# ソフトウェアロボット
# の作り方

― RPA を実現する ROBOWARE ―

# ソフトウェアロボットを作るには

ソフトウェアロボットがあれば、面倒な事務作業などの PC 操作が自動化できます。

通常ソフトウェアロボット制作するためには、かなり高度なプログラミング能力が要求されます。たとえば、マウスやキーボードの入力操作を一からプログラミングすることは、経験が少ないプログラマーや、Windows 等のパソコンの内部の仕組みを知らない人にとっては、とても難しく、時間もかかり、専門知識のあるプログラマーでなければ作成することさえ困難となります。

そのため、多くの RPA のツールベンダーは、メモリやストレージがいろいろ選択できる BTO パソコンのようにシステム化して、GUI で比較的簡単に設定できるソフトウェアロボットを業務に合わせ、いろいろな種類のパターンで作成しました。
これにより希望するソフトウェアロボットの形に近いものを選んでカスタマイズできれば、簡単にソフトウェアロボットが作成できます。

しかし、企業や組織の業務は多種多様で、すでにある形に無理に合わせようとすると、希望するソフトウェアロボットが作成できない場合も多いです。
自作パソコンをイメージしてみて下さい。まず PC のフレームを用意して、メモリやディスクやファンなど、すでに完成された部品を利用するでしょう。気に入った部品を揃えて組み立てれば、そんなに高度なパソコンの知識がなくても、思い通りのパソコンができるはずです。

ソフトウェアロボットについても、同様にフレームワーク化された API を搭載した ROBOWARE を使用すれば、高度なプログラミング知識がなくても、比較的容易に作成することができます。

目的に合ったソフトウェアロボットを作成するために、Ruby や PHP などの汎用プログラミング言語を使用して制作できる ROBOWARE での作り方と実行方法を解説します。

*ROBOWARE は、株式会社イーセクターの登録商標です。

# ROBOWARE --- ソフトウェアフレームワーク

ROBOWARE は、ソフトウェアロボットを開発、実行するためのソフトウェアフレームワークです。

ソフトウェアフレームワークとは、ソフトウェアロボットの開発・運用を行う際に、その基礎となるルール・構造・アイデア・思想などの集合で、ROBOWARE にはソフトウェアロボットを作成するために多くの API が含まれています。

ROBOWARE のソフトウェアフレームワークにより、ソフトウェアロボットの骨格が均意され、開発も容易で、実行時にソフトウェア同士の連携が簡単に取れます。

プラグラミング言語としては、Ruby、Java、PHP、C# にて作成されるスクリプトに対応したソフトウェアロボット専用のソフトウェアフレームワークです。


## フレームワークのメリット

ソフトウェアロボットを作るために必要な枠組みはすでに出来ているため開発コストが下がり、ある程度の品質が保証されます。また、設計、構成などがフレームワークとして用意されており、機能が明確に分離されているため、保守性が上がり、またコードが読みやすくなる、というメリットもあります。

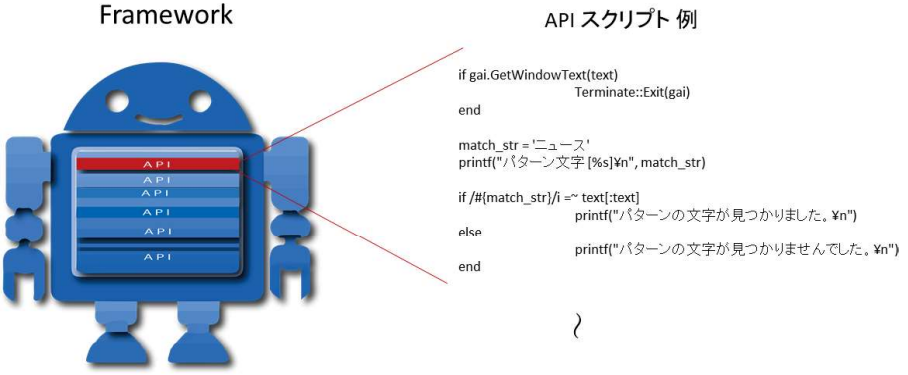
# ROBOWARE --- API

ROBOWARE は、ソフトウェアロボットを開発し、実行・運用するための多くの API（Application Programming Interface）を実装しております。

API とは、ソフトウェアが、他のソフトウェアとお互いやり取りをするために機能を共有出来るようにするためのインターフェースのことです。
たとえば、異なるアプリケーションを連携させるためには、通常内部のレコードフォーマットなどの仕様に合わせてアクセスできるようにしたインターフェースが必要なため、目的に応じた API を用意します。

ROBOWARE の場合の API とは、ROBOWARE に搭載されたソフトウェアフレームワークに対して、ソフトウェアロボットを開発、実行するためにスクリプトにより指示できるように実装されたインタフェースです。
この API に引数として、指示にひつような情報を与えることによって、ソフトウェアロボットとして稼働するための動きが指示できます。

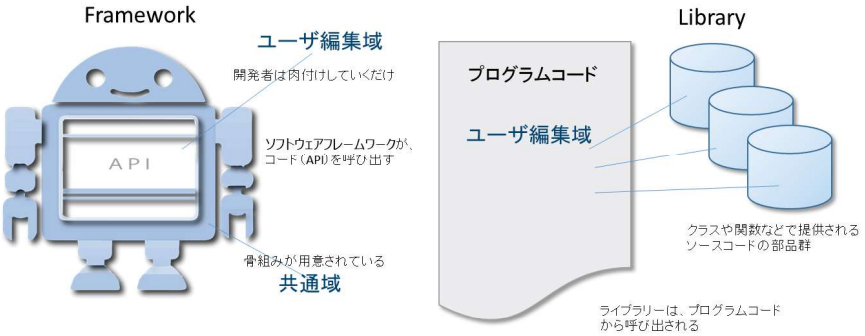
API を利用することで、どのような文字列をコピーしたり、クリックするかなどを複雑なプログラムコードを記述することなく、引数によって簡単に指示できます。
ROBOWARE の API の引数は、基本として ハッシュ型を採用しております。ハッシュは、連想配列とも呼ばれ、文字列を キー とする配列で、1 つの API に対し、いろいろな指示が可能です。



# ソフトウェアフレームワークとライブラリの違い

ROBOWARE は、ソフトウェアロボットが制作できるソフトウェアフレームワークです。

ソフトウェアフレームワークは、定義された API を持ち、具体的な実装を再利用可能な形で持っている点でライブラリとよく似ています。
しかし、ライブラリではプログラムの制御に関する主導権がプログラムコードにあるのに対し、ソフトウェアフレームワークでは、フレームワークがコード（API) の制御を行ないます。




プログラムのライブラリは、通常共通で使用できるプログラムの部品をひとまとまりのファイルにしたものです。
ソフトウェアロボットの場合、そのライブラリに共通部品を置いて使用したとしても、メインとなるプログラムコードを記述しなければならず、通常のプログラム開発をする工程と同じとなります。

一方、ソフトウェアフレームワークである ROBOWARE は、既にソフトウェアロボットの枠組みが構成されているため、API に引数を渡すスクリプトをコーデイングするだけで、ロボットに指示が与えられます。データの選択条件や、エラー時のハンドリングなどを詳しく指定する必要がある場合でも、通常のプログラム言語によるコーデイング追加によって、ロボットに対し更に詳しい指示が可能になります。
また、ソフトウェアロボットが動くためには、別の PC からの指令や、リモートから制御する仕組みが必要な場合があります。ROBOWARE であれば、単独のプログラムと比較して、実行・管理が出来る仕組みも予めフレームワークに備わっているという利点もあります。

# ROBOWARE の導入

ソフトウェアロボット制作のために、ROBOWARE を、操作を代行する PC に導入します。

稼働環境：Windows　および　Linux

インストーラーの起動で簡単に導入が出来ます。

（使用中のウィルス・スパイウェア対策のソフトに対象外の設定をする必要があります。）

- RBF API　：ROBOWARE が提供する各種 API
- RBF サーバ　：RBF API の命令に基づき処理を実行、または通信リレーを行うサーバソフトウェア
- サンプルスクリプト集：API のコーディング例

ROBOWARE の導入として、RBF ソフトウェアをインストールすると、上記、３つが使用可能になります。

導入後、マニュアルも導入フォルダ内に格納されます。
各 API、サンプルスクリプト集もプログラミング言語ごとに 4 種類用意され、Ruby、Java、PHP、C# に分かれてそれぞれのリファレンスマニュアルがあります。
RBF サーバのインストール後、RBF アプリケーションを起動するランチャーのショートカットが設定されます。

次に、ROBOWARE で開発したプログラムを管理するための Robowiser Manager をインストールします。


Robowiser Manager を導入すると、ROBOWARE で作成されたソフトウェアロボットの実行や管理ができます。

（導入後、ライセンスの登録が必要です。）

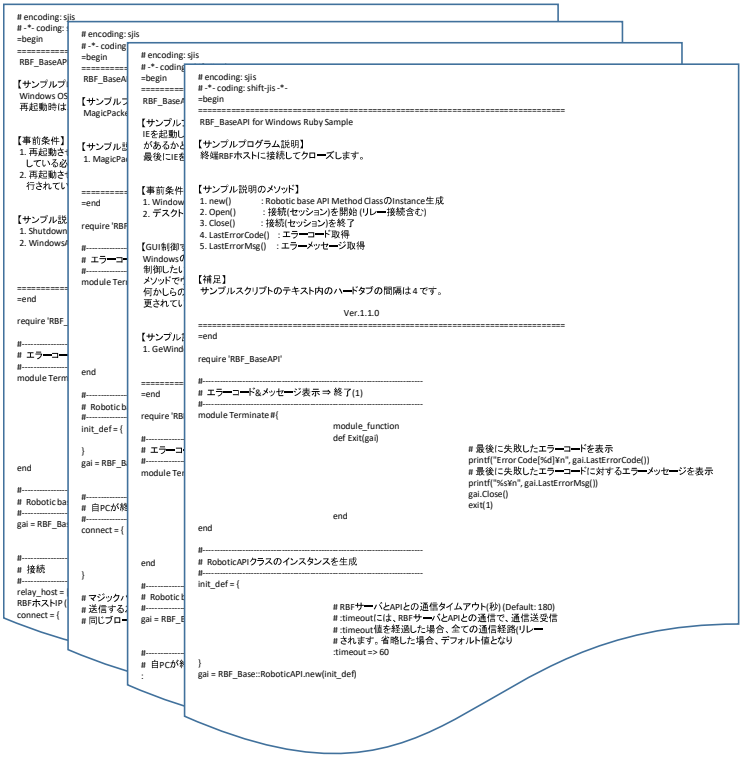
# サンプルスクリプトを選ぶ

API で使用するスクリプト用の言語を Ruby、Java、PHP、C# より選択します。

Ruby、Java、PHP、C# の各言語ごとに用意されている 50 種類以上のサンプルスクリプトファイルより、参考になりそうなサンプルコーディングを選びます。

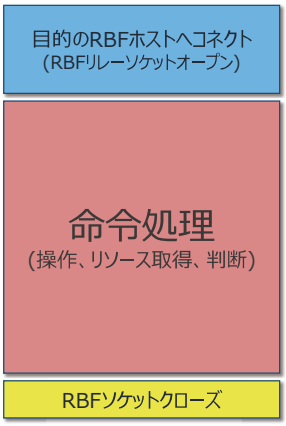
各言語のマニュアルを基に、API を使い、ROBOWARE にどのようにオペレーションするかを、プログラミング言語によって指示します。

このサンプルを参考に、言語の種類ごとユーザが使い慣れたエディタを使用してコーディングします。



# RBF API プログラム開発 リレー通信

ROBOWARE の開発言語基本記述構成




（Javaの例）

```
int  timeout = 30;
gai = RBF_Base::RoboticAPI.new

RBF_BaseAPI.Open connect = new RBF_BaseAPI.Open();

                    ～

RBF_BaseAPI.GetPID pids = new RBF_BaseAPI.GetPID();

pids.proc_name = "RBF_Server.exe";


if(gai.GetPID(pids)){
      Exit(gai);
}

System.out.printf("PID [%d] %n", pids.pid[0]);

                    ～

gai.Close()
```

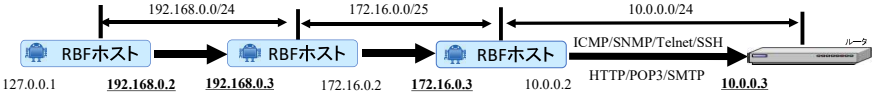
＊Ruby, Java, PHP, C#が使用可能

配列変数にリレーさせる経路順に IP(FQDN) 設定します。


```
  :
connect[:rbf_server]    = '127.0.0.1'
connect[:relay_host]    = [ '192.168.0.2', '192.168.0.3', '172.16.0.3' ]
  :
# ソケットオープン
if gai.Open(connect) then Terminate::Exit(gai); end
  :
```

リレーさせる場合、RBF_Serverは、原則 127.0.0.1 を指定します。

処理を分散させるなどのクラスタリング等を行う場合、別のRBFホストのRBF_Serverの IP(FQDN)を指定します。




```
  :
connect[:rbf_server]    = '127.0.0.1'
connect[:relay_host]    = [ '192.168.0.2', '192.168.0.3', '172.16.0.3' ]
  :
# ソケットオープン
if gai.Open(connect) then Terminate::Exit(gai); end
  :
ping[:dest_addr]    => '10.0.0.3'
ping[:timeout]      => 5
  :
if gai.Ping(ping) then Terminate::Exit(gai); end
  :
```

ROBOWARE は、処理したい目的の RBF ホスト（サーバ）が別のセグメントにある環境など、ROBOWARE に作成した API からの命令を、通信途中にある RBF ホストでリレー（中継）させることができます。
リレーする RBF ホストは、複数中継させることができます。

# API の種類

ROBOWARE は、Windows または Linux で動作可能な約 80 の API を提供します。

ROBOWARE の API に、プロセス名や座標などの引数を与えるだけで簡単にスクリプトが作成できます。

**ROBOWARE APIの例**

| | | |
|---|---|---|
| GetActiveProcessNum() | 実行されているプロセス数を取得 | |
| GetCPUusageRate () | 指定のPIDまたはOS全体のCPU使用率を取得 | |
| AddRemovePrograms() | インストールされたソフト一覧の取得/アンインストール | |
| ExecProcess () | プロセスや実行ファイルを起動 | |
| FileTransfer () | ファイルを転送 | |
| SendMail() | メールを送信 | |
| WindowCapture() | ウィンドウをキャプチャーする | |
| KeyboardTyping() | 指定のウィンドウハンドルにキー入力 | |
| TermPrint() | Telnet/SSHサーバにコマンドを送信する | …etc |

スクリプトを実行する RBF ホストの OS が Windows の場合に、終端でソフトウェアロボットが稼働する RBF ホストの環境が Linux であっても制御可能です。
同様に、スクリプト実行 RBF ホストの OS が Linux で、終端 RBF ホストが Windows でも可能ですが、終端 RBF ホストの OS で指定できる引数が異なる場合があります。

ROBOWARE には、多数のカテゴリーの API があり、各プログラミング言語のリファレンスマニュアルには、該当 API がどのサンプルスクリプトに例文があるのか記載されています。

- TCP 接続開始、終了、エラーに関する API Method
- OS またはプロセスのリソース取得設定に関する API Method
- OS またはプロセス制御に関する API Method
- 通信に関する API Method
- メールに関する API Method
- Windows UI 制御に関する API Method　　・・・　他

# ROBOWARE の画面表示認識

モニターに表示されているウィンドウのテキスト文字を RBF API の引数に指定して、目的の操作 ( クリック・キー入力等 ) を行います。
遠隔で制御したいロボット (RBF ホスト ) のモニターの解像度等に関係なく開発することができます。

① ウィンドウに表示される「親ウィンドウのみ」または「親ウィンドウと子ウィンドウ」の２つのテキスト文字を引数に指定して検索します。

②モニターに表示されているテキスト文字を検索する場合、ワイルドカード検索もできます。
・Windows OS が異なる場合で「サーバー」と「サーバ」と表示が異なった場合でも「サーバ *」と指定することで検索することができます。

③モニターに表示されている文字で同じ文字が複数あった場合は、検索して見つかった順番の番号、または Z オーダーを引数に指定します。

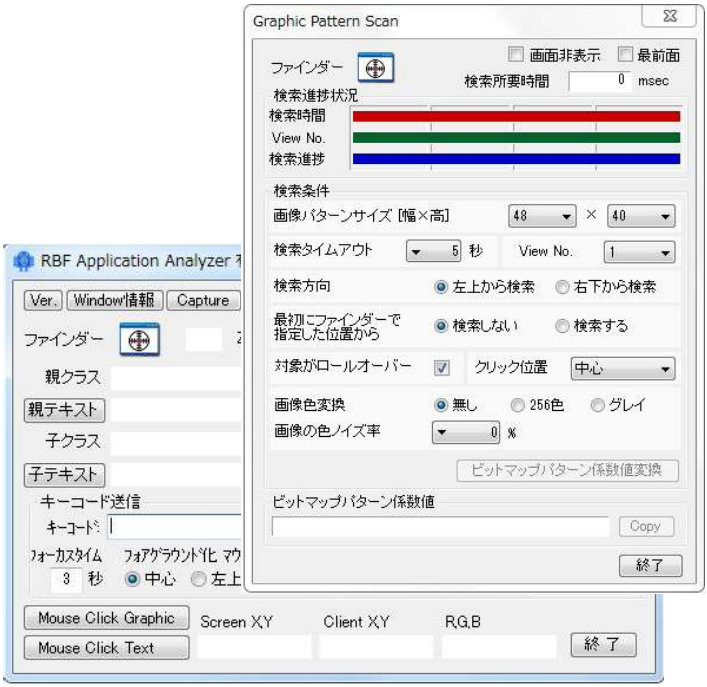
④モニターに表示されていないテキスト文字の場合 ( モニター枠からはみ出している場合など ) は、検索することができません。
・但し、ウィンドウがモニター外にはみ出していると判断した場合、自動的にモニター内にウィンドウが移動します。

# ROBOWARE　アナライザー

マウス・キーボード操作の開発は、RBF アナライザーを使用します。

自動化したいアプリケーションの画面を基にマウスやキーボード操作について、RBF アナライザーを使用して RBF API のメソッドの引数に渡す情報を取得できます。
これにより、画面上の操作を簡単にスクリプトコードにジェネレートできます。

典型的な RPA ツールであれは、ツール独自の GUI によって、スクリーン上の画像の座標を自動認識し、連続した操作を設定できるものが多いですが、ROBOWARE は、いろいろな条件で操作をハンドリングできるように、API の引数の情報をアナライザーによって取得し、その後、作成者自身でスクリプトに反映する方法となります。つまりプログラミングを支援するユーティリティとなります。
ROBOWARE は、スクリーン上のテキストとグラフィック両方のパターンを認識できます。

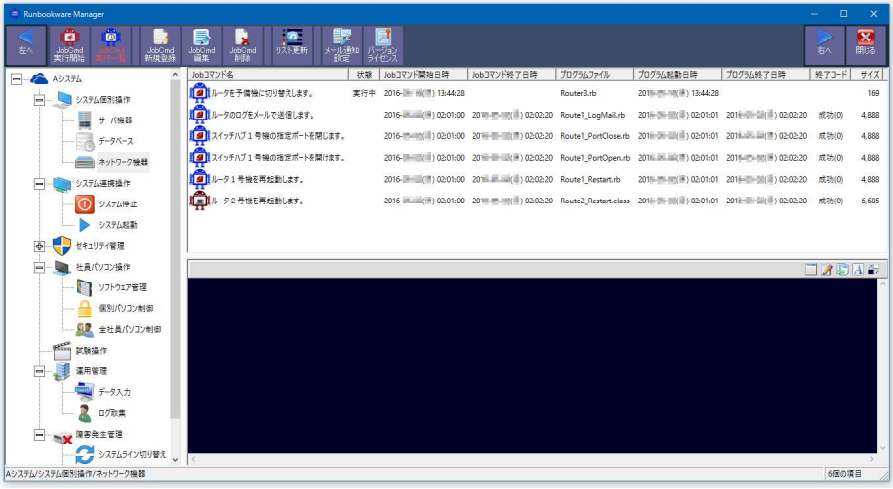


# ROBOWARE　プログラム管理

## ROBOWARE – Job Manager

ROBOWARE で開発したソフトウェアロボットを実行・管理

RBF API で開発したプログラムを Job コマンドとして、管理するソフトウェアです。
Job Manager の画面は、タッチパネルモニターの場合に、Job コマンドを実行するまでの操作を指でタップし易いように設計されています。




・Job の実行スケジュール機能
・Job の開始と終了結果のメール通知機能　など

よく使用するランチャーボタンを一覧表示し、Job コマンドを用途別にフォルダのツリー形式で管理でき、Job コマンドの一覧が出るエリアと、詳細を表示するエリアに分かれて表示されます。
分割されたエリアのサイズ変更や、フォルダのアイコンの変更も可能です。

# Job コマンドの新規登録

作成したソフトウェアロボットを実行するために Job Manager で Job コマンドを作成します。

Job Manager の Job コマンドツリービューに表示されたフォルダ毎に、Job コマンド登録情報を作成できます。
実行する Job について、プログラムファイルのパスや、プログラムを実行する時の引数、起動時に引数を入力させるかどうかなどを指定できます。
また、Job コマンドリストビューのから選択で、Job コマンド実行方法の設定ができます。

# ソフトウェアロボットの実行方法

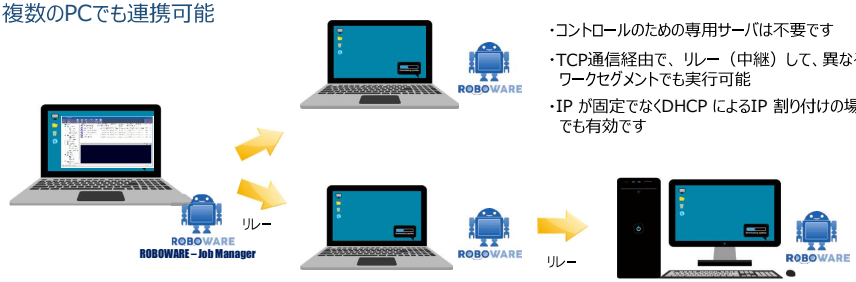
ROBOWARE には、スケジューラ機能もあり、作成したソフトウェアロボット毎に Job Manger で予め実行形態を登録できます。

・即時実行　　・時間サイクルで実行
・指定日時で実行　　・週サイクルで実行


バッチあるいはユーティリティコマンドからも起動停止が可能なので、他のスケジューラへの組み込みや、アプリケーション連携が簡単に行えます。

ROBOWARE は、他の ROBOWARE が導入済みの PC やサーバに対し、実行指令が可能です。
これによって、IP ネットワークで接続されていれば、テレワークに対応して、遠隔地の自宅やサテライトオフィスからでも、制御することが可能になります。
中央管理のための専用サーバは必要なく、ROBOWARE 同士で、異なるネットワークセグメントにおいてもリレー（中継）して実行することが可能で、相互のバックアップ（冗長化）になることもできます。

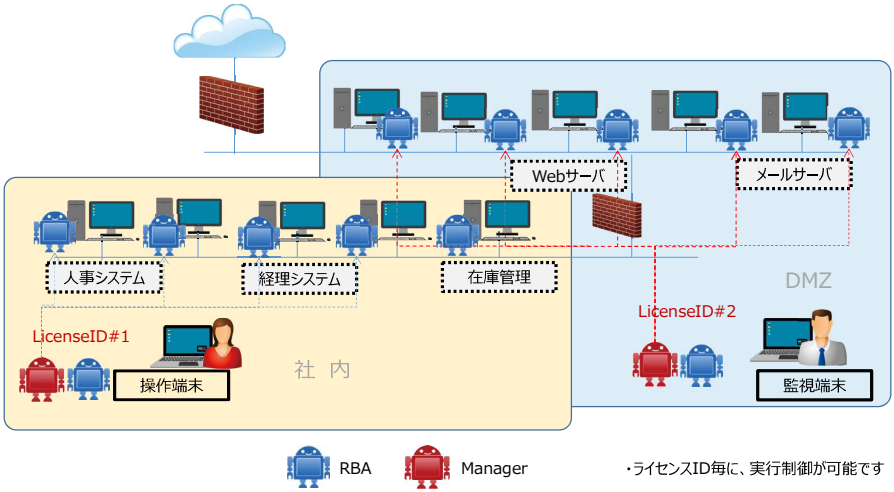
## 複数のPCでも連携可能




・コントロールのための専用サーバは不要です
・TCP通信経由で、リレー（中継）して、異なるネットワークセグメントでも実行可能
・IP が固定でなくDHCP によるIP 割り付けの場合でも有効です

# ROBOWARE　構成例

操作端末からの操作は社内のみ、監視端末からは DMZ のみの端末操作を行う場合の例




ROBOWARE は、ライセンス ID でお互いの通信を制限できます。
同じ社内の異なるシステムに ROBOWARE が共存する場合であっても、外部の ROBOWARE からそのシステムを制御されない作りになっております。

典型的な例では、実行したい PC 上に ROBOWARE を導入し、それぞれ作成したソフトウェアロボットが動く環境にします。
実行指示や、実行結果の管理を行う PC あるいはサーバには、ライセンス ID 毎に Job Maneger を導入して制御します。

ROBOWARE でソフトウェアロボットを作成すれば、様々な業務形態に対応して比較的簡単にオペレーションの自動化が実現できます。

（編集：春日井）

CCS 株式会社シーイーシーカスタマサービス
■お問合せ　ESECinfo@cec-ltd.co.jp
〒150-0022 東京都渋谷区恵比寿南 1-5-5 JR 恵比寿ビル 8F
TEL：03（5789）2443　FAX：03（5789）2575

（2017年7月4日公開の情報です）