフ
  - 再起動： VM を正常にシャットダウンし、再起動します。
- ボリュームの接続：  
ボリュームの接続オプションを使用すると、N 階層のアプリケーションですべての階層タイプに複数のボリュームを接続できます。ボリュームごとに、各ルート ディスクのサイズとストレージタイプを指定する必要があります。ボリュームの接続操作は、他のユーザまたは自分用にオーダーできるサービスを呼び出します。詳細については、「[CloudCenter のドキュメント](#)」の「[複数のボリュームを階層に接続する](#)」を参照してください。
- ボリュームの接続解除：  
ボリュームの接続解除オプションは、任意のユーザの VM からディスク ボリュームを削除するために使用されます。
- スナップショットの作成：  
指定した VM のイメージスナップショットを作成します。

- アプリケーション VM の同期 :  
このVMオプションは、ポーラー トリガーまたは手動インポートを待機する代わりに、CloudCenter から Prime Service Catalog に VM の最新情報（管理対象外 VM、展開済み VM、およびインポート済み VM）を取得します。展開の一部である場合は、展開の詳細も同期します。

#### 管理対象外 VM のインポート

Prime Service Catalog にインポートできる brownfield VM には 2 種類あります。

- 管理対象 VM : CloudCenter によって管理される VM。例 : CloudCenter で作成された展開の結果として作成された VM。または、クラウドから CloudCenter にインポートされた VM。
- 管理対象外 VM : CloudCenter に接続されているクラウド アカウントに存在する VM。これらは CloudCenter にはインポートされません。

ユーザは管理対象外 VM に制限付き権限を持っているため、Prime Service Catalog VM からこれらの VM を管理するには、まず VM を Prime Service Catalog にインポートする必要があります。

[サービス項目 (Service Items) ] ページで、管理対象外 VM で許可されるアクションは次のとおりです。

- 終了
- Import UnManagedVM : VM を Prime Service Catalog と CloudCenter にインポートし、VM のステータスを管理対象 VM に変換します。

#### VM のコンソール アクセス

VM の [サービス項目 (Service Items) ] ページの Web ベースのコンソールから CloudCenter VM にアクセスできるようになりました。管理対象 VM で読み取り/書き込み権限を持つユーザは、VM コンソールにアクセスできます。

- Windows VM の場合 :  
[ログイン詳細 ( Login Details ) ] ボタンをクリックしてログイン クレデンシャルを表示し、[RDP] ボタンをクリックします。新しい Web ページでコンソールが起動します。
- Linux VM の場合 :  
[SSH] ボタンをクリックして、新しい Web ページでコンソールを起動します。

#### 拡張サービス項目の検索

新しい検索条件表示名を使用して、サービス項目の表示名でサービス項目をフィルタリングできます。

#### VM でのカスタム ライフサイクル操作のサポート

CloudCenter VM でカスタム アクションをサポートするには、次の手順に従います。

1. サービス項目タイプとして [アプリケーション仮想マシン (Application Virtual Machine) ] を選択して、サービス項目ベースのディクショナリを作成します。
  - a. [サービス デザイナー ( Service Designer ) ] > [ディクショナリ ( Dictionaries ) ] を選択します。
  - b. [新規 ( New ) ] > [新しいディクショナリ ( New Dictionary ) ] を選択して、[新しいディクショナリ (New Dictionary) ] ページを表示します。
  - c. [新しい内部ディクショナリの追加 (Add New Internal Dictionary) ] セクションの [サービス項目 (Service Item) ] フィールドに「アプリケーション仮想マシン」と入力し、サービス項目を選択します。
  - d. ディクショナリ名やグループ名などの詳細を入力します。
  - e. [ディクショナリ属性 (Dictionary Attributes) ] セクションで、[フィールドの追加 ( Add Field ) ] をクリックして、ユーザ定義フィールド ShortName および ExecutionID を追加します。
  - f. カスタム アクションに必要なディクショナリ内のすべてのフィールド (カスタム FTL で使用されるすべてのフィールド) を選択します。[ExecutionID] フィールドがディクショナリに追加されていることを確認します。
  - g. [Save Dictionary] をクリックします。

Use	Name	Type	Maximum	Decimals	Multivalue	Show In Grid	Encrypt	PII
<input checked="" type="checkbox"/>	Name	ServiceItemIdentifier	128	0				
<input checked="" type="checkbox"/>	DisplayName	Text	128	0		<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	ExternalID	Text	128	0				
<input checked="" type="checkbox"/>	Status	Text	128	0				
<input type="checkbox"/>	HostName	Text	128	0				
<input type="checkbox"/>	PublicIP	Text	128	0				
<input type="checkbox"/>	PrivateIP	Text	512	0				
<input type="checkbox"/>	StartTime	Text	128	0				
<input type="checkbox"/>	EndTime	Text	128	0				
<input checked="" type="checkbox"/>	CloudConnectionID	Number	9	0				
<input type="checkbox"/>	CloudID	Text	128	0				
<input checked="" type="checkbox"/>	CloudVMType	Text	128	0				
<input type="checkbox"/>	CloudID	Number	9	0				
<input type="checkbox"/>	CloudAccountID	Number	9	0				
<input type="checkbox"/>	CloudRegionID	Number	9	0				
<input checked="" type="checkbox"/>	ExternalVMID	Number	9	0				
<input type="checkbox"/>	ExternalUserID	Number	9	0				
<input type="checkbox"/>	ProtectTermination	Number	9	0				
<input type="checkbox"/>	Actions	Text	512	0				
<input type="checkbox"/>	CPU	Number	9	0				
<input type="checkbox"/>	JobID	Text	128	0				
<input type="checkbox"/>	RAM	Number	9	0				
<input type="checkbox"/>	Storage	Number	9	0				
<input type="checkbox"/>	InstanceType	Text	128	0				
<input type="checkbox"/>	Tags	Text	128	0				
<input type="checkbox"/>	ScalingPolicy	Text	128	0				
<input type="checkbox"/>	AggregPolicy	Text	128	0				
<input type="checkbox"/>	SecurityProfile	Text	128	0				
<input type="checkbox"/>	NumberCPUS	Number	9	0				
<input type="checkbox"/>	Network	Text	128	0				
<input type="checkbox"/>	OSFamily	Text	128	0				
<input type="checkbox"/>	CustomerID	Number	9	0				
<input type="checkbox"/>	RequestorID	Number	9	0				
<input type="checkbox"/>	RequestorEntityID	Number	9	0				
<input type="checkbox"/>	OrganizationalUnitID	Number	9	0				
<input type="checkbox"/>	AccountID	Number	9	0				
<input type="checkbox"/>	AssignedDate	Date	0	0				
<input type="checkbox"/>	SubmittedDate	Date	0	0				
<input type="checkbox"/>	EstimatedPrice	ServiceItemPrice	512	0				
<input checked="" type="checkbox"/>	ErrorCode	Text	128	0				
<input checked="" type="checkbox"/>	ShortName	Text	512	0				
<input checked="" type="checkbox"/>	ExecutionID	Text	128	0				

2. アクティブ フォーム コンポーネントを作成し、上記で作成したディクショナリをフォームに追加します。
  - a. [サービス デザイナー ( Service Designer ) ] > [アクティブ フォーム コンポーネント ( Active Form Component ) ] を選択します。
  - b. [新規 ( New ) ] > [アクティブフォームコンポーネント ( Active Form Component ) ] を選択します。
  - c. 新しいフォームの名前と簡単な説明を入力します。
  - d. フォーム グループ フィールドをクリックし、フォームに関連付けるグループを 1 つ選択します。
  - e. [フォームの保存 ( Save Form ) ] をクリックします。
  - f. [フォーム コンテンツ ( FormContent ) ] タブを選択し、[ディクショナリの追加 ( Add Dictionaries ) ] をクリックします。
  - g. [ディクショナリの追加 ( Add Dictionaries ) ] ダイアログ ボックスで、ステップ 1 で作成したディクショナリを検索して選択します。[追加 ( Add ) ] をクリックします。  
**注：** フォームを送信する前に、必ずクラウドセンターに属する VM を選択し、すべての必須ディクショナリ フィールドに入力してください。
  - h. [フォームの保存 ( Save Form ) ] をクリックします。

3. 必ず [短い名前 ( Short Name ) ] フィールドに入力してください。データ取得ルールを作成することで完了します。
  - a. 作成したアクティブ フォーム コンポーネントの [アクティブ フォーム ルール ( Active Form Rules ) ] タブに移動します。

- b. **[新しいルール ( New Rule ) ]** > **[新しいデータ取得ルール ( New Data Retrieval Rule ) ]** の順に選択します。
- c. **[データ取得ルール ( Data Retrieval Rule ) ]** ウィザードの最初のページで、ルールの一意の名前と説明を入力し、**[ルールタイプ ( Rule Type ) ]** を **[区別ルール ( Distributing Rule ) ]** として指定します。
- d. 次の図のようにウィザードでその他詳細を指定し、**[ルールの保存 ( Save Rule ) ]** で短縮名を入力します。

4. カスタム サービス ( CustomOperationService など ) を作成し、サービスにフォームを追加します。
  - a. **[サービス デザイナー ( Service Designer ) ]** > **[新規 ( New ) ]** > **[新規サービス ( New Service ) ]** を選択します。
  - b. 提供されたフィールドに詳細を入力します。
  - c. **[Add This Service]** をクリックします。
  - d. サービスを追加した後、**[全般 ( General ) ]** タブに情報を入力することで設定を開始できます。
  - e. **[保存 ( Save ) ]** をクリックします。
  - f. 作成したサービスの **[フォーム ( Form ) ]** タブをクリックします。
  - g. ウィンドウの左下の **[フォームの追加 ( Add Forms ) ]** をクリックします。 **[フォームの追加 ( Add Form ) ]** ポップアップ ウィンドウが表示されます。
  - h. **[検索 ( Search ) ]** フィールドに、ステップ 2 で作成したフォーム名を入力します。
  - i. フォームを確認し、**[追加 ( Add ) ]** をクリックします。



5. サービスの **[プラン ( Plan )]** で下に指定されている 2 つのタスクを作成します。

The screenshot shows the 'Delivery Plan For Service Echo Service' configuration page. The 'Tasks' section contains a table with the following data:

Task	By	Time	Subtasks	Subtotal	
EchorepoTask		10:00	0:00	10:00	
Update history task		10:00	0:00	10:00	
				Total project duration	20:00
				Approximate days (as per working hours per day)	2.50

The 'General' section for the 'EchorepoTask' shows the following configuration:

- Workflow Type: Chiragent
- Task name: EchorepoTask
- Subtasks execute: one after the other (sequentially)
- Priority: Normal
- Duration: 10:00 hours
- Effort: 10:00 hours
- Condition: (Empty field)
- Form data for start date: (Empty field)
- Options:
  - Allow a scheduled start date
  - Evaluate condition when delivery phase starts (if condition evaluates to "false", times will be computed as zero)
  - Evaluate condition when task becomes active (delivery schedule will always include this task's duration)
  - Re-evaluate expressions (participant assignment expressions and task title expression) as plan advances
  - Do not allow cancellation of service after task starts
  - Display Effort sub-page on a delivery task

VM 操作タスクの実行と操作完了タスクは必須です。更新履歴タスクはオプションです。これらの各タスクの設定の詳細については、それぞれの項を参照してください。

- [VM 操作タスクの実行](#)
  - [Error! Reference source not found.](#)
6. サービスを保存します。
  7. **[サービス項目マネージャ ( Service Item Manager )]** > **[サービス項目の設計 ( Design Service Items )]** > **[アプリケーション仮想マシン ( Application Virtual Machines )]** > **[関連サービス ( Associated Services )]** でカスタム サービスを追加します。
  8. **[サービス項目マネージャ ( Service Item Manager )]**-**[サービス項目の設計 ( Design Service Items )]** > **[アプリケーション仮想マシン ( Application Virtual Machines )]** > **[操作 ( Operations )]** に移動して操作を追加し、サービスを操作に関連付けます。これにより、VM の歯車アイコンにサービスが表示されます。
  9. サーバ キャッシュをクリアし、サーバを再起動します。  
カスタム操作が正常に作成されたことを確認するには、カスタム サービスを注文します。

## 展開でのカスタム アクション操作のサポート

CloudCenter 展開でカスタム アクションをサポートするには、次の手順に従います。

1. サービス項目タイプとして **アプリケーション スタック** を選択して、サービス項目ベースのディクショナリを作成します。
  - a. **[サービス デザイナー ( Service Designer )]** > **[ディクショナリ ( Dictionaries )]** を選択します。
  - b. **[新規 ( New )]** > **[新しいディクショナリ ( New Dictionary )]** を選択して、**[新しいディクショナリ ( New Dictionary )]** ページを表示します。
  - c. **[新しい内部ディクショナリの追加 ( Add New Internal Dictionary )]** セクションの **[サービス項目 ( Service Item )]** フィールドに「**アプリケーション スタック**」と入力し、サービス項目を選択します。
  - d. ディクショナリ名やグループ名などの詳細を入力します。
    - a. **[ディクショナリ属性 ( Dictionary Attributes )]** セクションで、**[フィールドの追加 ( Add Field )]** をクリックして、ユーザ定義フィールド **ShortName**、**JobID**、および **ExecutionID** を追加します。
    - e. カスタム アクションに必要なディクショナリ内のすべてのフィールド (カスタム FTL で使用されるすべてのフィールド) を選択します。**[JobID]** フィールドが選択され、**[ExecutionID]** フィールドがディクショナリに追加されていることを確認します。
    - f. **[ディクショナリを保存 ( Save Dictionary )]** をクリックします

Use	Name	Type	Maximum	Decimals	Multivalue	Show In Grid	Encrypt	PII
<input checked="" type="checkbox"/>	Name	ServiceIdentifier	128	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	DisplayName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Description	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ServiceTierID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AppVersion	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	DeploymentEnvironment	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Cloud	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CloudAccount	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ShortName	Text	32	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	JobID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AppProfileName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AppProfileDisplayName	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	StackID	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Status	Text	32	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ApplicationTemplateID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ApplicationURL	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Flag	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ImageURL	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	StartTime	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	CloudConnectionID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ExternalUserID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CustomerID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RequestorID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RequestorEntryID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OrganizationUnitID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AccountID	Number	9	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	AssignedDate	Date	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	SubmittedDate	Date	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	EstimatedPrice	ServiceItemPrice	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ErrorCode	Text	128	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ErrorDescription	Text	512	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	ExecutionID	Number	25	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. 2つのアクティブフォームコンポーネントを作成し、上記で作成したディクショナリをフォームに追加します。
  - a. **[サービス デザイナー ( Service Designer ) ] > [アクティブ フォーム コンポーネント ( Active Form Component ) ]** を選択します。
  - b. **[新規 ( New ) ] > [アクティブフォームコンポーネント ( Active Form Component ) ]** を選択します。
  - c. 新しいフォームの名前と簡単な説明を入力します。
  - d. フォーム グループ フィールドをクリックし、フォームに関連付けるグループを1つ選択します。
  - e. **[フォームの保存 ( Save Form ) ]** をクリックします。
  - f. **[フォーム コンテンツ ( FormContent ) ]** タブを選択し、**[ディクショナリの追加 ( Add Dictionaries ) ]** をクリックします。
  - g. **[ディクショナリの追加 ( Add Dictionaries ) ]** ダイアログボックスで、ステップ1で作成したディクショナリを検索して選択します。**[追加 ( Add ) ]** をクリックします。

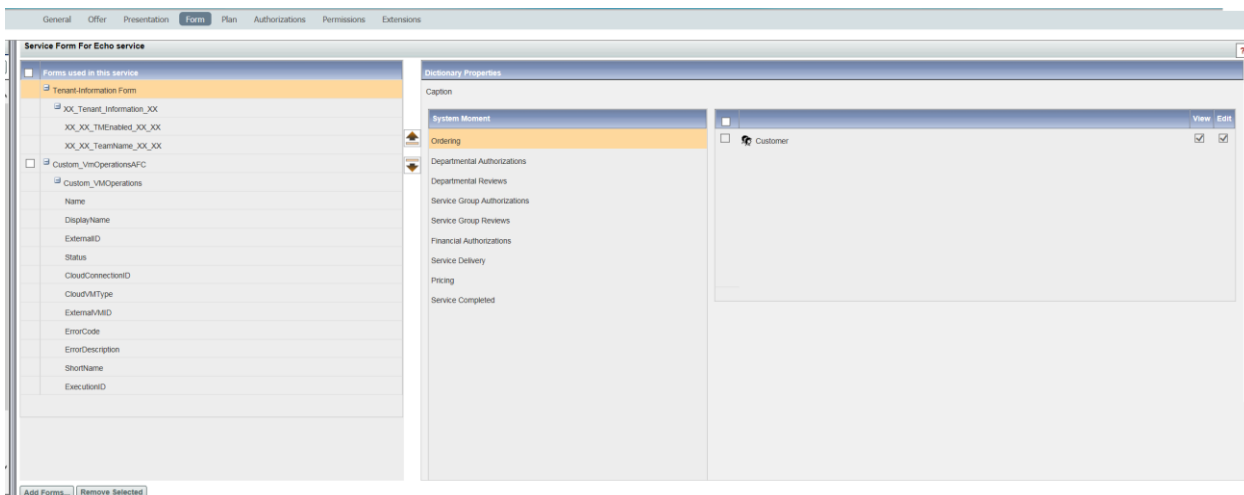
**注：** フォームを送信する前に、必ずクラウドセンターに属するVMを選択し、すべての必須ディクショナリ フィールドに入力してください。

  - h. **[フォームの保存 ( Save Form ) ]** をクリックします。

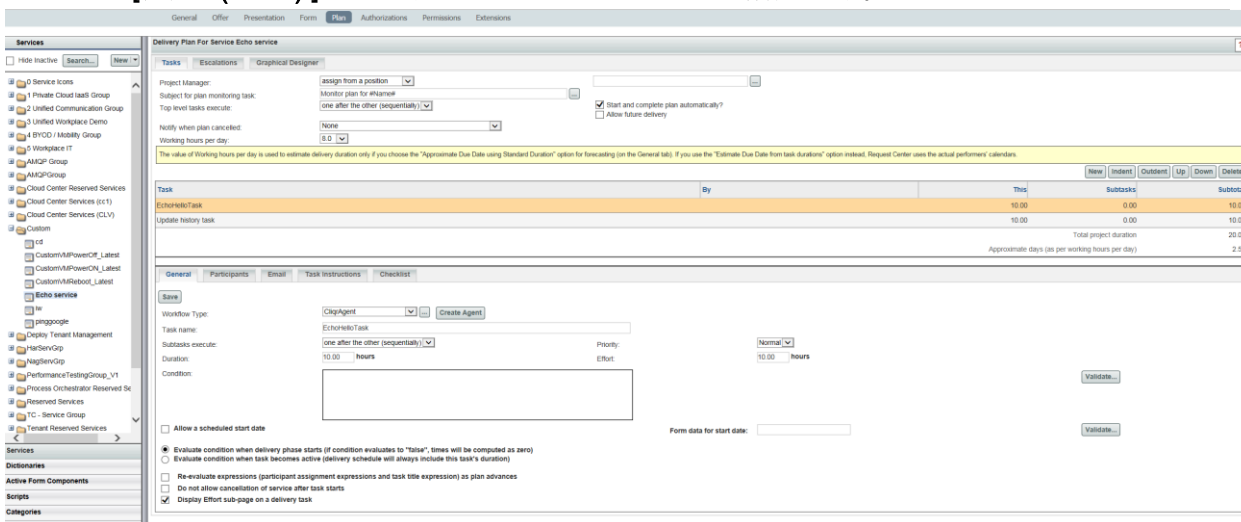
Name	Display as Grid	Show in Bundle	Show in Non-Bundle
Custom: CustomDepOperations	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DisplayName	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ShortName	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
JobID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Status	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CloudConnectionID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ErrorCode	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ErrorDescription	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3. カスタム サービス ( CustomOperationService など ) を作成し、サービスにフォームを追加します。
  - a. **[サービス デザイナー ( Service Designer ) ] > [新規 ( New ) ] > [新規サービス ( New Service ) ]** を選択します。
  - b. 提供されたフィールドに詳細を入力します。
  - c. **[Add This Service]** をクリックします。
  - d. サービスを追加した後、**[全般 ( General ) ]** タブに情報を入力することで設定を開始できます。

- e. **[保存 (Save)]** をクリックします。
- f. 作成したサービスの **[フォーム ( Form )]** タブをクリックします。
- g. ウィンドウの左下の **[フォームの追加 (Add Forms)]** をクリックします。[フォームの追加 (Add Form)] ポップアップ ウィンドウが表示されます。
- h. **[検索 (Search)]** フィールドに、ステップ 2 で作成したフォーム名を入力します。
- i. フォームを確認し、**[追加 ( Add )]** をクリックします。



4. サービスの **[プラン ( Plan )]** で下に指定されている 2 つのタスクを作成します。



VM 操作タスクの実行と操作完了タスクは必須です。更新履歴タスクはオプションです。これらの各タスクの設定の詳細については、それぞれの項を参照してください。

- [VM 操作タスクの実行](#)
- [Error! Reference source not found.](#)

5. サービスを保存します。
6. **[サービス項目マネージャ ( Service Item Manager )]** > **[サービス項目の設計 ( Design Service Items )]** > **[アプリケーション スタック ( Application Stack )]** - **[関連サービス ( Associated Services )]** でカスタム サービスを追加します。

7. [サービス項目マネージャ ( Service Item Manager ) ] > [サービス項目の設計 ( Design Service Items ) ] > [アプリケーションスタック ( Application Stack ) ] > [操作 ( Operations ) ] に移動して操作を追加し、サービスを操作に関連付けます。これにより、展開の歯車アイコンにサービスが表示されます。
8. サーバ キャッシュをクリアし、アプリケーションサーバを再起動します。  
カスタム操作が正常に作成されたことを確認するには、カスタム サービスを注文します。

#### VM 操作タスクの実行

[VM 操作の実行 ( Perform VM Operation ) ] タスクを追加するには、次の手順に従います。

前提条件 :

- CloudCenter のカスタム アクションのベース URL とペイロードを生成する変換を含む FTL ファイルを作成します。次に例を示します。

```
<!--
* FTL は、ヘッダー、Post パラメータ、URL (get) パラメータ、およびペイロードをサポートします。
*
* =====
* Header ***** Header_propertyname=PropertyValue
* Post ***** Post_propertyname=PropertyValue
* BaseUrl ***** give you the url and fallowd by query parameters
* Payload ***** Payload=payload content
* Header_AuthToken **** auth token/cookie used for authentication
* =====
-->

Protocol=https
RequestMethod=POST
CommunicationMethod=REST
AuthenticationMethod=Basic
Header_Content-Type=application/json
Header_Accept=application/json
Header_Accept-Encoding=UTF-8
BaseUrl=https://{authority}/v1/actions/34/executions
Payload={
  "resourceType": "VIRTUAL_MACHINE",
  "executionSpecs": [],
  "executionResources": [
    {
      "id": "<#list doc['message']['task-started']['requisition']['requisition-entry']['data-values']['data-value'] as
datavalue><#if (datavalue['name'])=='Custom_VMOperations.ExternalVMID'>${datavalue['value']}</#if></#list>"
    }
  ]
}
AssertResponseToken=${id}::notNullValue
ExtractResponseToken=${id}
ExtractResponseTokenError=${errors[0].message}
```

注：強調表示された「Custom\_VMOperations」は、Prime Service Catalog で作成されたディクショナリを FTL にマッピングする必要があることを示します。

- RequestCenter が展開されている PSC サーバの次の場所に FTL ファイルを追加します。  
ISEE.war/WEB-INF/classes/config/cloud
- 次の場所にある InterCloud.properties ファイルにマッピングを追加します。  
ISEE.war / WEB-INF / クラス/config/intercloud.properties

エントリは次の形式にする必要があります。

**KEY=ENCODING\_SCHEME,PATH\_TO\_FTL\_FILE\_FROM\_CONFIG\_FOLDER**

キーは CLIQR\_{FTL 拡張子のないNAME\_OF\_FTL\_FILE} である必要があります  
 例 : CLIQR\_EchoHello=json,config/cloud/Echohello.ftl

1. Go to [サービス デザイナー ( Service Designer ) ] > [サービス ( Service ) ] > [プラン ( Plan ) ] タブに移動します。ここでは、サービスはカスタム操作専用で作成された CustomOperationService です。
2. 以下に示すように、CliqrAgent タスクを使用して VM 操作タスクを実行し、省略記号内でパラメータを設定します。

以下を指定して、アウトバウンドエージェントをオーバーライドします。

パラメータ	サービス データ マッピング
CommunicationMethod	REST
Dictionary	ステップ 1 で作成したディクショナリの名前を入力します。
OperationName	カスタム動作名を入力します。CloudCenter のアクション名と一致する <CustomOperationName> を使用することをお勧めします  たとえば、CloudCenter で指定されたアクション名が <b>echohello</b> の場合、カスタム操作名は <b>echohello</b> になります。

#### 履歴の更新タスク

VM の更新履歴タスクを追加するには、次の手順に従います。

前提条件 :

- パスの SyncCustomVMExecutionOperations.properties にエントリを追加します。  
ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config\cloud
- UpdateVMHistoryOperations.properties およびSyncCustomVMExecutionOperations.properties ファイルのエントリは次の形式である必要があります。  
<Operation\_Name>=Operation\_Name

例 :

Echohello = echohello

Createdirectory = createdirectory

注 : ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config\cloud にも同じファイルが存在します。このファイルについても同じエントリを更新する必要があります。

1. Go to **[サービス デザイナー ( Service Designer ) ] > [サービス ( Service ) ] > [プラン ( Plan ) ]** タブに移動します。ここでは、サービスはカスタム操作専用で作成された CustomOperationService です。
2. 以下のように設定された SyncPlugin として、ワークフロー タイプとともにの更新履歴を作成します。

Service Catalog - Internet Explorer provided by Cisco Systems, Inc.

http://10.78.0.249:8080/RequestCenter/administration/synctasktriggerparams.do?formAct

Sync Task Plugin Class:

Param1:

Param2:

Param3:

Param4:

Param5:

OK Cancel

展開の履歴の更新タスクを追加するには、次の手順に従います。

前提条件 :

- パスの UpdateDeploymentHistoryOperations.properties にエントリを追加します。  
ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config\cloud
- エントリと UpdateDeploymentHistoryOperations.properties ファイルは次の形式である必要があります。  
<Operation\_Name>=Operation\_Name

例 :

pinggoogle=pinggoogle

invokewebservice =invokewebservice

注 : SyncCustomdeploymentExecutionOperations.properties ファイルは、ServiceLinkServer\deployments\ServiceLink.war\WEB-INF\classes\config\cloud にも存在します。このファイルについても同じエントリを更新する必要があります。

1. Go to **[サービス デザイナー ( Service Designer ) ] > [サービス ( Service ) ] > [プラン ( Plan ) ]** タブに移動します。ここでは、サービスはカスタム操作専用で作成された CustomOperationService です。
2. 以下のように設定された SyncPlugin として、ワークフロー タイプとともにの更新履歴を作成します。

Service Catalog - Internet Explorer provided by Cisco Systems, Inc.

http://10.78.0.249:8080/RequestCenter/administration/synctasktriggerparams.do?formAd

Sync Task Plugin Class:

Param1:

Param2:

Param3:

Param4:

Param5:

OK Cancel

## 同期アプリケーション VM プラグイン タスク

- Go to **[サービス デザイナー ( Service Designer ) ] > [サービス ( Service ) ] > [プラン ( Plan ) ]** タブに移動します。  
ここで、サービスは *VM でのカスタム ライフサイクル操作のサポート* セクションのステップ 4 で作成されたものです。
- 次のように設定されたカスタム SyncPlugin を作成します。  
注 : Param1 と Param2 は必須です。

Service Catalog - Internet Explorer provided by Cisco Systems, Inc.

http://10.78.0.235:8080/RequestCenter/administration/synctasktriggerparams.do?formAd

Sync Task Plugin Class:

Param1:

Param2:

Param3:

Param4:

Param5:

OK Cancel

## 同期展開プラグイン タスク

- Go to **[サービス デザイナー ( Service Designer ) ] > [サービス ( Service ) ] > [プラン ( Plan ) ]** タブに移動します。  
ここで、サービスは *VM でのカスタム ライフサイクル操作のサポート* セクションのステップ 4 で作成されたものです。

2. 次のように設定されたカスタム SyncPlugin を作成します。

注：Param1 と Param2 は必須です。



## GDPR サポート

欧州連合一般データ保護規則（GDPR）の法律に準拠し、ユーザデータを処理するために、Prime Service Catalog にはコンプライアンス モジュールが導入されています。個人を特定できるデータ（PII）は、個人を明確に特定するために使用できるあらゆるデータです。このモジュールを使用すると、管理者は GDPR サポート設定を管理して、Prime Service Catalog システムのユーザデータ保護を強化できます。

コンプライアンス モジュールは、サイト管理者が **[管理 (Administration)] > [設定 (Settings)]** の **[PII サポートの有効化 (Enable PII Support)]** を使用してのみ有効にできます。デフォルトでは、このオプションは無効になっています。

### コンプライアンス モジュールの機能

サイト管理者は、データ保護の責任を引き継ぐために任意のユーザに権限を共有できます。アクセス、読み取り専用、書き込みなどのコンプライアンス モジュールの新機能が提供されます。これらの機能を使用してカスタム ロールを作成し、任意のユーザに割り当ててコンプライアンス モジュールを維持できます。

### PII としてのユーザデータの識別

Prime Service Catalog のユーザデータは、主に個人プロフィール、サービス項目、および要求で使用されます。これらのデータは、次のモジュールから PII として識別できます。

- [サービス デザイナー ( Service Designer ) ] > [ディクショナリ ( Dictionary ) ]**  
 サービスに対する読み取り/書き込み権限を持つユーザには、ディクショナリ フィールド属性をマークするオプションが提供されます。必要なディクショナリ属性のチェックボックス「PII」を選択します。
- [サービス項目マネージャ ( Service Item Manager ) ] > [サービス項目の設計 ( Design Service Items ) ]**  
 サービス項目の読み取り/書き込み権限を持つユーザには、サービス項目属性をマークするオプションが提供されます。必要な項目属性のチェックボックス「Is PII」を選択します。
- [コンプライアンス ( Compliance ) ] > [個人 PII ( Person PII ) ]**  
 コンプライアンス モジュールに対する読み取り/書き込み権限を持つユーザには、個人プロフィール データを PII として選択するオプションが提供されます。詳細については、[グローバル個人プロフィールの PII 設定](#)を参照してください。



### グローバル個人プロフィールの PII 設定

コンプライアンスの管理者は、システム内のすべてのユーザの特定の個人プロフィール データを PII として設定できます。たとえば、管理者は、すべてのユーザの携帯電話番号、生年月日、および個人アドレスを PII として設定できます。[**コンプライアンス ( Compliance )**] > [**個人PII ( Person PII )**] に移動して、PII として設定するデータのチェックボックスをオンにし、[保存 (Save)] を選択します。

### ユーザの PII データの表示

管理者またはエンドユーザは、[PII サポート ( PII Support )] タブにアクセスして、ユーザの PII データをすべて 1 か所で表示できます。管理者として、ユーザ ピッカーを使用してユーザを検索し、[**PII データの取得 ( Get PII Data )**] をクリックすると、そのユーザにリンクされた PII のコンテキスト内のすべてのデータが下に表示されます。このページでは、それぞれの領域に次のデータがロードされます。

- 個人プロフィール：要求されたユーザまたはログインしたユーザの PII データ
- 要求：PII としてマークされたフィールドを持つディクショナリがあるユーザの要求のみ
- サービス項目：PII としてマークされたフィールドを持つユーザのサービス項目のみ

エンドユーザは、[PII サポート ( PII support )] タブへの読み取り専用アクセス権を持ち、ログイン ユーザの PII データを表示します。ただし、エンドユーザは [**データのエクスポート ( Export Data )**] オプションを使用して、PII データを含む要求およびサービス項目の .csv ファイルを生成できます。

### PII 要求およびサービス項目データのエクスポート

[PII サポート ( PII Support )] タブの [要求およびサービス項目 ( Requisitions and service Items )] 領域では、ログイン ユーザまたはサイト管理者に、外部使用のために PII データをエクスポートするオプションが提供されます。クリックすると、そのユーザの PII データを含む各サービス項目または要求の詳細を含む .csv ファイルがリストされます。

### ユーザ データの匿名化

そのようなデータの誤用を防ぐため、特定のユーザ データを匿名として設定する必要があるかどうかを判断するのは、サイト管理者の責任です。ユーザ ピッカーを使用してユーザを検索し、[**PII データの取得 ( Get PII Data )**] をクリックすると、そのユーザのすべてのユーザプロフィール データ、要求、およびサービス項目が下に表示されます。

このページ内のそれぞれの領域で、[**PII データの匿名化 ( Anonymize PII Data )**] をクリックして、任意の数のエントリを選択し、それらを匿名としてマークできます。各エリアには、PII として設定されているエントリのみを表示するための [**PII のみ ( PII Only )**] オプションもあります。[**アーカイブデータのみを表示 ( View Archived Data Only )**] チェックボックスには、アーカイブデータのみが表示されます。

### 匿名化されたデータの表示

[匿名化されたデータ ( Anonymized Data )] タブには、日付までデータが匿名化されたユーザのリストが表示されます。サイト管理者またはコンプライアンス モジュールの管理者は、このページにアクセスして、ユーザの匿名化された PII データの詳細を表示できます。

### ログイン画面でのデータ保護ポリシー URL の設定

コンプライアンス モジュールの [プロパティ ( Properties )] タブには、管理者が読み取り/書き込みアクセス権でアクセスできます。このページで、管理者はデータ保護ポリシーを含む内部 Web ページの URL を設定できます。この URL は、Prime Service Catalog のログイン画面に表示されます。

## 12.1\_Patch\_v3 で導入された API

### CloudCenter APIs

Prime Service Catalog へのアプリケーション プロファイルを更新する

メソッド：PUT

REST URL : /RequestCenter/nsapi/cloudCenter/applicationProfiles?connectionId=2

Requestbody:

```
{
  "ApplicationProfileList":
  {
```

```

    "applicationProfiles" : [
      {
        "name" : "Magento",
        "version" : "1.1.9.1"
      },
      {
        "name" : "Magento1",
        "version" : "1.1.9.2"
      }
    ]
  }
}

```

CloudCenter 接続IDを取得する

メソッド : PUT

REST URL : /RequestCenter/nsapi/cloudCenter/connectionIds

サンプル応答 :

```

{
  "ArrayList" : [
    1
  ]
}

```

VM 詳細

メソッド : GET

REST URL : RequestCenter/nsapi/cloudCenter/vm/id/{nodeID}

展開へのユーザのマッピング

メソッド : PUT

REST URL : /RequestCenter/nsapi/v2/serviceitem/cloudCenter/{connectionId}/applications?action=mapUsers

サンプル応答 :

200 OK

管理対象 VM の同期

メソッド : PUT

REST URL : /RequestCenter/nsapi/v2/serviceitem/cloudCenter/{connectionId}/virtualMachines?action=sync

サンプル応答 :

200 OK

各展開のメタデータの表示

メソッド : GET

REST URL : /RequestCenter/nsapi/serviceitem/cloudconnection/{shortName}/application/{applicationName}/metadata

例

[http://localhost:8080/RequestCenter/nsapi/serviceitem/cloudconnection/CLV/application/migrateEvening\\_from243FinalDeployment/metadata](http://localhost:8080/RequestCenter/nsapi/serviceitem/cloudconnection/CLV/application/migrateEvening_from243FinalDeployment/metadata)

サンプル応答 :

```
{
  "Map" : {
    "Name_Non_Editable_DBInfo" : "SQL Server 2016",
    "DB_MandatoryRamSize" : "32GB",
    "Install cliqr agent" : "agent installed",
    "Name_Mandatory_Filed_RAMInformation" : "10GB",
    "Cluster" : "RAC system",
    "Windows" : "2012 64 bit",
    "bit" : "32 Bit",
    "NewFiledaddedonreimport" : "New Filed added on reimport",
    "Suppoert" : "Support from the oracle",
    "Name" : "migrateEvening_from243FInalDeployment_run_1",
    "Imported VM" : "Yes depoloted form Imported",
    "Oracle" : "DB",
    "Name_JOB_Name" : "migrateEvening_from243FInalDeployment_run_1",
    "CLient" : "Intalled",
    "RAM" : "8gb"
  }
}
```

各展開のコストと時間を取得

メソッド : GET

REST URL : /RequestCenter/nsapi/serviceitem/cloudconnection/{shortName}/application/{applicationName}

例

[http://localhost:8080/RequestCenter/nsapi/serviceitem/cloudconnection/CLV/application/migrateEvening\\_from243FInalDeployment](http://localhost:8080/RequestCenter/nsapi/serviceitem/cloudconnection/CLV/application/migrateEvening_from243FInalDeployment)

サンプル応答 :

```
{
  "Map" : {
    "vmHours" : "18 Hours",
    "cost" : "19.0$"
  }
}
```

**PII API**

サービス項目消去 PII データ

サイト管理者のみがこの API にアクセスできます。

メソッド : POST

REST URL : /RequestCenter/nsapi/serviceitems/clearPII?customerId=5

```
{
  "serviceItems":[
    {
      "serviceItemTypeId":"164",
      "serviceItemIds":"1,2"
    },
    {
      "serviceItemTypeId":"162",
      "serviceItemIds":"5,6"
    }
  ]
}
```

注 : customerId が指定されている場合は、ペイロード データよりも優先されます。ペイロードでは、serviceItemTypeId と serviceItemIds の両方が必須です。

指定したユーザ ID の要求のすべての PII フィールド データを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/requisitions/PII

クエリ パラメータ :

```
gdprState= ALL,ANONYMIZED,NONANONYMIZED
customerId = 1 (mandatory)
startRow = 1 (default : 1)
recordSize = 1 (default : 20)
PIIOnly = true/false (default : false)
sortOrder = ASC/DESC (default : DESC)
sortBy = RequisitionID, CreatedOn (default : RequisitionID)
isArchived = true/false (default - false)
```

応答 :

```
{
  "startRow" : 1,
  "recordSize" : 1,
  "totalRecords" : 1,
  "requisitions" : [
```

```

{
  "requisitionId" : 33,
  "createdOn" : "Wed May 02 06:47:40 IST 2018",
  "customerId" : 5,
  "archived" : false,
  "anonymizedOn" : "Wed May 02 06:49:50 IST 2018",
  "requisitionEntries" : [
    {
      "requisitionEntryId" : 42,
      "serviceId" : 141,
      "serviceName" : "CustomerInformation_Delete",
      "formFields" : [
        {
          "fieldName" : "Customer_Title",
          "dictionaryName" : "CustomerDeltelInformation",
          "dictionaryCaption" : "CustomerDeletelInformationCaption",
          "isPII" : true,
          "fieldValue" : "aa 02-05-2018 08:50:33.927"
        },
        {
          "fieldName" : "Customer_Birthdate",
          "dictionaryName" : "CustomerDeltelInformation",
          "dictionaryCaption" : "CustomerDeletelInformationCaption",
          "isPII" : true,
          "fieldValue" : "aa 02-05-2018 08:50:33.928"
        }
      ]
    }
  ]
}

```

指定したユーザ ID のサービス項目のすべての PII フィールド データを取得する  
メソッド : GET

REST URL : RequestCenter/nsapi/serviceitems/PII

クエリ パラメータ : PIIOnly=true/false

customerID = <specific\_customer\_id>

応答 :

```
{
  "AllServiceItems" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 1,
    "serviceitemsubscriptions" : [
      {
        "id" : 416,
        "serviceitemClassificationName" : "Hardware",
        "organizationalUnitID" : 22,
        "assignedDate" : "05/23/2018",
        "requisitionID" : 0,
        "submittedDate" : "05/23/2018",
        "submittedDateRaw" : 1527034608000,
        "anonymizedOn" : "05/23/2018",
        "anonymizedOnRaw" : 1527056601000,
        "customerID" : 6,
        "requisitionEntryID" : 0,
        "serviceitemClassificationID" : 23,
        "organizationalUnitName" : "ShashiOU",
        "customerName" : "shashi215 user",
        "accountID" : 0,
        "accountName" : " ",
        "agreementID" : 0,
        "agreementName" : " ",
        "serviceitem" : {
          "id" : 162,
          "name" : "UserServiceitem",
          "logicName" : "SiUserServiceitem",
          "shareable" : true,
          "serviceitemData" : [
            {
              "rowId" : 4,
              "serviceitemURL" : null,
              "myservicesServiceitemURL" : null,
              "serviceitemURLOnly" : null,
              "protectedAttributes" : [
```

```

        "Str32",
        "Str512"
    ],
    "items" : [
        {
            "Name" : "Test4",
            "Str32" : "ss 2018-05-23",
            "Str512" : "ss 2018-05-23 11:53:21"
        }
    ],
    "subscription" : {}
}
]
},
"displayName" : "Test4"
}
}

```

顧客の PII 要求データをエクスポートする

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/requisitions/exportPII

クエリ パラメータ :

customerId = 1 (mandatory)

PIIOnly = true/false

応答 :

すべての顧客データを含む csv 形式の Excel ファイルを生成します。

顧客の PII サービス項目データをエクスポートする

メソッド : GET

REST URL : /RequestCenter/nsapi/serviceitems/exportPII?CustomerID=<specific\_customer\_id>

応答 :

すべての顧客データを含む csv 形式の Excel ファイルを生成します。

指定したユーザの要求を匿名化する

メソッド : POST

REST URL : /RequestCenter/nsapi/transaction/requisitions/clearUserData

クエリ パラメータ :

customerId=1

replaceWithString=gdpr\_safe

サンプル リクエスト :

要求データ : オプション

```

{
  "requisitionIds": "1,2,3,4"
}

```

}

注：顧客 ID は、要求 ID よりも優先されます。したがって、クエリ パラメータで顧客 ID がパスされた場合、要求本文からの `requisitionsIds` 値は無視されます。

個人フィールドを PII として設定する

メソッド：PUT

REST URL：/RequestCenter/nsapi/directory/people/gdprConfiguration?responseType=json

サンプル ペイロード：

```
{
  "persongdprconfigurations" : {
    "totalCount" : 0,
    "persongdprconfiguration" : [
      {
        "fieldName" : "FIRSTNAME",
        "entityTypeID" : 5,
        "entityTypeName" : "person",
        "entityTypeLocalizedName" : "Person",
        "protectedField" : true,
        "configurationID" : 1,
        "displayName" : "First Name"
      },
      {
        "fieldName" : "LASTNAME",
        "entityTypeID" : 5,
        "entityTypeName" : "person",
        "entityTypeLocalizedName" : "Person",
        "protectedField" : true,
        "configurationID" : 2,
        "displayName" : "Last Name"
      }
    ]
  }
}
```

個人の PII フィールドを取得する

メソッド：GET

REST URL：/RequestCenter/nsapi/directory/people/gdprConfiguration?responseType=json

サンプル応答：



```
{
  "persongdprconfigurations" : {
    "totalCount" : 0,
    "persongdprconfiguration" : [
      {
        "fieldName" : "FIRSTNAME",
        "entityTypeID" : 5,
        "entityTypeName" : "person",
        "entityTypeLocalizedName" : "Person",
        "protectedField" : true,
        "configurationID" : 1,
        "displayName" : "First Name"
      },
      {
        "fieldName" : "LASTNAME",
        "entityTypeID" : 5,
        "entityTypeName" : "person",
        "entityTypeLocalizedName" : "Person",
        "protectedField" : true,
        "configurationID" : 2,
        "displayName" : "Last Name"
      },
      {
        "fieldName" : "BIRTHDATE",
        "entityTypeID" : 5,
        "entityTypeName" : "person",
        "entityTypeLocalizedName" : "Person",
        "protectedField" : false,
        "configurationID" : 3,
        "displayName" : "Birth Date"
      }
    ]
  }
}
```

コンプライアンス モジュールを設定する

メソッド : GET

REST URL : /RequestCenter/nsapi/v1/common/moduleconfiguration/compliance

サンプル応答 :

```
{
  "profileURL":
  ".././../myservices/navigate.do?query=personprofile&id=10&moduleId=49&personProfileModuleId=5&force=true&layout=popup&utid=70BA0BAA5AE8B7683ED76EF94BDB2D80",
  "portalNSApiPageSizeForDirectory": 20,
  "portal_refactor_menu": null,
  "isNextGenUser": null,
  "accessAuthorization": true,
  "accessProfile": true,
  "personInfo": {
    "firstName": "test1",
    "lastName": "test1",
    "timeZoneId": 256,
    "timeZoneDisplayName": "(GMT-08:00) Pacific Time (US and Canada), Tijuana",
    "languageId": 0,
    "placeId": 1,
    "homeOrganizationalUnitId": 24,
    "goals": "",
    "login": "test1",
    "localeId": 1,
    "status": 0,
    "managerId": 0,
    "languageName": "<s ID=\"153\"/>",
    "id": 10,
    "recordStateId": 1,
    "tenantId": 1,
    "dailyLoad": 0,
    "supervisorId": 0,
    "isActive": true,
    "count": 0,
    "queue": false,
    "shouldCreatePerson": true,
    "cloudPassword": "admin",
    "guid": "26F6B280-1EB5-4120-854D-39F8F57A2042",
    "isInactive": false,
    "isLocked": 0,
    "dirNetworkInfo": {
      "loginname": "test1",
      "personID": 0,
      "networkInfold": 0,
      "isLocked": 0,
      "login": "test1"
    },
    "passwordChange": false,
    "loginInfoDTO": {
      "loginname": "test1",
      "personID": 0,
      "networkInfold": 0,
      "isLocked": 0,
      "login": "test1"
    },
    "personalAddressDTO": {
      "addressTypeId": 0,
      "objectId": 0,

```

```
"objectInstanceId": 0,
  "id": 0,
  "tenantId": 0,
  "personId": 0
},
"businessAddressDTO": {
  "addressTypeId": 0,
  "objectId": 0,
  "objectInstanceId": 0,
  "id": 0,
  "tenantId": 0,
  "personId": 0
},
"homeOUID": 24,
"personalAddressDTO": {
  "addressTypeId": 0,
  "objectId": 0,
  "objectInstanceId": 0,
  "id": 0,
  "tenantId": 0,
  "personId": 0
},
"businessAddressDTO": {
  "addressTypeId": 0,
  "objectId": 0,
  "objectInstanceId": 0,
  "id": 0,
  "tenantId": 0,
  "personId": 0
},
"birthDateAsString": "",
"hireDateAsString": "",
"idEncrypted": "YjkNQUU2GzrZyirVnzLQyw%3D%3D",
"loginInfoDTO": {
  "loginname": "test1",
  "personID": 0,
  "networkInfold": 0,
  "isLocked": 0,
  "login": "test1"
}
},
"accessMyStuff": true,
"accessIntegration": false,
"showTeams": false,
"hasComplianceAccess": false,
"refactor_menu": null,
"enableHeaderFooter": false,
"moduleId": "28",
"accessMyVDC": false,
"hasComplianceReadCapability": false,
"accessVDCReport": false,
"viewRequisitions": true,
"isComplianceWebsiteOn": false,
"accessMyServers": false,
"EUIOBSearch": false,
"shortDateFormat": "MM/dd/yyyy",
"issiteadmin": false,
"enableCloudCenter": true,
"debugJ2EE": false,
"enableCustomCSS": true,
"sessionToken": "70BA0BAA5AE8B7683ED76EF94BDB2D80",
"accessMyApplications": false,
```

```
"hasComplianceWriteCapability": false
}
```

データが匿名化されるユーザのリストを取得する  
方法 : GET

REST URL : /RequestCenter/nsapi/directory/people/anonymized

応答 :

```
{
  "userList" : {
    "startRow" : 1,
    "totalCount" : 3,
    "recordSize" : 3,
    "user" : [
      {
        "personId" : 5,
        "login" : "ramesh2"
      },
      {
        "personId" : 8,
        "login" : "user1"
      },
      {
        "personId" : 9,
        "login" : "user2"
      }
    ]
  }
}
```

ユーザを匿名化する  
メソッド : PUT

REST URL : /RequestCenter/nsapi/directory/people/anonymized?responseType=json&PersonId=1  
サンプルペイロード

```
{
  "status-message" : {
    "code" : "Success",
```

```
"value" : "Updated Successfully"  
}  
}
```

匿名化されたデータのリストを取得する  
メソッド : GET

REST URL : /RequestCenter/nsapi/directory/people/id/{personID}/anonymizedData?responseType=json&PIIOnly=false

サンプル応答 :

```
{  
  "persongdprconfigurations": {  
    "totalCount": 30,  
    "anonymizedOnStr": "",  
    "persongdprconfiguration": [  
      {  
        "fieldName": "FIRSTNAME",  
        "fieldValue": "cctest111",  
        "entityTypeID": 5,  
        "entityTypeName": "person",  
        "entityTypeLocalizedName": "Person",  
        "protectedField": true,  
        "configurationID": 31,  
        "displayName": "First Name"  
      },  
      {  
        "fieldName": "LASTNAME",  
        "fieldValue": "cctest111",  
        "entityTypeID": 5,  
        "entityTypeName": "person",  
        "entityTypeLocalizedName": "Person",  
        "protectedField": true,  
        "configurationID": 32,  
        "displayName": "Last Name"  
      },  
      {  
        "fieldName": "BIRTHDATE",  
        "entityTypeID": 5,  
        "entityTypeName": "person",  
        "entityTypeLocalizedName": "Person",
```

```
"protectedField": true,
"configurationID": 33,
"displayName": "Birth Date"
},
{
"fieldName": "SSN",
"entityTypeID": 5,
"entityTypeName": "person",
"entityTypeLocalizedName": "Person",
"protectedField": true,
"configurationID": 34,
"displayName": "SSN"
},
{
"fieldName": "STREET1",
"entityTypeID": 2,
"entityTypeName": "personaladdress",
"entityTypeLocalizedName": "Personal Address",
"protectedField": true,
"configurationID": 35,
"displayName": "Street 1"
},
{
"fieldName": "STREET2",
"entityTypeID": 2,
"entityTypeName": "personaladdress",
"entityTypeLocalizedName": "Personal Address",
"protectedField": true,
"configurationID": 36,
"displayName": "Street 2"
},
{
"fieldName": "CITY",
"entityTypeID": 2,
"entityTypeName": "personaladdress",
"entityTypeLocalizedName": "Personal Address",
"protectedField": true,
"configurationID": 37,
"displayName": "City"
```

```
},
{
  "fieldName": "STATEPROVINCE",
  "entityTypeID": 2,
  "entityTypeName": "personaladdress",
  "entityTypeLocalizedName": "Personal Address",
  "protectedField": true,
  "configurationID": 38,
  "displayName": "State or Province"
},
{
  "fieldName": "ZIP",
  "entityTypeID": 2,
  "entityTypeName": "personaladdress",
  "entityTypeLocalizedName": "Personal Address",
  "protectedField": true,
  "configurationID": 39,
  "displayName": "ZIP"
},
{
  "fieldName": "COUNTRY",
  "entityTypeID": 2,
  "entityTypeName": "personaladdress",
  "entityTypeLocalizedName": "Personal Address",
  "protectedField": true,
  "configurationID": 40,
  "displayName": "Country"
},
{
  "fieldName": "EMAIL",
  "fieldValue": "cctest111@cisco.com",
  "entityTypeID": 6,
  "entityTypeName": "email",
  "entityTypeLocalizedName": "Email",
  "protectedField": true,
  "configurationID": 41,
  "displayName": "Email"
},
{
```

```
"fieldName": "WORKPHONE",
"entityTypeID": 7,
"entityTypeName": "workphone",
"entityTypeLocalizedName": "Work Phone",
"protectedField": true,
"configurationID": 42,
"displayName": "Work Phone"
},
{
"fieldName": "HOMEPHONE",
"entityTypeID": 8,
"entityTypeName": "homephone",
"entityTypeLocalizedName": "Home phone",
"protectedField": true,
"configurationID": 43,
"displayName": "Home phone"
},
{
"fieldName": "FAX",
"entityTypeID": 9,
"entityTypeName": "fax",
"entityTypeLocalizedName": "Fax",
"protectedField": false,
"configurationID": 44,
"displayName": "Fax"
},
{
"fieldName": "MOBILEPHONE",
"entityTypeID": 10,
"entityTypeName": "mobilephone",
"entityTypeLocalizedName": "Mobile phone",
"protectedField": true,
"configurationID": 45,
"displayName": "Mobile phone"
},
{
"fieldName": "PAGER",
"entityTypeID": 11,
"entityTypeName": "pager",
```



```
"entityTypeLocalizedName": "Pager",
"protectedField": false,
"configurationID": 46,
"displayName": "Pager"
},
{
"fieldName": "OTHER",
"entityTypeID": 12,
"entityTypeName": "other",
"entityTypeLocalizedName": "Other",
"protectedField": false,
"configurationID": 47,
"displayName": "Other"
},
{
"fieldName": "MAINPHONE",
"entityTypeID": 13,
"entityTypeName": "mainphone",
"entityTypeLocalizedName": "Main Phone",
"protectedField": true,
"configurationID": 48,
"displayName": "Main Phone"
},
{
"fieldName": "PRIMARYPHONE",
"entityTypeID": 14,
"entityTypeName": "primaryphone",
"entityTypeLocalizedName": "Primary Phone",
"protectedField": true,
"configurationID": 49,
"displayName": "Primary Phone"
},
{
"fieldName": "PRIMARYFAX",
"entityTypeID": 15,
"entityTypeName": "primaryfax",
"entityTypeLocalizedName": "Primary Fax",
"protectedField": true,
"configurationID": 50,
```

```
"displayName": "Primary Fax"
},
{
  "fieldName": "SALESPHONE",
  "entityTypeID": 16,
  "entityTypeName": "salesphone",
  "entityTypeLocalizedName": "Sales Phone",
  "protectedField": false,
  "configurationID": 51,
  "displayName": "Sales Phone"
},
{
  "fieldName": "SUPPORTPHONE",
  "entityTypeID": 17,
  "entityTypeName": "supportphone",
  "entityTypeLocalizedName": "Support Phone",
  "protectedField": false,
  "configurationID": 52,
  "displayName": "Support Phone"
},
{
  "fieldName": "BILLINGPHONE",
  "entityTypeID": 18,
  "entityTypeName": "billingphone",
  "entityTypeLocalizedName": "Billing Phone",
  "protectedField": false,
  "configurationID": 53,
  "displayName": "Billing Phone"
},
{
  "fieldName": "OTHERCONTACT",
  "entityTypeID": 19,
  "entityTypeName": "othercontact",
  "entityTypeLocalizedName": "Other Contact",
  "protectedField": false,
  "configurationID": 54,
  "displayName": "Other Contact"
},
{
```

```

    "fieldName": "STREET1",
    "entityTypeID": 1,
    "entityTypeName": "companyaddress",
    "entityTypeLocalizedName": "Company Address",
    "protectedField": false,
    "configurationID": 55,
    "displayName": "Street 1"
  },
  {
    "fieldName": "STREET2",
    "entityTypeID": 1,
    "entityTypeName": "companyaddress",
    "entityTypeLocalizedName": "Company Address",
    "protectedField": false,
    "configurationID": 56,
    "displayName": "Street 2"
  },

```

### その他 API

要求の添付ファイルを取得する

メソッド: GET

REST URL: /RequestCenter/nsapi/transaction/v2/requisition/{requisitionID}/attachment

サンプル応答:

```

{
  "ArrayList" : [
    {
      "startRow" : 0,
      "totalCount" : 2,
      "recordSize" : 2,
      "requisitionId" : 31,
      "document" : [
        {
          "documentID" : 4,
          "objectID" : 29,
          "objectInstID" : 31,
          "fileDisplayName" : "Mac Setup Assistant Summary.pdf",
          "docSize" : 44789,
          "lastUploadDate" : "05/18/2018 8:38 AM"
        }
      ]
    }
  ]
}

```

```

    },
    {
        "documentID" : 3,
        "objectID" : 29,
        "objectInstID" : 31,
        "fileDisplayName" : "Screen Shot 2018-05-04 at 2.57.58 PM.png",
        "docSize" : 58862,
        "lastUploadDate" : "05/18/2018 8:38 AM"
    }
]
}
]
}

```

添付ファイルを削除する

メソッド : DELETE

REST URL : /RequestCenter/nsapi/transaction/requisitions/id/{requisitionID}/attachments?documentIds=6

応答 :

<nsapi-response>添付ファイルが正常に削除されました。</nsapi-response>

ドキュメント ID の添付ファイルをダウンロードする

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/requisition/attachment/{requisitionID}

要求と承認待ちのステータスを取得する

メソッド : GET

REST URL : /RequestCenter/nsapi/transaction/v2/requisition/{requisitionIDs}/status

サンプル応答 :

```

{
    "ArrayList" : [
        {
            "status" : {
                "requisitionId" : 13,
                "statusId" : 2,
                "statusName" : "Closed"
            }
        },
        {
            "status" : {
                "requisitionId" : 12,

```

```

        "statusId" : 4,
        "statusName" : "Rejected"
    }
},
{
    "status" : {
        "requisitionId" : 10,
        "statusId" : 5,
        "statusName" : "Delivery Cancelled"
    }
}
]
}

```

要求のコメント (ユーザ コメントとシステム コメントの両方) を取得する  
メソッド: GET

REST URL : /RequestCenter/nsapi/transaction/v2/requisition/comments

サンプル応答:

```

{
    "entities" : {
        "totalCount" : 1,
        "hasMore" : false,
        "items" : [
            {
                "startRow" : 0,
                "totalCount" : 0,
                "recordSize" : 0,
                "id" : 14,
                "usercomments" : [
                    {
                        "commentedOn" : "05/16/2018 2:20 PM",
                        "commentText" : "See my comments",
                        "originatorName" : "Dhananjya k",
                        "commentsID" : 36
                    }
                ],
                "systemcomments" : [
                    {
                        "commentedOn" : "05/16/2018 10:37 AM",

```

```

        "commentText" : "admin admin has completed task [Monitor plan for
Apple Macbook] in requisition number : 14.",
        "originatorName" : "admin admin",
        "commentsID" : 27
    },
    {
        "commentedOn" : "05/16/2018 10:36 AM",
        "commentText" : "In Requisition 14, activity Second Level Auth, for
Service Apple Macbook, admin admin performed [Approve].",
        "originatorName" : "admin admin",
        "commentsID" : 25
    },
    {
        "commentedOn" : "05/16/2018 10:36 AM",
        "commentText" : "In Requisition 14, activity First Level Auth, for
Service Apple Macbook, admin admin performed [Approve].",
        "originatorName" : "admin admin",
        "commentsID" : 24
    },
    {
        "commentedOn" : "05/16/2018 10:36 AM",
        "commentText" : "In Requisition 14, activity Review1, for Service Apple
Macbook, admin admin performed [OK].",
        "originatorName" : "admin admin",
        "commentsID" : 23
    },
    {
        "commentedOn" : "05/16/2018 10:35 AM",
        "commentText" : " has updated service form details of requisition
number : 14.",
        "originatorName" : "admin admin",
        "commentsID" : 21
    }
}
]
}
]
}
}

```

サービス項目一括操作（作成、更新、削除）

注：特定のタイプのサービス項目は、一括操作でサポートされません。

メソッド : PUT

REST URL : /RequestCenter/nsapi/serviceitem/v2/process?isBatch=true&commitType=2&batchSize=50)

クエリ パラメータ :

- `isBatch=true/false` は、一括操作が使用されることを示します。
- `commitType = 1/2`
  - 1は、コミットタイプがすべてであることを意味します。つまり、いずれか1つのサービス項目が更新される際に問題がある場合、どのサービス項目も更新されません。
  - 2はコミットタイプが部分的であることを意味します。つまり、1つ以上のサービス項目に問題がある場合でも、他のサービス項目は更新されます。
- `batchSize =<numeric value>` は、一括操作でのバッチ サイズを示します。たとえば、一括更新のために 1000 個のサービス項目が送信され、`batchSize` が 50 に設定されている場合、20個のバッチがデータベースで実行されます。

**注** : `isBatch` が `false` に設定されている場合は、操作が非バッチ方式で実行され、上記の他のクエリ パラメータも適用されません。

サンプル ペイロード :

```
<serviceitem>
<name>CustomSIType02</name>
<serviceItemData>
  <serviceItemAttribute name="Name">D_1</serviceItemAttribute>
  <serviceItemAttribute name="Age">12</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">245</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max11</serviceItemAttribute>
  <subscription>
    <loginID>dinesh01</loginID>
    <ouname>CustomOU1</ouname>
    <accountName>CustomTenant</accountName>
    <requisitionEntryID>1</requisitionEntryID>
  </subscription>
</serviceItemData>
<serviceItemData>
  <serviceItemAttribute name="Name">D_2</serviceItemAttribute>
  <serviceItemAttribute name="Age">12</serviceItemAttribute>
  <serviceItemAttribute name="Rupees">245</serviceItemAttribute>
  <serviceItemAttribute name="DLongInte">2453232</serviceItemAttribute>
  <serviceItemAttribute name="DDoubleFloat">2453232.11</serviceItemAttribute>
  <serviceItemAttribute name="DDateTime">2018-01-01 11:30</serviceItemAttribute>
  <serviceItemAttribute name="Dstringmax">I am string max12</serviceItemAttribute>
  <subscription>
    <loginID>dinesh01</loginID>
    <ouname> CustomOU1</ouname>
```

```

<accountName> CustomTenant </accountName>
<requisitionEntryID>1</requisitionEntryID>
</subscription>
</serviceItemData>
<serviceitem>

```

MySelf および Ordered-For-Other で注文されたすべての要求を取得する

メソッド : Get

REST URL : /RequestCenter/nsapi/transaction/requisitions/ViewName=Orders with my involvement

サンプル応答 :

```

{
  "requisitions" : {
    "startRow" : 1,
    "totalCount" : 15,
    "recordSize" : 15,
    "requisition" : [
      {
        "tenantId" : 1,
        "userId" : 30,
        "ownerId" : 30,
        "serviceId" : 128,
        "customerId" : 30,
        "expectedDuration" : 0.0,
        "actualDuration" : 0.0,
        "startedDate" : "05/17/2018 10:56 AM",
        "closedDate" : "05/17/2018 10:56 AM",
        "dueDate" : "05/18/2018 6:00 PM",
        "expectedCost" : 0.0,
        "status" : "Closed",
        "requisitionId" : 60,
        "lateFlag" : false,
        "customerName" : "OOBUser2 User2",
        "organizationalUnitName" : "OOB_OU2",
        "submitDate" : "05/17/2018 10:56 AM",
        "statusId" : 2,
        "serviceName" : "Create Team",

```



```
    "ownerName" : "OOBUser2 User2",
    "organizationalUnitId" : 53,
    "startedDateRaw" : 1526534789093,
    "closedDateRaw" : 1526534791630,
    "dueDateRaw" : 1526646600000,
    "submitDateRaw" : 1526534789333,
    "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=60&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">60</a>",
    "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=60",
    "milestoneLink" : " ",
    "percentageCompleted" : 0.0,
    "customerEmail" : "oobuser2@cisco.com",
    "isCancelable" : true,
    "rating" : "0",
    "reviewsCount" : 0,
    "selfRating" : 0,
    "isServiceOrderable" : 1,
    "isServiceActive" : 1,
    "teamId" : 0
  },
  {
    "tenantId" : 1,
    "userId" : 30,
    "ownerId" : 30,
    "serviceId" : 174,
    "customerId" : 30,
    "expectedDuration" : 0.0,
    "actualDuration" : 0.0,
    "startedDate" : "05/17/2018 10:57 AM",
    "dueDate" : "05/17/2018 4:00 PM",
    "expectedCost" : 0.0,
    "status" : "Ongoing",
    "requisitionId" : 62,
    "flagImage" : "/RequestCenter/images/flaglate.gif",
    "lateFlag" : true,
    "customerName" : "OOBUser2 User2",
    "organizationalUnitName" : "TeamB",
```

```
"submitDate" : "05/17/2018 10:57 AM",
"statusId" : 1,
"serviceName" : "OOB_Service",
"ownerName" : "OOBUser2 User2",
"organizationalUnitId" : 60,
"startedDateRaw" : 1526534825223,
"dueDateRaw" : 1526553000000,
"submitDateRaw" : 1526534831743,
"requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=62&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">62</a>",
"requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=62",
"milestoneLink" : " ",
"percentageCompleted" : 0.0,
"customerEmail" : "oobuser2@cisco.com",
"isCancelable" : true,
"rating" : "0",
"reviewsCount" : 0,
"selfRating" : 0,
"isServiceOrderable" : 1,
"isServiceActive" : 1,
"teamId" : 0
},
{
"tenantId" : 1,
"userId" : 30,
"ownerId" : 30,
"serviceId" : 174,
"customerId" : 29,
"expectedDuration" : 0.0,
"actualDuration" : 0.0,
"startedDate" : "05/22/2018 6:30 AM",
"dueDate" : "05/22/2018 4:00 PM",
"expectedCost" : 0.0,
"status" : "Ongoing",
"requisitionId" : 95,
"flagImage" : "/RequestCenter/images/flaglate.gif",
```

```

    "lateFlag" : true,
    "customerName" : "OOBUser1 User1",
    "organizationalUnitName" : "TeamA",
    "submitDate" : "05/22/2018 6:30 AM",
    "statusId" : 1,
    "serviceName" : "OOB_Service",
    "ownerName" : "OOBUser2 User2",
    "organizationalUnitId" : 58,
    "startedDateRaw" : 1526950820487,
    "dueDateRaw" : 1526985000000,
    "submitDateRaw" : 1526950828123,
    "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=95&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">95</a>",
    "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=95",
    "milestoneLink" : " ",
    "percentageCompleted" : 0.0,
    "customerEmail" : "oobuser1@cisco.com",
    "isCancelable" : true,
    "rating" : "0",
    "reviewsCount" : 0,
    "selfRating" : 0,
    "isServiceOrderable" : 1,
    "isServiceActive" : 1,
    "teamId" : 0
  },
  {
    "tenantId" : 1,
    "userId" : 30,
    "ownerId" : 30,
    "serviceId" : 145,
    "serviceImage" :
"/RequestCenter/nsapi/definition/images?tenantId=1&type=def&documentId=371",
    "customerId" : 30,
    "expectedDuration" : 36.0,
    "actualDuration" : 0.0,
    "startedDate" : "05/22/2018 8:54 AM",
    "closedDate" : "05/22/2018 9:03 AM",

```

```

    "dueDate" : "05/25/2018 10:00 PM",
    "expectedCost" : 0.0,
    "status" : "Closed",
    "requisitionId" : 99,
    "lateFlag" : false,
    "customerName" : "OOBUser2 User2",
    "organizationalUnitName" : "OOB_OU2",
    "submitDate" : "05/22/2018 8:54 AM",
    "statusId" : 2,
    "serviceName" : "CentOS 6.x (v2.0) (CC1)",
    "ownerName" : "OOBUser2 User2",
    "organizationalUnitId" : 53,
    "startedDateRaw" : 1526959443110,
    "closedDateRaw" : 1526959988160,
    "dueDateRaw" : 1527265800000,
    "submitDateRaw" : 1526959443350,
    "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=99&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">99</a>",
    "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=99",
    "milestoneLink" : " ",
    "percentageCompleted" : 0.0,
    "customerEmail" : "oobuser2@cisco.com",
    "isCancelable" : true,
    "rating" : "0",
    "reviewsCount" : 0,
    "selfRating" : 0,
    "isServiceOrderable" : 1,
    "isServiceActive" : 1,
    "teamId" : 0
  },
  {
    "tenantId" : 1,
    "userId" : 30,
    "ownerId" : 30,
    "serviceId" : 9,
    "serviceImage" :
"/RequestCenter/nsapi/definition/images?tenantId=1&type=def&documentId=11",

```

```

"customerId" : 30,
"expectedDuration" : 10.0,
"actualDuration" : 0.0,
"startedDate" : "05/25/2018 8:32 AM",
"closedDate" : "05/25/2018 8:33 AM",
"dueDate" : "05/28/2018 6:00 PM",
"expectedCost" : 999.95,
"status" : "Closed",
"requisitionId" : 115,
"lateFlag" : false,
"customerName" : "OOBUser2 User2",
"organizationalUnitName" : "OOB_OU2",
"submitDate" : "05/25/2018 8:32 AM",
"statusId" : 2,
"serviceName" : "Apple Macbook",
"ownerName" : "OOBUser2 User2",
"organizationalUnitId" : 53,
"startedDateRaw" : 1527217376520,
"closedDateRaw" : 1527217380063,
"dueDateRaw" : 1527510600000,
"submitDateRaw" : 1527217379393,
"requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=115&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">115</a>",
"requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=115",
"milestoneLink" : " ",
"percentageCompleted" : 0.0,
"customerEmail" : "oobuser2@cisco.com",
"isCancelable" : true,
"rating" : "0",
"reviewsCount" : 0,
"selfRating" : 0,
"isServiceOrderable" : 1,
"isServiceActive" : 1,
"teamId" : 0
},
{
"tenantId" : 1,

```

```
"userId" : 30,
"ownerId" : 30,
"serviceId" : 9,
"serviceImage" :
"/RequestCenter/nsapi/definition/images?tenantId=1&type=def&documentId=11",
"customerId" : 29,
"expectedDuration" : 10.0,
"actualDuration" : 0.0,
"startedDate" : "05/25/2018 8:33 AM",
"closedDate" : "05/25/2018 8:33 AM",
"dueDate" : "05/28/2018 6:00 PM",
"expectedCost" : 999.95,
"status" : "Closed",
"requisitionId" : 116,
"lateFlag" : false,
"customerName" : "OOBUser1 User1",
"organizationalUnitName" : "OOB_OU1",
"submitDate" : "05/25/2018 8:33 AM",
"statusId" : 2,
"serviceName" : "Apple Macbook",
"ownerName" : "OOBUser2 User2",
"organizationalUnitId" : 52,
"startedDateRaw" : 1527217395690,
"closedDateRaw" : 1527217407687,
"dueDateRaw" : 1527510600000,
"submitDateRaw" : 1527217407360,
"requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=116&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">116</a>",
"requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=116",
"milestoneLink" : " ",
"percentageCompleted" : 0.0,
"customerEmail" : "oobuser1@cisco.com",
"isCancelable" : true,
"rating" : "0",
"reviewsCount" : 0,
"selfRating" : 0,
"isServiceOrderable" : 1,
```

```

        "isServiceActive" : 1,
        "teamId" : 0
    },
    {
        "tenantId" : 1,
        "userId" : 30,
        "ownerId" : 30,
        "serviceId" : 175,
        "customerId" : 30,
        "expectedDuration" : 0.0,
        "actualDuration" : 0.0,
        "startDate" : "05/25/2018 8:34 AM",
        "dueDate" : "05/25/2018 4:00 PM",
        "expectedCost" : 0.0,
        "status" : "Ongoing",
        "requisitionId" : 117,
        "flagImage" : "/RequestCenter/images/flaglate.gif",
        "lateFlag" : true,
        "customerName" : "OOBUser2 User2",
        "organizationalUnitName" : "TeamB",
        "submitDate" : "05/25/2018 8:34 AM",
        "statusId" : 1,
        "serviceName" : "Copy of OOB_Service",
        "ownerName" : "OOBUser2 User2",
        "organizationalUnitId" : 60,
        "startDateRaw" : 1527217467360,
        "dueDateRaw" : 1527244200000,
        "submitDateRaw" : 1527217469977,
        "requisitionURL" : "<a
href=/RequestCenter/myservices/navigate.do?reqid=117&layout=popup_p&query=requisitionstatus' onclick=\"return
GB_showFullScreen('Requisition', this.href)\">117</a>",
        "requisitionURLOnly" : "/RequestCenter/myservices/navigate.do?reqid=117",
        "milestoneLink" : " ",
        "percentageCompleted" : 0.0,
        "customerEmail" : "oobuser2@cisco.com",
        "isCancelable" : true,
        "rating" : "0",
        "reviewsCount" : 0,

```

```

        "selfRating" : 0,
        "isServiceOrderable" : 1,
        "isServiceActive" : 1,
        "teamId" : 0
    }
}

```

サービス定義を取得する  
メソッド : GET

REST URL : /RequestCenter/nsapi/definition/servicedefs

クエリ パラメータ : servicelds

応答 :

```

{
  "services" : {
    "startRow" : 1,
    "totalCount" : 1,
    "recordSize" : 1,
    "service" : [
      {
        "serviceld" : 149,
        "serviceName" : "Cassandra Cluster App (v2.1.1) (CLQ)",
        "description" : "Cassandra Cluster Admin is a GUI tool to help people administrate their
Apache Cassandra cluster.",
        "topDescription" : "",
        "middleDescription" : "",
        "bottomDescription" : "",
        "revisionNumber" : 293,
        "status" : "Active",
        "statusId" : 1,
        "expectedDuration" : 0.0,
        "expectedDurationUnits" : "hours",
        "canStartLater" : false,
        "isBundle" : false,
        "dateQualityId" : 4,
        "serviceLevelDescription" : "",
        "isOrderable" : true,
        "showOrderSummary" : false,
        "hideInServiceCatalog" : false,

```



```

        "hideInMyOrders" : false,
        "maxQuantity" : 0,
        "isReportable" : false,
        "serviceImage" :
"/RequestCenter/nsapi/definition/images?tenantId=1&type=def&documentId=108",
        "serviceURL" : "<a
href=/RequestCenter/myservices/navigate.do?query=serviceid&sid=149&layout=popup_p' onclick=\"return
GB_showFullScreen('Cassandra Cluster App (v2.1.1) (CLQ)', this.href)\">Cassandra Cluster App (v2.1.1) (CLQ)</a>",
        "serviceOrderURL" : "<a
href=/RequestCenter/myservices/navigate.do?query=orderform&sid=149&layout=popup_p' onclick=\"return
GB_showFullScreen('Cassandra Cluster App (v2.1.1) (CLQ)', this.href)\">Order</a>",
        "serviceURLOnly" : "/RequestCenter/myservices/navigate.do?query=serviceid&sid=149",
        "serviceOrderURLOnly" :
"/RequestCenter/myservices/navigate.do?query=orderform&sid=149",
        "orderingMode" : 2,
        "computePrice" : false,
        "serviceExtensions" : {
            "serviceExtension" : [
                {
                    "name" : "ServiceID",
                    "logicName" : "ServiceID",
                    "value" : "149",
                    "type" : ""
                },
                {
                    "name" : "searchFacets",
                    "logicName" : "searchFacets",
                    "value" : "{\"Ratings\":[\"None\"]}",
                    "type" : "Search Facets"
                }
            ]
        },
        "serviceAccessories" : [],
        "servicePrerequisites" : [],
        "portalText1" : "",
        "portalText2" : "",
        "portalText3" : "",
        "hasPortal1" : false,
        "hasPortal2" : false,
        "hasPortal3" : false,

```

```
        "isTemplate" : false,  
        "isBaseService" : true,  
        "templateType" : "Application",  
        "hoursPerDay" : 0.0,  
        "templateRelevant" : false,  
        "tenantRelevant" : true  
    }  
  ]  
}  
}
```

### 12.1\_Patch\_v2 で導入

IBM Cognos 10.2.2 のインストール手順については、12.1 readme ファイルを参照してください。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザー側の責任になります。

対象製品のソフトウェアライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

Cisco が導入する TCP ヘッダー圧縮は、カリフォルニア大学バークレー校 (UCB) により、UNIX オペレーティング システムの UCB パブリック ドメイン バージョンの一部として開発されたプログラムに適応したものです。全著作権所有。著作権©1981、カリフォルニア大学の評判。

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、Cisco およびその供給者は、このマニュアルに適用できるまたは適用できないことによって、発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性が Cisco またはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワークトポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

ハード コピーおよびソフト コピーの複製は公式版とみなされません。最新版はオンライン版を参照してください。

Cisco は世界各国 200 箇所にオフィスを開設しています。各オフィスの住所、電話番号、FAX 番号については、Cisco のウェブサイト [www.cisco.com/go/offices](http://www.cisco.com/go/offices) をご覧ください。

Cisco および Cisco ロゴは、シスコ またはその関連会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。シスコの商標の一覧は、[www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks) でご確認いただけます。掲載されている第三者の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語は、シスコと他社との間のパートナーシップ関係を意味するものではありません。(1110R)

© 2021 Cisco Systems, Inc. All Rights Reserved.