

ロボット構築 はじめの一步 for beginners training

本資料では、Kapow Katalyst のインストールを完了したユーザが初めて Design Studio を起動し、ロボットを作成、Management Console へデプロイするまでの流れについて初歩的な知識・手順を紹介します。本資料は Windows ユーザを対象に解説いたします。（Linux で Design Studio を使用するためには XServer の起動が必要です。）

なお、本文中の不明な用語については「[\[Kapow Katalyst 9.1\] Glossary.pdf](#)」を、ロボット作成時点での Step の操作方法などについては「[\[Kapow Katalyst 9.1\] Step handling.pdf](#)」をご覧ください。

内容

- Design Studio とは 2
- ロボットを作成する手順の確認 2
 - ロボットの目的や役割、達成すべき成果の明確化 2
 - ロボットが達成すべき成果を得るためのプロセスの明確化 3
 - プロセスの実行に必要なデータ項目の洗い出し 3
- Design Studio の起動 4
- プロジェクトの作成／登録 4
- Type によるデータ項目の定義 5
- Robot によるプロセスの定義 6
- デバッグ 8
- デプロイ（運用環境へのアップロード） 9
- Appendix1: 運用スケジュール登録 10
- Appendix2: Attribute 型一覧 11
- Appendix3: Step Action 型一覧 12

● Design Studio とは

Design Studio とは Kapow Katalyst における主要なツールの 1 つであり、ロボットを作成するための統合開発環境です。実際に動作するロボットは Robot（ロボット）・Type（型）・Snippet（スニペット）という要素から構成されており、そのすべてを Design Studio で作成することができます。

また、Design Studio はデバッガーでもあり、運用環境へのアップロードツールでもあります。つまり、ロボットの作成にかかわる全てのことは Design Studio 内で実施することができます。

● ロボットを作成する手順の確認

実際のロボットを作成する工程はすべて Design Studio で実施しますが、もっとも重要なことは、ロボットの作成に取り掛かる前にどんなロボットを作るのが明確になっていることです。

まずはやる気持ちを抑えて、作るべきロボットの構想を明確にしましょう。

以下に一般的なロボット開発の手順をまとめます。

1. ロボットの目的や役割、達成すべき成果の明確化
2. ロボットが達成すべき成果を得るためのプロセスの明確化
3. プロセスの実行に必要なデータ項目の洗い出し
4. Design Studio の起動
5. プロジェクトの作成／登録
6. Type によるデータ項目の定義
7. Robot によるプロセスの定義
8. デバッグ
9. デプロイ（運用環境へのアップロード）
10. 運用スケジュール登録

● ロボットの目的や役割、達成すべき成果の明確化

ロボットの目的や役割、達成すべき成果を明確にしましょう。

その際、何をしないかについても意識していることがロボットをシンプルに保つために重要です。

● ロボットが達成すべき成果を得るためのプロセスの明確化

業務処理をロボットで自動化する場合、その実行手順は担当者によってバラバラですし、自分でも意識していない事柄も時には存在します。ロボットを作り始めてからやっぱりこうかな？それともこうかな？と迷子になることを避けるために、作り始める前にプロセスを明確にしておきましょう。

(画面のハードコピーに具体的な操作内容を記述していくような画面フロー図がきっと役に立ちます。)



画面フロー図例

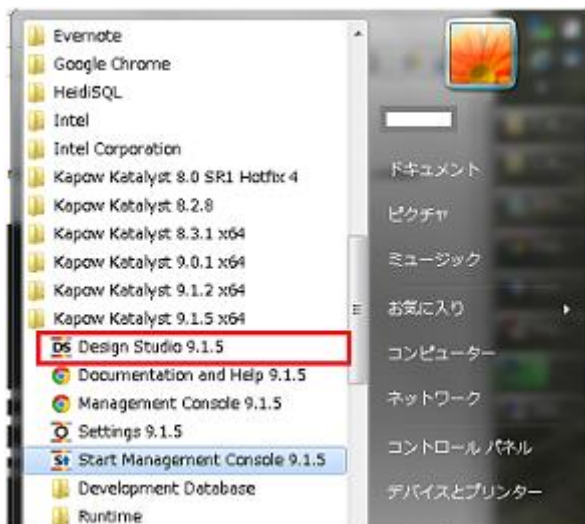
● プロセスの実行に必要なデータ項目の洗い出し

ロボットが実施する業務は●●の情報を▼▼に入力して、結果として表示される■■を抽出するといったパターンが多くなるでしょう。そして、●▼■のデータを扱うための領域をロボットに用意してあげる必要があります。細かい項目まで完璧に規定する必要はありませんが、登場人物とその特性をまとめておく程度の整理は必要でしょう。

入力データ		取得データ	
ログイン情報	ユーザーID	売上情報	日付
	パスワード		カテゴリ
			売上高
		来客情報	日付
			時間
			人数

● Design Studio の起動

Windows のスタートメニューから「Design Studio 9.1.x」を開き、Design Studio を起動します。
(ライセンスサーバの登録については既に終了している前提です。)



● プロジェクトの作成／登録

ロボットはその構成要素（.robot、.type、.snippet、Database Mapping など）をプロジェクトという範囲で管理共有します。

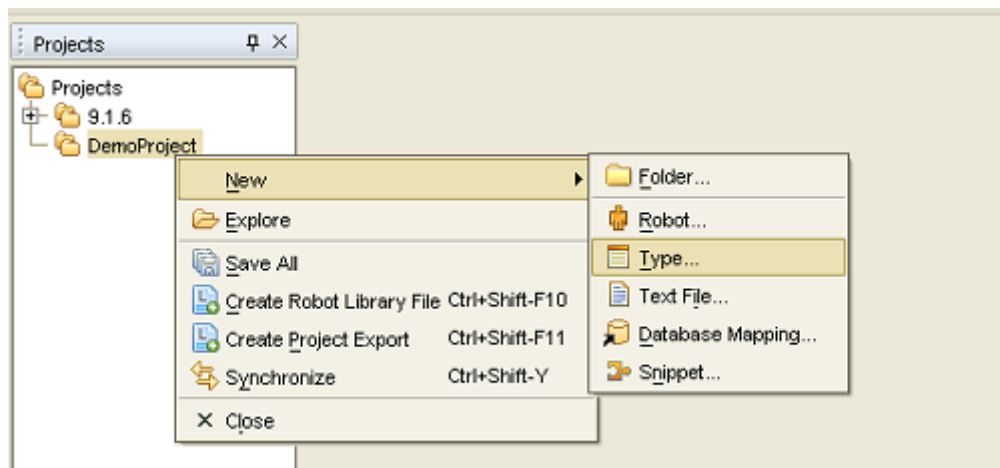
全く新しいプロジェクトを作成する場合にはメニューバーより「New Project..」を選択して新規のプロジェクトを作成、また別の環境で作成したプロジェクトを読み込んで新たに編集する場合には「Open Project..」を選択して既存のプロジェクトフォルダをプロジェクトとして登録しましょう。

メニューバーから指定	プロジェクトリストから指定

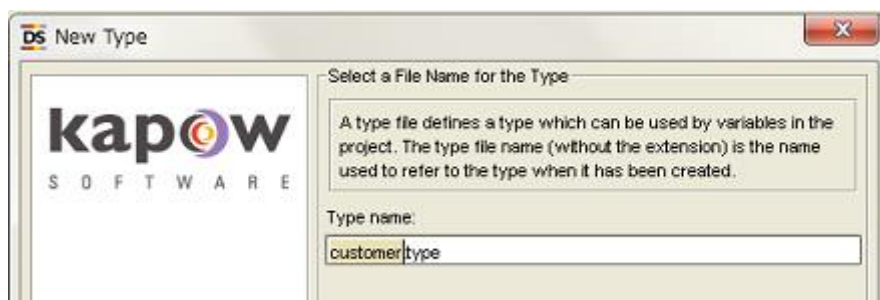
- **Type によるデータ項目の定義**

事前に洗い出したデータ項目を Type エディタで定義します。

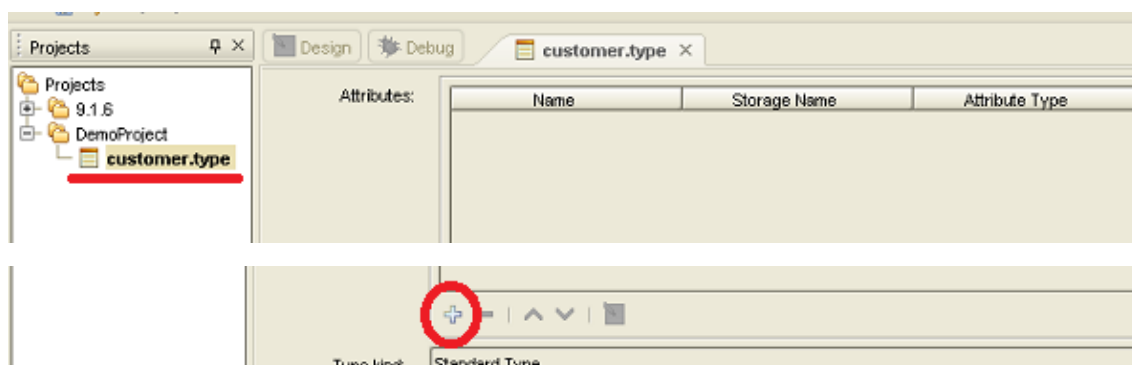
Type を追加したいプロジェクト上で右クリックし、「New」->「Type...」を選択します。



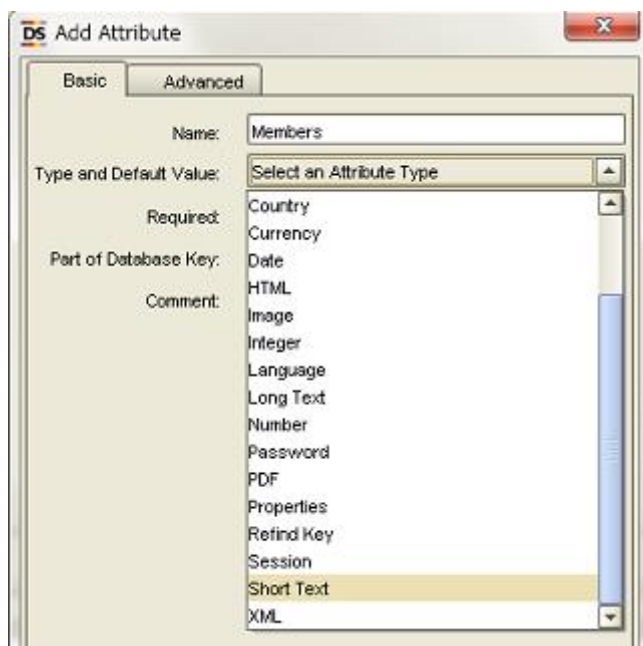
Type のファイル名に指定する文字列を入力し、「Finish」ボタンをクリックします。



Type ファイルが生成され、Type エディタに空の Type が表示されたら、画面中央の「+」ボタンをクリックして Attribute を追加します。

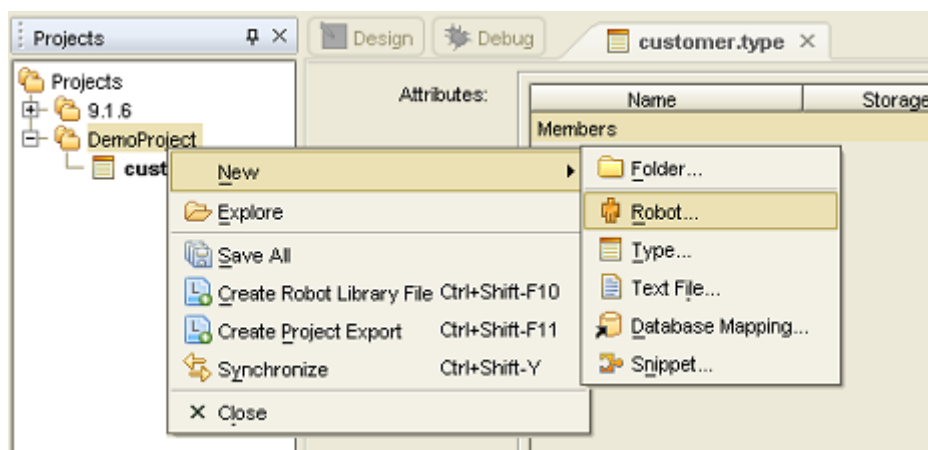


Attribute の名称と型を設定します。Attribute には様々な種類の型が用意されており、それぞれに入力可能な値が異なります。（詳細は Appendix2:Attribute 型一覧 参照）

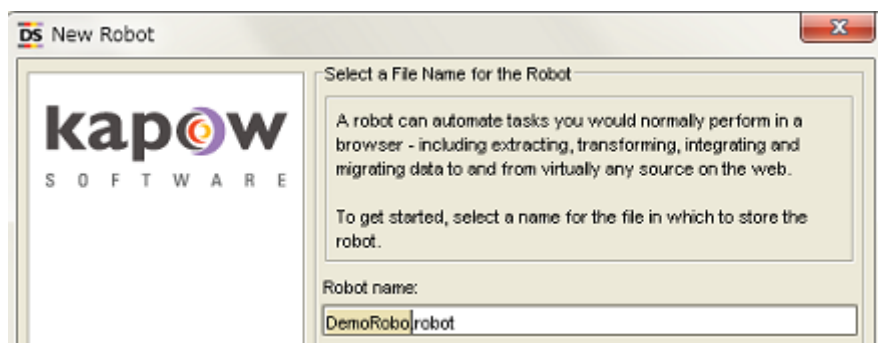


- Robot によるプロセスの定義

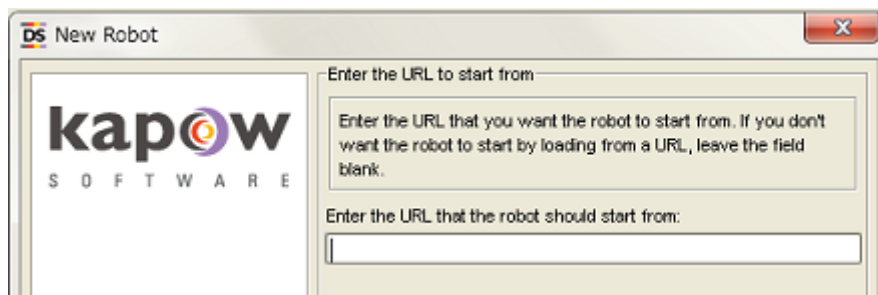
Robot を追加したいプロジェクト上で右クリックし、「New」->「Robot...」を選択します。



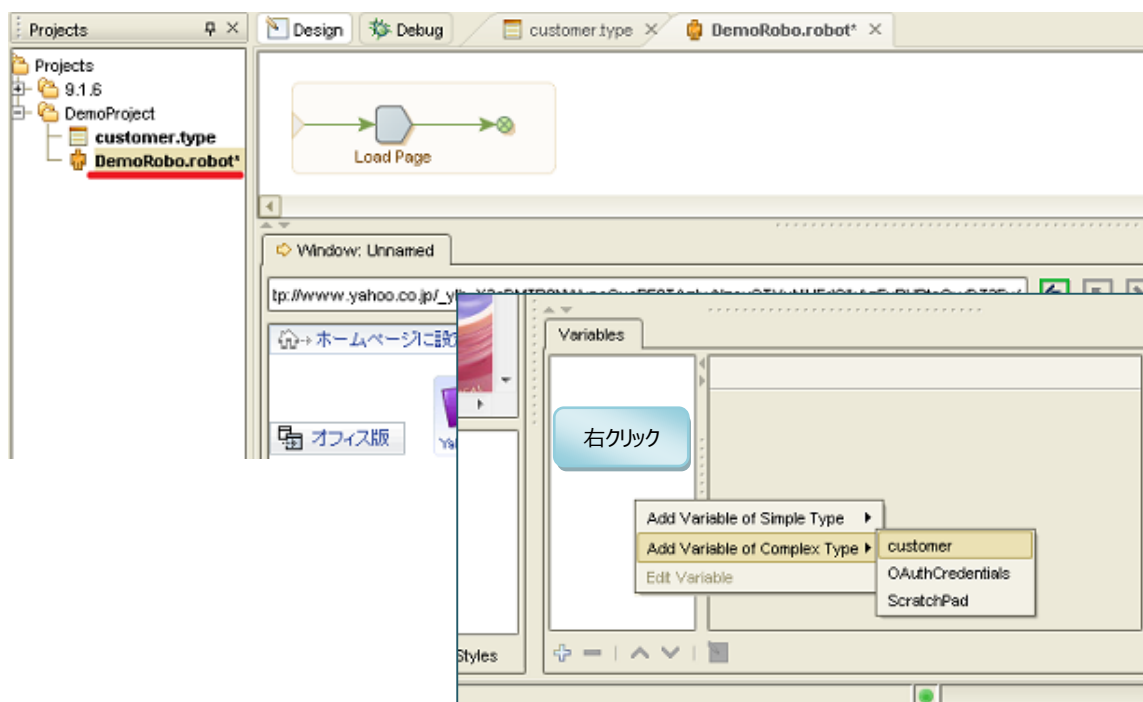
Robot のファイル名に指定する文字列を入力し、「Finish」ボタンをクリックします。



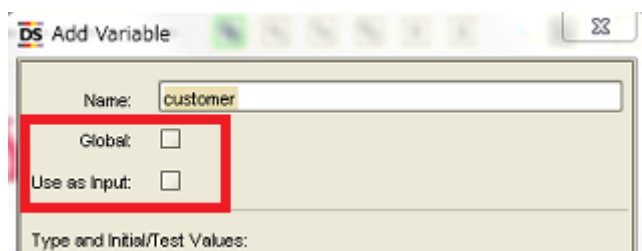
Robot に読み込ませるサイトの URL を入力し、「Finish」ボタンをクリックします。特に読みこませたくない場合は、空白のままでも OK です。



Robot ファイルが生成され、Robot エディタに Load Page ステップのみ（または空の）Robot が表示されたら、画面右下の Variables リストエリアで右クリックして事前に作成した Type を登録します。



用途によって「Global」または「Use as Input」を指定して OK をクリックします。



Type はロボットを作成していく過程でも追加することが可能です。また、Type 自体を修正した場合、その内容は自動的に Robot に登録されている Type にも反映されます。

以降、事前に明確化したプロセスをロボットに組み込んでいきます。

以下に一般的なプロセスの組み込み手順をまとめます。

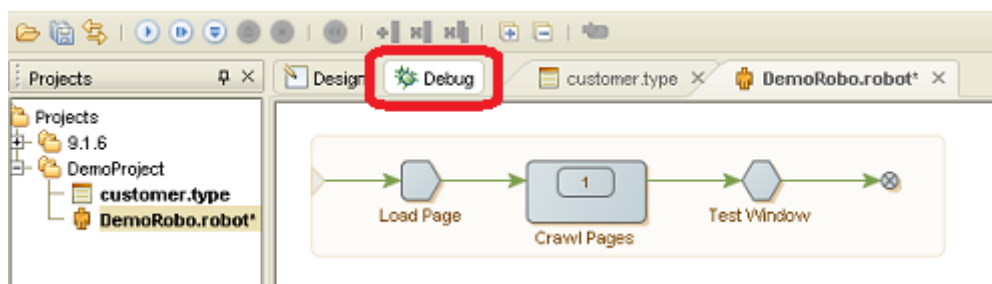
1. ブラウザビューに表示される Web の画面を操作して、プロセスの基本フローを作成。
2. 分岐、繰り返しなど画面から指定できないプロセスの制御条件や詳細な構造をロボットビューの各 Step に対して組み込み。
3. ロボット実行時にエラーが発生した場合の処理など、ロボットの品質を高めるための命令の組み込み。

なお、Design Studio で用意されている Action には 100 以上の機能が存在し、きめ細かい動作の設定が可能です。（詳細は Appendix3:Step Action 一覧 参照）

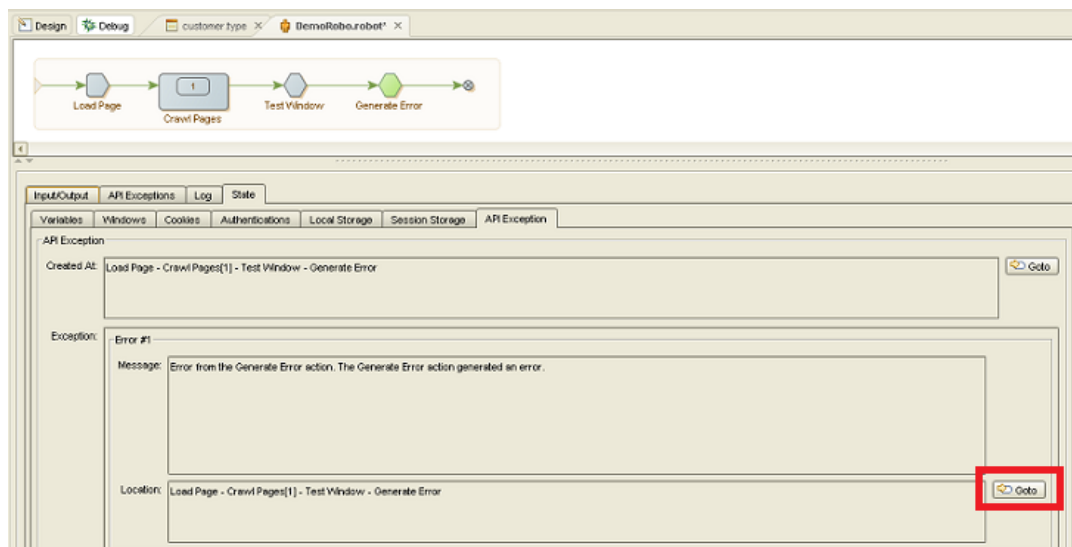
● デバッグ

ロボットの処理が完成したら、その品質を確認するためにデバッグを行います。

「Debug」アイコンをクリックしてデバッグモードに切り替えたうえで、三角の「Run」アイコン、もしくは F9 キーを押下してデバッグを実行します。



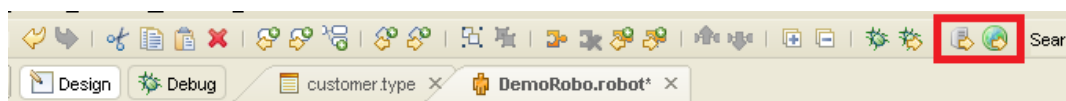
デバッグ中にエラーが出た場合には、エラー内容をよく確認するとともに、「Goto」ボタンをクリックして実際にエラーが発生した箇所をデザインモードで確認のうえ、修正しましょう。



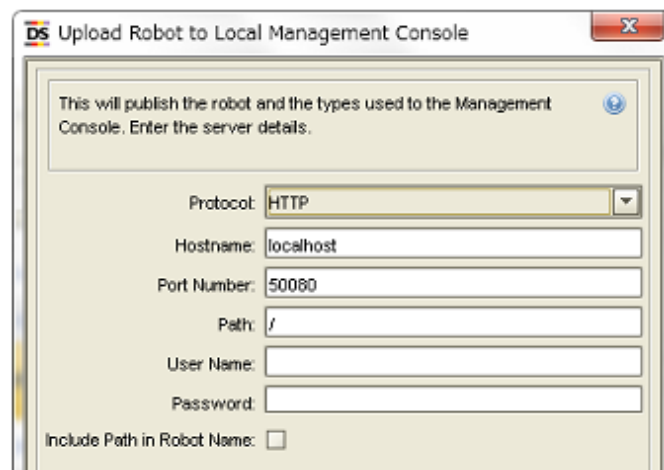
● デプロイ（運用環境へのアップロード）

デバッグを終え完成したロボットをデプロイします。

開発環境上の Management Console ヘデプロイする場合は下記左側のマシンアイコン、本番環境、またはステージング環境上の Management Console ヘデプロイする場合は下記右側の地球アイコンをクリックします。

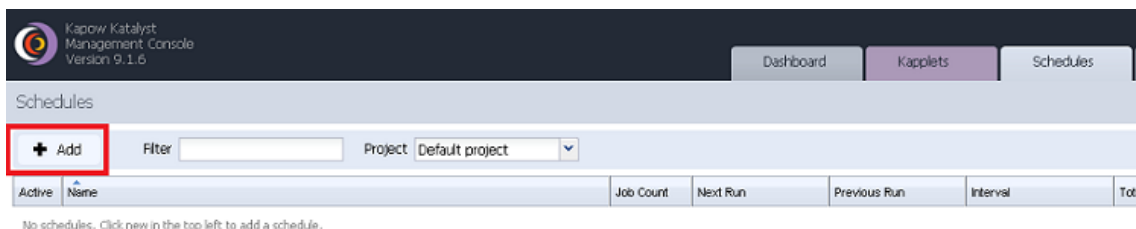


アップロード先サーバの情報を入力したら、「Upload」ボタンを押下してデプロイ手続き完了です。

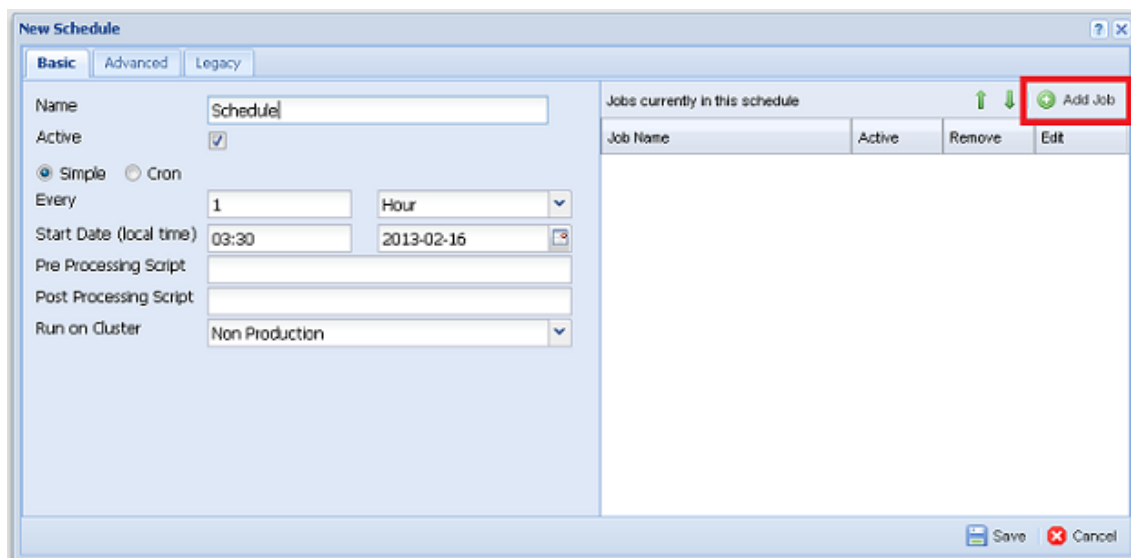


● Appendix1: 運用スケジュール登録

Management Console の Schedule タブを開き、「+ Add」アイコンをクリック。



「+ Add Job」アイコンをクリックして実行するロボットを設定します。



あとは画面の指示に従ってスケジュールを登録すれば完了です。

● **Appendix2: Attribute 型一覧**

Integer	整数 e.g. 12.
Number	実数 e.g. 12.345.
Boolean	正否値 i.e. either "true" or "false".
Character	1 文字 e.g. "A".
Short Text	1 行文字列
Long Text	複数行文字列
Password	パスワード文字列（入力値は暗号化され、*として表示）
HTML	HTML 文字列（ただし、プレビュー機能が付いている以外は Log Text と同じ）
XML	XML 文字列（ただし、整形された文字列を入力できること以外は Log Text と同じ）
Date	日付型（形式は必ず以下であること） yyyy-mm-dd hh:mm:ss.n, e.g. "1992-04-25 10:33:06.0"
Binary	バイナリーデータ
Image	イメージ（ただし、プレビュー機能が付いている以外は Binary と同じ）
PDF	PDF（ただし、プレビュー機能が付いている以外は Binary と同じ）
Properties	名前と値がペアになっているプロパティのリスト。（This is the same as a Long Text, except that the text represents a list of properties, where each property is a name/value pair.）
Session	ページ、URL、リファラー、cookies、認証情報、およびタイムスタンプで構成されるセッション情報
Currency	ISO-4217 standard で定義された通貨コード e.g. "EUR" for Euro.
Country	ISO-3166 standard で定義された国コード e.g. "DE" for Germany.
Language	ISO-639 standard で定義された言語コード e.g. "de" for German.
Refind Key	データ検索用の特別なリファインドキー

● Appendix3: Step Action 型一覧

Assign/Transform Variable

Action	Description
Assign Variable	Variable に値を割り当てます。
Convert Variables	1 つまたは複数の Variable の値を Converter を利用して変換します。
Transform XML	This action transforms XML using XSLT.

Browser Session

Action	Description
Save Session	別のロボットとセッション情報をやり取りするために Variable もしくは session プールにセッション情報を保存します。
Restore Session	Save Session で保存されている Session 情報を取り込みます。
Extract Cookie	指定した name, domain ,path のパターンにマッチする cookie の値を抽出します。
Create Cookie	指定した domain, path, name (,value) の cookie を生成します。
Remove Cookie	指定した name, domain, path ,value のパターンにマッチする Cookie を削除します。
Extract Web Storage	This action extracts data from the local and/or session storage. The data is stored in a variable in JSON format.
Load Web Storage	This action loads data into the local and/or session storage. The data must be specified in JSON format.
Clear Web Storage	local / session storage のデータをクリアします

Browser Windows

Action	Description
New Window	新しいウィンドウを開きます。
Set Current Window	現在選択されているのとは違うウィンドウを選択します。（選択することにより、そのウィンドウ内の情報を操作できるようになります。）
Close Window	指定したウィンドウを閉じます。

Call Web Service

Action	Description
Call REST Web Service	このアクションは REST web service を呼び出し、戻り値をカレント

	ウィンドウ内もしくは variable 内に展開します。
Call SOAP Web Service	このアクションは SOAP 形式での XML リクエストの提出およびレスポンスの取得を行います。

Click/Move Mouse

Action	Description
Click	指定タグに対してマウスクリックをエミュレートします。
Move Mouse To	指定タグに対してマウスオーバーをエミュレートします。
Move Mouse From	指定タグに対してマウスアウトをエミュレートします。
Scroll	ドキュメントやタグに対するスクロールをエミュレートします。

Database

Action	Description
Store in Database	データベースに値を投入します。
Find in Database	データベースから値を抽出します。
Calculate Key	This action calculates the key that will be used to store the value of the selected variable.
Delete from Database	データベースから値を削除します。
Query Database	SQL の SELECT 文を実行して結果を取得します。
Execute SQL	INSERT,UPDATE,DELETE などの SQL 文を実行します。
Store in HBase Table	HBase のテーブルに値を投入します。

Enter Data in Form

Action	Description
Enter Text	テキストフィールドへテキストを入力します。
Enter Password	テキストフィールドへパスワード文字列を入力します。
Press Enter	フォームに対する Enter キー入力をエミュレートします。
Select Option	ドロップダウンボックスやリストボックスのオプションを選択します。
Select Multiple Options	リストボックスから複数のオプションを選択します。（ドロップダウンボックスに対しては使用できません）
Set Checkbox	チェックボックスを選択／非選択にします。
Select Radio Button	ラジオボタンを選択します。
Select File	アップロード用のファイルを選択します。

Extract

Action	Description
Extract	画面からテキストを抽出して variable にセットします。
Extract Selected Option	選択されたオプションのテキストや値を抽出して variable にセットします。
Extract URL	選択したタグから URL を抽出して variable にセットします。
Extract Clip	選択したタグから独立した HTML クリップを抽出します。
Extract Image	画像を抽出して variable にセットしたり、ファイルとして出力します。また、併せてファイル名やファイルタイプも抽出することができます。
Extract Target	URL をターゲットにバイナリデータを抽出して variable にセットしたり、ファイルとして出力します。また、併せてファイル名やファイルタイプも抽出することができます。
Extract Tag Attribute	選択したタグからタグ属性を抽出して variable にセットします。
Extract Form Parameter	選択したタグ上の URL から指定したパラメータの値を抽出します。
Extract CSV	選択したタグから 1 行単位で CSV データを読み取ります。 選択対象のタグには CSV 形式のデータで構成されている必要があります。
Extract from Excel	Variable に格納された Excel ファイルのデータからスプレッドシートの情報をカレントウィンドウに抽出します。
Extract from Flash	Flash オブジェクトから構成情報を抽出します。
Extract from PDF	Variable に格納された PDF ファイルのデータからテキスト情報をカレントウィンドウに抽出します。
Extract Binary Content	カレントウィンドウからバイナリコンテンツを抽出します。

File System

Action	Description
Load File	ローカルファイルのコンテンツをカレントウィンドウや Variable へロードします。
For Each File	指定したディレクトリ内に存在するファイル数分繰り返しを行います。
Write File	ファイル出力をします。
Test File Existence	指定したファイルの有無により処理の継続・停止を判定、制御します。
Get File Info	ファイルシステム上のメタデータを取得します。（最終更新日時、ファイルサイズ）

Copy File	ローカルファイルをコピーします。 但し、コピー先のファイルが既に存在する場合はエラーとなります。
Delete File	指定したローカルファイルやディレクトリを削除します。
Make Directory	ディレクトリを作成します。
Rename File	指定したローカルファイルまたはディレクトリをリネームします。 但し、リネーム先のファイルやディレクトリが存在する場合にはエラーとなります。

Loop

Action	Description
For Each Tag	指定したタグの直下にあるタグに対して、その数分繰り返します。
For Each Tag Path	指定したタグの配下にある特定パス上のタグに対して、その数分繰り返します。（直下である必要は無し）
For Each URL	指定したタグの配下にある URL を含むタグに対して、その数分繰り返します。
For Each Window	タブで開かれているウィンドウ数分繰り返し、それぞれを順にカレントウィンドウにセットします。
For Each Text Part	文字列を特定の区切り文字で分割し、切り分けられた要素の数分繰り返します。
For Each Option	ドロップダウンボックスやリストボックス内のオプションに対し、その含まれる数分繰り返し、それぞれを順に選択します。
For Each Radio Button	ラジオボタンのグループに対し、そのグループに含まれるラジオボタン数分繰り返し、それぞれを順に選択します。
Loop Field Values	テキストフィールド内に特定の値の集合を、集合の数分一つずつ繰り返し入力します。
Repeat	Next アクションとペアで延々に繰り返す処理を生成します。
Next	Repeat アクションとペアで延々に繰り返す処理を生成します。
Crawl Pages	指定した URL を起点にリンクされているページ数分ページをクロールします。
Get Iteration	繰り返し中のループにおけるその時点での繰り返し数を取得します。

Load Page

Action	Description
Load Page	URL で指定されたページをロードします。

Create Page	新しいページを作成します。
Raw HTTP	選択したメソッドで Raw HTTP 通信を実行します。

Make Snapshot

Action	Description
Make Snapshot	フレームや画像などのリソースを含め、カレントウィンドウのスナップショットを取得します。
Rewrite Page	The Rewrite Page step action extracts the HTML content of the current window and additionally rewrites and outputs the links to style sheets, images and other pages.
Rewrite Style Sheet	The Rewrite Style Sheet step action acts as a helper for Rewrite Page. It task is to rewrite links to other style sheets or images in a given style sheet.

Modify Page

Action	Description
Insert Tag	新しいタグを挿入します。
Replace Tag	指定したタグを新しいタグで置き換えます。
Remove Tags	指定した特定のタグを削除します。タグは複数設定可能です。
Remove Tag Range	指定した範囲のタグを削除します。.
Set Top Tag	指定したタグを最上位タグに変換し、その範囲外タグを削除します。
Hide Tag	指定したタグを非表示にします。
Unhide Tag	指定した非表示タグを表示します。
Divide Tag	This action divides the found tag into sub-tags, one of which is outputted per iteration.
Divide Text	This action divides the text in the found tag into pieces.
Divide Table	This action divides the input <table>-tag into several sub <table>-tags, one of which is outputted in each iteration.
Remove Table Rows	This action removes from the input <table>-tag all rows (<tr>-tags) that do not have a specified number of columns (<td>- and <th>-tags).
Transpose Table	Table タグの縦列・横列を入れ換えます。

Normalize Table	Table タグの rowspan、colspan 設定を無効化し、N×N のテーブルフォーマットに整形します。
---------------------------------	--

Other

Action	Description
Set Current Tag	後続の手順でタグを検索するときにリファレンスとして使用することができるように、カレントタグとして指定したタグにマークを付けます。
Clear Current Tags	後続の処理で不要なカレントタグをクリアします。
Do Nothing	何も実行しないステップを作ります。
Wait	指定した時間、一時的にロボットを停止させます。
Stop	ロボットの実行を停止させます。
Generate Error	明示的にエラーを生成します。
Execute Command Line	コマンドライン／シェルを実行します。処理の実行に当たっては、RoboServer に十分な権限が付与されていることが前提です。
Change Proxy	プロキシサーバを切り替えます。
Execute JavaScript	JavaScript を実行します。

Output

Action	Description
Return Value	ロボットから呼び出し元に値を返します。
Send Email	メールを送信します。（添付ファイル可）
Write File	任意の情報をファイルとして出力します。
Write Log	指定したメッセージをログに書き込ませます。

Test

Action	Description
Test Tag	指定したタグの内容を判断条件にフローを制御します。
Test URL	指定したタグ内の URL の内容を判断条件にフローを制御します。
Test Value	正否値を判断条件にフローを制御します。
Test Variables	指定した Variable の値を判断条件にフローを制御します。
Test Row	This action tests the number of columns in a table row.
Test Window	指定した条件を含むウィンドウが存在するかどうかを判断条件にフローを制御します。