

Help

Advanced Topics

This section covers advanced topics which may be of interest to system administrators and developers.

Help

Blue Prism Resource PC Modes

Blue Prism has several modes of operation, each of which can be selected using command-line options. See the [command-line reference](#) for full details of these options.

Interactive Mode

Normally, Blue Prism runs in interactive mode. While running in interactive mode, the local PC is available as a Resource PC only to the user logged in to Blue Prism at the time. Only one interactive session may run on a PC at any one time.

The interactive run mode may be disabled via [System Manager](#) in System, Settings, (uncheck "Start the process engine on this machine when users sign in") in order to avoid it conflicting with the [Resource PC](#) mode, for example).

Resource PC Mode

In Resource PC mode, Blue Prism functions as a Resource PC, listening for instructions from remote instances of Control Room. A Resource PC can be [public](#) (any Blue Prism user can start processes from any PC), [local](#) (the resource PC is only visible from the local machine), or [exclusive](#) to a particular user. In order to run more than one Resource PC on a single machine (or to run a Resource PC at the same time as running in Interactive mode), any additional Resource PC instances must be launched specifying a different port number. See the [command-line reference](#) for full details

Command Mode

In command mode, Blue Prism is launched from the command line to perform an operation on a remote (or local!) Resource PC. Once the command has been sent, Blue Prism terminates. There is no restriction on the number of simultaneous instances in command mode and command mode can be used while Blue Prism is already running in Interactive Mode, or Resource PC Mode.

Blue Prism

Command Line Options

Blue Prism provides two utilities accepting command line switches:

- **Automate.exe**

The graphical Blue Prism application. Any messages or feedback from this application is made visually. A return code of zero indicates success; a non-zero return code indicates an error.

- **AutomateC.exe**

A commandline utility which returns messages and feedback to the command line (via standard output). A return code of zero indicates success; a non-zero return code indicates an error.

Tips:

- Dynamic help is available for AutomateC, using the "/help" switch; under Automate.exe the "/help" switch will show this document, in a graphical window.
- Some switches require additional parameters, (shown below as <parameter>), which must follow the switch.
- Switches and parameters are separated by spaces. If the value for a parameter contains spaces or other special characters, it must be enclosed in "quotes". For this reason the actual value cannot contain quotes, so care must be taken to avoid this.
- When passing XML, you must enclose the xml string in quotation marks. Since quotation marks are used to delimit the start/end of the parameters xml string on standard input, Blue Prism recommends that you delimit your xml attributes using single quote marks. Alternatively, you may [escape](#) any quotation marks present, by entering two quotation marks, for each quotation mark within the parameters xml.

Examples of usage

Run Blue Prism as a Resource PC exclusive to user "admin":

```
automate /resourcepc /user admin mypwd
```

or for a [Single Sign-on](#) user:

```
automate /resourcepc /sso
```

Run Blue Prism as a public Resource PC:

```
automate /resourcepc /public
```

Run a process on the local PC:

```
AutomateC /run "My Scheduled Process" /user admin mypwd
```

or for a [Single Sign-on](#) user:

```
AutomateC /run "My Scheduled Process" /sso
```

Run a process on a remote PC with startup parameters:

```
AutomateC /run "Excel Test" /resource YourPCHostName /user admin mypwd /startp "<inputs>  
<input name='Comment' type='text' value='Hello World' /></inputs>"
```

Start background Process Alert monitoring without starting the main Blue Prism application:

```
automate /alerts /user admin mypwd
```

or for a [Single Sign-on](#) user:

```
automate /alerts /sso
```

Automate.exe - Available Switches

/help

Opens this document, in the Blue Prism html help browser.

/resourcepc

Starts Blue Prism in [Resource PC Mode](#). The [/public](#) switch can be added to make the Resource PC available to all Blue Prism users. Otherwise it will be available only to the user specified with the [/user](#) option, or the [/sso](#) option.

This option should not be confused with the [/resource](#) switch.

/public

Used in conjunction with the [/resourcepc](#) switch only. Makes the resource PC publicly available, as described in the [run_modes help](#).

/sslcert

Used in conjunction with the [/resourcepc](#) switch only, this enables transport security (SSL/TLS) on connections to the Resource PC, using the certificate with the given thumbprint, which must be installed in the user or machine's certificate store.

Enabling transport security will affect both connections from Control Room and the Scheduler, as well as exposed Web Services routed to the Resource. The CN in the certificate will need to match the address used to connect to the Resource, and the connecting client will need to trust the CA the certificate is issued by.

/wslocationprefix

(Advanced) Used in conjunction with the [/resourcepc](#) switch only, this overrides the displayed addressable location of published web services and the associated resources such as WSDLs hosted on this device. The specified value will need to contain the full path address e.g. "https://blueprismsw.ms.yorg.com:8181". Blue Prism will automatically append [/ws/](#) and associated paths to the locally available resources.

It is essential that appropriate network routing is manually applied to ensure the specified prefix is a valid address that routes to this device.


/nohttp

(Advanced) Used in conjunction with the [/resourcepc](#) switch only, this instructs the Runtime Resource to ignore any HTTP communications received on it's listening port. This option should only be used where the Resource is not required to expose any Web Services and does not need to be controlled via the [HTTP interface](#).

/dbwait

On startup, wait for the database to become available for the specified number of seconds before giving up. Connection will be retried at intervals during this period. This may be useful on small installations when running a Resource PC on startup with a database 'server' on the same machine - in this case the database may not be immediately ready to accept connections when the Resource PC starts. For example, using [/dbwait 120](#) will retry for 2 minutes before giving up.

/invisible

Used in conjunction with the [/resourcepc](#) switch only. The resource PC information form is made invisible by default. It can be shown again using the context menu in the notification area (look out for the following icon: .

/local

Used in conjunction with the [/resourcepc](#) switch only. The resource PC will only be controllable from the local machine; remote resources will not be aware of its presence.

/port

Used in conjunction with the [/resourcepc](#) switch only. Specifies the port on which Blue Prism should make itself available as a Resource PC. In this manner, you may run several Resource PCs from the same machine.

This option may be useful in specific rare situations. E.g.:

- **Overriding the default port**

If the default port of 8181 is already used for another service on your network, you may wish to use another. Be sure to use the same port for all resource PCs on your network, unless you have a specific reason not to do this.

- **Using Terminal Services**

When interacting with a [terminal services](#) host Blue Prism can only run one session at a time on each resource PC. This is due to a technical restriction of the Microsoft Terminal Services implementation. By running several resource PCs on different ports, a single machine can run several sessions as normal.

/resource <name>

This determines the target Resource PC for process alerts. If this is not specified, the local PC will be used.

This option should not be confused with the [/resourcepc](#) switch.

/user <name> <pwd>

Used to provide Blue Prism login credentials when required. When your database is configured for [Single Sign-on](#) you should use the [/sso](#) switch instead.

/sso

Instructs Blue Prism to authenticate the command based on the current user's credentials in a [Single Sign-on](#) (SSO) configuration. If your database is not configured for Single Sign-on, then you should use the [/user](#) switch instead.

For example, when using the [/run](#) switch, instead of specifying

```
Automate.exe /run Process1 /user admin pwd
```

you would specify

```
Automate.exe /run Process1 /sso
```

/useropts

Makes all Blue Prism configuration options applicable to the currently logged in operating system user. By default, the configuration applies to all users.

/setdbserver <databaseservername> /setdbname <database> /setdbusername <username> /setdbpassword <password>

Use these parameters together to create a new database connection with the same name as the given database name (see also [/dbconname](#)), or update that connection if it exists. For SQL Server Authentication, which is recommended, all four parameters must be supplied. For Windows Authentication, the username and password should be omitted. The connection is also set to be the current one. After this, Blue Prism will exit.

/ag <host> <port>

Use with [/setdbserver](#) to specify an Availability Group connection instead of a direct database connection. In this case the 'databaseservername' will be the name of the Availability Group Listener.

/agport <host> <port>

Use with [/ag](#) to specify the port for the Availability Group Listener. If not given, the default of 1433 will be used.

/multisubnetfailover <host> <port>

Use with [/ag](#) to specify that Multi Subnet Failover should be enabled.

/setbserver <host> <port>

This creates a new Blue Prism server connection with the same name as the given host. (See also [/dbconname](#), [/connectionmode](#) and [/bserversecure](#)) The connection is also set to be the current one. After this, Blue Prism will exit.

/dbconname <name>

This can be used with the [/setdbserver](#) or [/setbserver](#) options to specify a particular connection name rather than using the default. If used without one of those parameters, it specifies the name of the database connection to actually use, instead of the current one, for this session only. (i.e. the current connection is left unchanged)

/connectionmode <connectionmode>

This can be used with the [/setbserver](#) option to select the connection mode used for connection between the client and the Blue Prism Server. Note that this option must match the [corresponding value](#) set in the Blue Prism Server configuration.

Default value is 0. Available connection modes are:

- 0 - WCF: SOAP with Message Encryption & Windows Authentication
- 1 - WCF: SOAP with Transport Encryption & Windows Authentication
- 2 - WCF: SOAP with Transport Encryption
- 3 - .NET Remoting: Secure
- 4 - .NET Remoting: Insecure
- 5 - WCF Insecure

/bpserversecure False

This can be used with the /setbpserver option to disable security on the Blue Prism Server connection. This should never be used unless alternative arrangements are made to secure the connection. NOTE: the /bpserversecure option has been deprecated and the /connectionmode option should now be used instead.

/alerts

Starts background Process Alert monitoring. Needs to be used in conjunction with one of the [/user](#) or [/sso](#) switches so the correct user's [Process Alerts configuration](#) is used.

/p:<mode>

Set the Blue Prism application's priority to the given mode. Normally used in to control the priority of the Blue Prism Windows process when Blue Prism is functioning as a resource PC. This can be useful when Blue Prism needs a higher priority than other processes on the PC which may otherwise hog CPU resources and prevent Blue Prism from operating efficiently. Valid modes are:

- *below* - below normal priority
- *above* - above normal priority
- *high* - high priority
- *realtime* - the highest possible priority

/run <processname>

Used to run a process from the command line. The processname parameter must be enclosed in quotes if it contains a space. The named process must be published in order to be run (see [/publish](#)). Input parameters may be specified using the [/startp](#) parameter.

User credentials are required to run a process - see [/user](#) and [/sso](#).

/startp <paramsxml>

Used in conjunction with the [/run](#) parameter in order to supply startup parameters to a session.

The paramsxml parameter should contain valid Blue Prism parameters xml, enclosed in quotes (""). Since quotation marks are used to delimit the start/end of the parameters xml string on standard input, Blue Prism recommends that you delimit your xml attributes using single quote marks ('). Eg:

```
AutomateC /run "Excel Test" /resource YourPCHostName /user admin mypwd /startp "<inputs>
  <input name='Comment' type='text' value='Hello World' /></inputs>"
```

Alternatively, you may [escape](#) any quotation marks present, by entering two quotation marks, for each quotation mark within the parameters xml. Eg:

```
AutomateC /run "Excel Test" /resource YourPCHostName /user admin mypwd /startp "<inputs>
  <input name=""Comment"" type=""text"" value=""Hello World"" /></inputs>"
```

[Single Sign-on](#) users should substitute the [/sso](#) switch in place of the [/user](#) switch in the examples above. Eg:

```
AutomateC /run "Excel Test" /resource YourPCHostName /sso /startp "<inputs>
  <input name=""Comment"" type=""text"" value=""Hello World"" /></inputs>"
```

/showdbconfig

Displays the database configuration form. Once the user exits this form (using either the "OK" or "Cancel" buttons), the application exits.

AutomateC.exe - Available Switches**/help**

Prints some quick help tips to the command line standard output.

/createdb <dbpassword>

Creates a database, using the already configured database connection (see Automate.exe [/setdbserver](#)). The required password is the one supplied as part of the configured database connection.

This differs from the [/replacedb](#) switch in that it drops any existing database with the same name, whereas the /replacedb switch drops existing tables within the database, without dropping the database itself.

/replacedb <dbpassword>

Creates a database, using the already configured database connection (see Automate.exe [/setdbserver](#)). The required password is the one supplied as part of the configured database connection.

This differs from the [/createdb](#) switch in that if a database of the same name already exists, then it merely drops the existing tables within that database, whereas the /createdb command will drop the entire database and recreate it.

/setadmingroup <groupname>

Determines the Active Directory group for Blue Prism administrators at the time the database is created. Relevant only if you wish to use [Single Sign-on](#). Must be used in conjunction with [/setaddomain](#) and one of [/createdb](#) or [/replacedb](#).

/setaddomain <name>

Determines the Domain Name for Blue Prism administrators at the time the database is created. Relevant only if you wish to use [Single Sign-on](#). Must be used in conjunction with [/setadmingroup](#) and one of [/createdb](#) or [/replacedb](#).

/upgradedb<dbpassword>

Upgrades an existing database, if needs be. The required password is the one supplied as part of the configured database connection.

/serverconfig <name> <connection> <port>

Creates or updates the BP server configuration identified by the specified name, setting the database connection name and listening port to the provided values.

An optional [/connectionmode](#) switch can be used to set the connection mode.

If no Encryption Schemes have been configured for the BP server configuration, a new Encryption Scheme record will be created. By default the scheme will be named `Default Encryption Scheme` and it will be configured to use AES 256 (AESCryptoService) and a randomly generated key. Alternatively use the [/encryptionscheme](#) switch to define the settings that will be used.

/encryptionscheme <name> <encryptionmethod>

Used in conjunction with the [/serverconfig](#) switch, this is an optional switch that can be used to define the settings that will be used when an encryption scheme is created.

<name> defines the name of the encryption scheme.

<encryptionmethod> (optional) defines the encryption algorithm to be used.

- 1 = Triple DES (192 bit)
- 2 = AES-256 RijndaelManaged (256 bit)
- 3 = AES-256 AesCryptoService (256 bit) - default

Triple DES (192 bit) is provided for backwards compatibility. It is strongly recommended that new encryption schemes are not configured to use this method.

/refreshdependencies {force}

Requests that the object/process dependency repository be refreshed if it has been marked as out of date by a Blue Prism upgrade. If the 'force' keyword is specified then the repository will be refreshed regardless of its current state.

/license <licensefile>

Adds the license key in the specified file to the Blue Prism Database. Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

Licenses can also be managed using [System Manager](#) in the client application.

/removelicense <licensefile>

Removes the license key in the specified file from the Blue Prism database. Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

Licenses can also be managed using [System Manager](#) in the client application.

/regobject <clsid>

Registers a [COM business object](#) for use on the local machine. This is equivalent to visiting the "External Business Objects" area in [System Manager](#).

The clsid parameter refers to the COM class ID.

/regwebservice <servicename> <WSDLURL>

Registers a [Web Service](#) for consumption in Blue Prism processes. This is equivalent to registering a web service via the wizard in the "Web Services" area in [System Manager](#).

- Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.
- May optionally be used in conjunction with the [/objectname](#) switch.
- The [/timeout](#) switch may optionally be specified.
- If credentials are required in order to retrieve the WSDL document, then the [/wsauth](#) switch may be used.

The <servicename> identifier must match the name of a service which appears in the Web Service's WSDL document (see the [Web Services Glossary](#) for more information about WSDL documents). This is required because one WSDL document can describe several different web services. The name provided is distinct from that given using the [/objectname](#) switch, which specifies the name by which the web service will be known internally, within your Blue Prism processes.

The <WSDLURL> parameter must specify the location (URL) of a WSDL document for the web service of interest. This will be retrieved using either http or https, as appropriate.

/unregwebservice <servicename>

Unregisters a [Web Service](#). This is equivalent to deleting a web service via the wizard in the "Web Services" area in [System Manager](#).

- Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

The <servicename> identifier must match the name by which the web service is known internally, within your Blue Prism processes.

/objectname <objectname>

Used in conjunction with the [/reqwebservice](#) switch to specify the business object name that will be given to the web service, once registered.

/wsauth <username> <password>

Used in conjunction with the [/reqwebservice](#) switch to specify any user credentials to be used in the http request when downloading the WSDL document. The use of this switch is only necessary when the WSDL is password-protected by the hosting server.

/timeout <milliseconds>

Used in conjunction with the [/reqwebservice](#) switch to specify the timeout value to be associated with the web service, (measured in milliseconds). This will be the timeout applied at runtime when waiting for a response from the web service. It may later be modified by visiting [System Manager](#) and editing the web service.

The use of this switch in conjunction with the [/reqwebservice](#) command is optional. If not used then the default value of 10,000 milliseconds will be applied.

/report <filespec>

Writes a system report out to the specified file. The system report is a broad overview of the current state of the system - the number of sessions and log entries, the number of queue items, the validation state of the processes etc.

/elementusage <filespec>

Writes an element usage report out to the specified file, in CSV format. Use the [/process](#) option to specify the Business Object to report on. The report will contain details of the Page and Stage where each element from the Application Model is used.

/import <filespec>

Imports a Blue Prism process (or [visual business object](#)) into the database, by default using the ID found in the file (if one exists - otherwise a new ID is generated). See [/forceid](#) to override this behaviour.

The filespec parameter refers to location of the xml file to be imported.

By default, if the imported process / object already exists in the selected environment, this operation will fail. See [/overwrite](#) to avoid this.

/forceid {new|<guid>}

Use with [/import](#) to force use of the given ID for the process, instead of using the one defined in the file being imported.

The 'new' keyword will generate a new ID for the imported object, otherwise a valid GUID should be provided.

/overwrite

By default, an [/import](#) will fail if the process / object being imported already exists on the database.

If the [/overwrite](#) option is used, this will import a new version of the process / object over the top of the currently existing one.

Note: If a different process / object exists with the same name as the imported process / object, but with a different ID, the import operation will fail - /overwrite works only if a process / object with the same ID is found.

To deal with this case the [Import Wizard](#) can be used from within the product.

/importrelease <filespec>

Imports a Blue Prism release file into the database. This action is functionally the same as performing the import interactively via the Blue Prism client (see [Release tab](#)), except that it is non-interactive and any conflicts are handled automatically where possible. In general, the default action will be to **Overwrite** where the item being imported already exists within the database. However, some conflicts will prevent a release from being imported non-interactively. The following table summarises the handling of conflicts:

Conflict	Outcome
Item with same ID/Name and type already exists in database	The item will be overwritten
Process requires publishing or retiring after import	The process will be published/retired
The following are examples of conflicts which would result in a failure to import:	The release will not be imported
<ul style="list-style-type: none"> User does not have relevant database permissions to import a tile datasource Credentials cannot be imported because a Default encryption scheme has not been configured An Object in the release has the same name (or ID) as an existing Process in the database A Process in the release has the same name (or ID) as an existing Object in the database 	

Note: When importing new credentials interactively, the user is prompted to enter the username and password (as this information is not written to the export file). When importing non-interactively, the credentials will be imported with a blank username/password and will require updating manually in the Blue Prism client.

The filespec parameter refers to location of the `.bprelease` file to be imported.

/export <processname>

Exports a Blue Prism process (or [visual business object](#)) from the database to the local filesystem.

/archive

Archives session logs to files. See [Archiving](#) for more details. With no options, all session logs will be archived. To limit the range of logs archived, use either the [/age](#) option, or a combination of the [/from](#) and [/to](#) options. You can also use [/process](#) to limit the archiving to logs for a particular process.

To delete the session logs, without exporting them, you can add the [/delete](#) option.

/restorearchive

Restore session logs from files to the database. See [Archiving](#) for more details. With no options, all session logs will be restored. To limit the range of logs archived, use either the [/age](#) option, or a combination of the [/from](#) and [/to](#) options.

/setarchivepath <path>

Set the archiving path. This is the path used by the [/archive](#) and [/restorearchive](#) commands, as well as in the System Manager interface.

/from <yyyyMMdd>

Used with [/archive](#) and [/restorearchive](#), this sets the inclusive start of a date range.

/to <yyyyMMdd>

Used with [/archive](#) and [/restorearchive](#), this sets the non-inclusive end of a date range.

/age <value>

Used with [/queueclearworked](#) this deletes all worked items which are older than the specified age (this includes items marked as an exception before this time). Used with [/archive](#) and [/restorearchive](#), this bypasses date-range-based selection and selects sessions older than a particular age. The value is a number followed by one of 'y' (years), 'm' (months) 'w' (weeks) or 'd' (days). For example, "6m" specifies that logs older than 6 months should be archived.

/process <name>

Used with [/archive](#), this limits the archiving to sessions that ran the named process. Used with [/elementusage](#), this selects the process to generate a report on.

/delete

Used with [/archive](#), causes the logs to be deleted without exporting them first.

/publish <processname>

Publishes the named Blue Prism process. If the process' name contains spaces then you will need to enclose the name in quotes.

/unpublish <processname>

Unpublishes the named Blue Prism process. If the process' name contains spaces then you will need to enclose the name in quotes.

/publishws <name> [/forcedoclitencoding] [/useGlobalNamespace]

Exposes the named Blue Prism object or process as a web service, for consumption by external applications. If the object or process name contains spaces then you will need to enclose the name in quotes.

If the optional [/forcedoclitencoding](#) is present, then the web service will always return SOAP messages in Document/Literal form for WS-I compliance.

If the [/forcedoclitencoding](#) parameter is not present, then you can force the use of a Global Namespace.

/unpublishws <name>

Conceals a Blue Prism object or process that had previously been exposed as a web service. If the object or process name contains spaces then you will need to enclose the name in quotes.

/status <sessionid>

Gets the status of a running session. SessionID is a unique identifier in the form xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx. Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

/getlog <sessionid>

Gets the log of the specified session. SessionID is a unique identifier in the form xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx. Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

The log is written to standard output, line by line. There is a data limit of $2^{32} - 1$ rows; however this limit is irrelevant in practice, corresponding to hundreds of gigabytes of data.

/dbconname, /user, /sso, /resource, /port, /startp

The usage for these commands under AutomateC.exe, corresponds exactly to the usage under Automate.exe. See above for details of each ([/dbconname](#), [/user](#), [/sso](#), [/resource](#), [/port](#), [/startp](#)).

/requeststop <sessionid|sessionnumber>

Sends a 'Stop Request' to the specified session, setting a flag that the running process can check using the `IsStopRequested()` function in a decision stage.

One of the session ID (a GUID) or the session number (a number) must be provided to indicate the session which should be flagged with the stop request.

Note that this can only be executed by an authenticated user (provided using the [/user](#) or [/sso](#) switches) with the Full Access to Session Management permission.

/createqueue <keyfield> >running< <maxattempts>

Creates a [work queue](#) with the [keyfield](#), [running](#) and [maxattempts](#) values, as would be required when creating a queue by hand in System Manager.

Must be used in combination with [/queueName](#). Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

/setencrypt <encryption-scheme-name>

Sets the encryption scheme that will be used by the specified queue. The specified scheme must be configured within this environment to succeed. The default scheme name, set in the configuration of the [Blue Prism Server](#), is referred to as `Default Encryption Scheme`.

To disable the encryption key of a queue, see the [/resetencrypt](#) switch.

Must be used in combination with [/queueName](#). Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

/resetencrypt

Resets the encryption scheme on the specified queue such that it is no longer encrypted.

Must be used in combination with [/queueName](#). Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches.

/reencryptdata [/batchsize <size>] [/maxbatches <max>]

This option re-encrypts any credentials, resource screen captures or queue item data that is not encrypted according to the currently selected scheme, and could be used for example when revoking an old key prior to deleting an [Encryption Scheme](#). The parameter `/batchsize` controls how often updated records are committed to the database (default is 1000), and `/maxbatches` controls how many batches are processed (default is 1).

Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches, and access to the `Security - Manage Encryption Schemes` permission.

/exportqueue <filespec>

Exports data from a [work queue](#), and stores it at the location specified by `<filespec>`. Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches. Must be used in combination with [/queueName](#). May be accompanied by one or both of [/queuefilter](#) and [/clearexported](#).

Note that `/exportqueue` requires user credentials to be supplied via one of the `/user` or `/sso` switches, and access to one of the following permissions: `Full Access to Queue Management`, `Read Only Access to Queue Management`

/queueclearworked /queueName <name> [/age <age>]

Clears worked cases (i.e. cases marked as completed or exception) from a [work queue](#). Requires user credentials to be supplied via one of the [/user](#) or [/sso](#) switches. Must be used in combination with [/queueName](#).

If the `/age` switch is used only worked items older than the specified age will be deleted from the queue, otherwise all worked items will be deleted from the named queue.

You may wish to use the [/exportqueue](#) command in combination with the [/clearexported](#) instead, as it allows you to generate a record of which items were deleted in the process - this command creates no such record.

/queueName <queueName>

Specifies the name of a queue, when used in conjunction with a queue-related command, such as [/exportqueue](#) or [/createqueue](#).

/queuefilter

Specifies the name of a filter to use when exporting a queue with [/exportqueue](#). This is the name of a filter created in the "Queue Management" area of Control Room.

/clearexported

When used in conjunction with [/exportqueue](#), causes the worked and referred items exported to be deleted from the queue.

Requires user credentials to be supplied via one of the `/user` or `/sso` switches, and access to the `Full Access to Queue Management` permission.

/createcredential <name> <username> <password> [/description <string>] [/expirydate <date>] [/invalid <flag>]

Creates a new credential using the specified name, password and username.

Requires user credentials to be supplied via one of the `/user` or `/sso` switches, and access to the `Security - Manage credentials` permission.

The credential created will be universally available to all user roles, resources and processes by default.

/updatecredential <name> [/username <username>] [/password <password>] [/description <string>] [/expirydate <date>] [/invalid <flag>]

Updates an existing credential found using the specified name.

Requires user credentials to be supplied via one of the `/user` or `/sso` switches, and access to the `Security - Manage credentials` permission.

/password <password>

Use with `/updatecredential` to update the credential password.

/username <username>

Use with `/updatecredential` to update the username held in the credential.

/description <text>

Use with `/updatecredential` or `/createcredential` to set the text held as the description for the credential.

`/expirydate <date>`

Use with `/updatecredential` or `/createcredential` to set the date upon which the credential will expire. Date must be in format `yyyyMMdd`.

`/invalid <flag>`

Use with `/updatecredential` or `/createcredential` to specify if a credential should be marked as invalid.

`/setcredentialproperty <credname> <propertyname> <propertyvalue>`

Create a new credential property or update it if it already exists. Requires user credentials to be supplied via one of the `/user` or `/sso` switches, and access to the Security - Manage credentials permission.

Scheduler options

You can start and delete schedules and view reports and timetables using the `automatec` command line program. To perform any of the schedule actions, valid login details must be provided.

`/schedule <name|...>`

Specifies the name or names of the schedules to be used in conjunction with the [/startschedule](#), [/deleteschedule](#), [/viewschedtimetable](#) and [/viewschedreport](#) actions. Any parameters following a `/schedule` switch will be treated as schedule names until another switch is reached (ie. a parameter starting with a `"` character) or the end of the command is reached. When no schedules are specified, the commands are generally treated as running on all schedules.

`/startschedule [/schedule <name|...>]`

Initiates the specified schedule at the current time. If the scheduler is running it should execute the schedule within 30 seconds.

`/deleteschedule [/schedule <name|...>]`

Deletes the specified schedule and any related schedule logs.

`/format {txt|csv}`

Specifies the output format of the schedule report or timetable, either `'txt'` (human readable) or `'csv'` (comma-separated-variable). The default is `'txt'` if no format is explicitly specified.

`/viewschedtimetable {<name> | <no-of-days> <date>} [/schedule <name|...>] [/format {txt|csv}]`

Outputs the specified timetable. You can specify a named timetable which has been [created and configured](#) in the Blue Prism client. Alternatively, you can specify an ad hoc timetable by indicating the date from which the timetable should run, the number of days to show and the schedules to include.

`/viewschedreport {<name> | <no-of-days> <date>} [/schedule <name|...>] [/format {txt|csv}]`

Outputs the specified report. You can specify a named report which has been [created and configured](#) in the Blue Prism client. Alternatively, you can specify an ad hoc report by indicating the date up to which the report should include, the number of days back to show and the schedules to include.

Resource Pool options

You can create and delete resource pools and add and remove resources from existing pools using the `automateC` command line program. To perform any of the resource pool actions, valid login details must be provided.

`/poolcreate /pool <name>`

Creates a resource pool with the specified name.

`/pooldelete /pool <name>`

Deletes the specified resource pool.

`/pooladd /pool <name> [/resource <name>]`

Adds a resource to the specified pool. If no resource name is specified, the local resource is added to the pool.

`/poolremove [/resource <name>]`

Removes a resource from the pool in which it resides. If no resource name is specified, the local resource is removed from its pool.

Help

Connections Dialog

The connections dialog allows you to store different database connection configurations. All connections require a name, and depending on the connection type other information is required.

Connection Type

From the dropdown you can choose between five connection types:

- Blue Prism Server
- SQL Server (SQL Authentication)
- SQL Server (Windows Authentication)
- Availability Group (SQL Authentication)
- Availability Group (Windows Authentication)

Blue Prism Server

Connection to a dedicated Blue Prism Server is the most secure connection configuration.

See the [Blue Prism Server](#) page for details on how to configure and run a server.

Server Name

You must supply the hostname of the machine on which your Blue Prism Server is running. The local machine must be able to resolve the supplied hostname.

Connection Mode

This allows you to select the connection mode that is used for connections between the client and the Blue Prism server. Note that this setting must match the [corresponding value](#) set in the Blue Prism Server configuration.

Port

You must supply the port on which your Blue Prism Server is listening.

Callback Port

Specifies a port that the client will listen on for callbacks from the server. This can be left set to 0 to allow an available port to be chosen automatically. Note that if you use this option to force a specific port, only one instance of any Blue Prism application will be able to connect to the server from that machine. This setting is only applicable to Blue Prism Server connection using a .NET Remoting connection mode.

SQL Server (SQL Authentication)

SQL Authentication uses credentials of an SQL user to access the database.

Server Name

You must supply the hostname of the machine on which your database server is hosted. The local machine must be able to resolve the supplied hostname.

Database Name

The name of your Blue Prism database as it appears on the database server. If no such database exists yet then you may choose a database name here. You may then create a database via the Blue Prism user interface as described on the [Create Database](#) help page.

Database User ID

Supply here the username of the database user that Blue Prism should use when accessing the database.

Database User Password

Supply here the password of the user named in the field above. This password will never be revealed by Blue Prism.

Additional SQL Connection Parameters (non-BP Server only)

Any further semicolon-separated parameters which should be appended to the query string created to access the database. *eg.*
`Encrypt=true;TrustServerCertificate=true.`

SQL Server (Windows Authentication)

Windows Authentication uses the currently logged on windows user credentials to access the database. Access must be granted on the database server to allow the Windows user to access the database.

Server Name

You must supply the hostname of the machine on which your database server is hosted. The local machine must be able to resolve the supplied hostname.

Database Name

The name of your Blue Prism database as it appears on the database server. If no such database exists yet then you may choose a database name here. You may then create a database via the Blue Prism user interface as described on the [Create Database](#) help page.

Availability Group

The two Availability Group options are the same as the corresponding SQL Server options, but connect to a SQL Server Availability group. The Server Name refers to the Availability

Group Listener, and two extra settings are available:

Port

The port the Availability Group Listener is listening on.

Multi Subnet Failover

Allows multi subnet failover to be enabled on the connection. Refer to Microsoft's documentation on this for more information.

Using Multiple Connections

You may wish to store more than one connection - for example one database for testing and another for live. This dialog allows you to move quickly from one connection to another. New connections are added using the "New Connection" button. The details of the new connection are then filled out on the right hand side.

To edit an existing connection, simply select it from the list on the left and fill out the details on the right. Whichever connection is selected when the "OK" button is pressed will be the active connection used by Blue Prism.

For security purposes, only the password of the current connection is stored, so you will have to retype the password when you switch from one connection to another.

Help - Blue Prism Resource PC Commands

Overview

This document contains a list of commands available when communicating with the Blue Prism Resource PC listener. These commands are used by Blue Prism for its network communications, and are available to third party clients via an HTTP interface.

Recommended usage

When learning these commands, it is instructive to try them out as you read this document. You may do this as follows:

- Start a Blue Prism Resource PC listener using the command:

```
automate.exe /public /resourcepc
```

- Connect to the listener, using telnet:

```
telnet localhost 8181
```

- (Tip: if for whatever reason it is not convenient to use the (default) listener port of 8181, you may start the listener on a different port using the command line switch "/port". In such cases you should amend the port of your telnet connection accordingly).
- Type the command "quit" to close the connection.

Command Reference

createas

Authorisation: Authed

Uses token authentication to either create a pending session with given process ID: 'createas <token> <procid>', create pending session for given queue with given process ID: 'createas <token> <procid> <queueid>' or create a pending session using process name: 'createas name <token> <name>'. The response is 'SESSION CREATED : <sessionid>' or an error code.

deleteas

Authorisation: Authed

Uses token authentication to delete a pending session: 'deleteas <token> <sessionid>'. The response is 'SESSION DELETED' or an error code.

startas

Authorisation: Authed

Uses token authentication to either start a session running: 'startas <token> <sessionid>' or start the last session created on this connection: 'startas last <token>'. The response is 'STARTED' or an error code.

getauthtoken

Authorisation: Any

Get an authorisation token using the provided credentials. Use 'getauthtoken <userid> <password>' This method is available for custom solutions designed for secured networks. It is not used by Blue Prism in normal operation.

action

Authorisation: Authed

Either 'action <sessionid> <actionname>' to run an action on the specified session, or 'action last <actionname>' to run an action on the last session created on this connection. The response is 'STARTED' or an error code.

availability

Authorisation: AuthedOrLocal

Reports the current availability of this Resource PC for running more processes. The response is 'AVAILABILITY:level' where 'level' is one of the following four values:

None - indicates that no process can be run at the moment, because an exclusive process is already running (or pending).

Background - indicates that only a background process can be run, because there is already a foreground process running (or pending).

Foreground - indicates that either a background or foreground process can be run, because there are no foreground or exclusive processes running (or pending).

Exclusive - indicates that any kind of process can be run (including an exclusive process), because there are no processes running (or pending).

busy

Authorisation: AuthedOrLocal

Check if there are there any sessions pending/running? Responds yes or no.

caps

Authorisation: Authed

Get resource capabilities

connections

Authorisation: AuthedOrLocal

Get a list of active connections involving this Resource PC. The list is separated into two sections, labelled INBOUND and OUTBOUND. The inbound list shows connections into this Resource (as remote address and authenticated username) and the outbound list shows connections this Resource has open to other resources (which will be none, unless the Resource is a pool controller).

controller

Authorisation: Authed

Signifies that this is a connection from the controller of a pool. This command is used internally between Resource PCs only.

create

Authorisation: Authed

Either 'create <procid>' to create pending session with given process id, 'create <procid> <queueident>' to create pending session for given queue with given process id or 'create name <name>' to create pending session using process name. The response is 'SESSION CREATED : <session id>' or an error code. Use createas instead when the Session Management enforces permissions of controlling user setting is enabled

delete

Authorisation: Authed

Delete a pending session - 'delete <sessionid>'. Use deleteas instead when the Session Management enforces permissions of controlling user setting is enabled

getparams

Authorisation: Authed

Use 'getparams <sessionid>' to get the startup parameters used to start a process. The process must have already been started, but it does not necessarily have to be on the Resource from which the request is made. The response is PARAMS: followed by the parameters in XML format, NONE if there were no parameters, or otherwise an error description.

internalauth

Authorisation: Any

Use 'internalauth <token>' to perform token-based authentication. This uses a single-use token generated and registered with the database to confirm identity of the connecting user, to mitigate sniffing and replay attacks, and to allow direct connection in an AD environment. Blue Prism software only ever uses this method. Non-Blue Prism software can use AutomateC to generate a token.

members

Authorisation: AuthedOrLocal

List the members of the Pool for which this Resource PC is the controller. Example response:

```
MEMBERS - 2
9efb8f78-0777-4d28-aaa3-6fbd5da70f36 PC1
21e960d0-140a-488c-b09d-d3d7257bae54 PC2
```

The reported values are the ID and Name of the Resource.

outputs

Authorisation: Authed

Use 'outputs <sessionid>' (or 'outputs last') to get the output parameters from a process or action that has completed successfully, or the failure reason from one that has failed. When running Business Object Actions, this also sets the object back to idle state, and must be used before another Action is started. The response is 'OUTPUTS:' followed by the output parameters of the completed session, in XML format.

password

Authorisation: Any

Use 'password <pwd>' to complete authentication after a 'user' command. See the documentation for 'user' for more information.

ping

Authorisation: AuthedOrLocal

Returns the message 'pong'. Used for diagnostics.

pool

Authorisation: Authed

Determine which pool this resource is a member/controller of. The response will either be 'Not in a pool', 'Controller of <pool name>' or 'Member of <pool name>'.

proclist

Authorisation: Authed

Provides a list of processes which are available to be run. Output is in the form of one process per line, formatted as follows:

```
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx - Process One
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx - Process Two
```

The value in the first field is the unique identifier of the process. This is used in conjunction with the 'create' command to create a pending session for a process.

proxyfor

Authorisation: Authed

Sets the userid that the pool member will run the session as

quit

Authorisation: Any

Terminate this session. The client is disconnected. The server continues to run.

setvar

Authorisation: Authed

setvar <sessionid> [<varname>] type "<value>" "<description>"

Sets the value of a session variable within a session. If the operation is successful, the response is 'SET' - otherwise an error code.

shutdown

Authorisation: AuthedOrLocal

Shut down the Resource PC. Use optional parameter 'waitforsessions' to wait for any currently running sessions to complete before shutting down. Use the 'loginagent' parameter to ignore the shutdown request if the Resource PC is not used by the Login Agent.

start

Authorisation: Authed

Either 'start <sessionid>' to start a session running, or 'start last' to start the last session created on this connection. The response is 'STARTED' or an error code. Use startas instead when the Session Management enforces permissions of controlling user setting is enabled

startp

Authorisation: Authed

Set startup parameters to be used for the next session created on this connection. The parameters are given in XML format. The response is 'PARAMETERS SET'.

status

Authorisation: AuthedOrLocal

Retrieves a list of pending and running sessions on the Resource PC. Output is in the form of one session per line, formatted as follows:

```
RESOURCE UNIT
- PENDING 9efb8f78-0777-4d28-aaa3-6fbd5da70f36
- PENDING 21e960d0-140a-488c-b09d-d3d7257bae54
Total running: 2
```

The value in the last field is the unique identifier of the session.

stop

Authorisation: Authed

Use 'stop <sessionid>' to stop process running. Optional extra parameters '<userid> <resourceid>' may be specified as a pair if this information is to be logged against session. The resourceid may be replaced by 'name <resource name>'. The response is 'STOPPING' or an error code.

user

Authorisation: Any

Start username/password-based authentication. Use either 'user <userid>' to set user id, or 'user name <name>' to set user using user name. This must then be followed by the 'password' command to complete the authentication sequence. This authentication method is available for diagnostics and custom solutions designed for secured networks. It is not used by Blue Prism in normal operation.

userlist

Authorisation: Authed

List all users. Users are displayed one per line, in the following format:

```
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx - User One
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx - User Two
```

The value in the first field is the unique identifier of the user.

vars

Authorisation: Authed

Gets all the current session variables and their values for the given session ID. Example response: VARIABLES:2

```
[CasesWorked] number 212
[Exceptions] number 4
```

varupdates

Authorisation: Authed

Enable or disable session variable updates. Specify either 'on' or 'off'. The response is 'SET' or an error code.

version

Authorisation: Any

Get version information - the response is the internal protocol version number.

Help - Resource PC HTTP Interface

Overview

This document describes mechanisms for controlling a Blue Prism Resource PC using HTTP.

HTTP Mechanisms - overview

Communication can be via either [HTTP GET](#) or [HTTP POST](#). Each mechanism simply wraps the basic [resource PC commands](#), which can be issued over a telnet interface.

HTTP GET Requests

Commands can be issued in the form of a URL. The command of interest (together with the arguments) appears as if it were the name of a document to be retrieved. For example the URL:

```
http://localhost:8181/create%20name%20MyProcess
```

corresponds to the command:

```
create name MyProcess
```

Note that special characters must be escaped - eg the space character must be written as %20. Any non alphanumeric characters in a process name for example will also have to be escaped. This can be achieved in javascript using the "escape()" function, or in the .NET framework using the "System.Web.HttpUtility.UrlEncode()" function.

Multiple commands can be specified inline. For example, you may authenticate yourself (using the [user](#) and [password](#) commands) and start a process (using the [run](#) command) all in one go, as follows:

```
http://localhost:8181/user%20admin&password%20MyPassword&create%20name%20My%20Brilliant%20Process
```

Note that commands are separated by the & character.

HTTP POST Requests

Commands can be issued as parameters within an HTTP POST request. For example the following message posted to "/automate" performs authentication and starts a process, just as the HTTP GET example above does:

```
POST / HTTP/1.1
Host: localhost:8181
User-Agent: Mozilla/4.0
Content-Length: 92
Content-Type: text/plain

param1=user%20admin&param2=password%20MyPassword&param3=create%20name%20My%20Process
```

The naming of parameters is not important, because the Resource PC listener ignores them.

Help - Single Sign-on

Overview

Blue Prism provides two methods of managing authentication to the platform:

- Native Blue Prism Authentication
- Single Sign-on for Blue Prism (recommended for enterprise deployments)

The choice of authentication scheme must be selected when the database is created - it cannot be changed afterwards. [Connections](#) can be configured to different databases, and each database can implement a different Blue Prism sign-on method.

Native Blue Prism Authentication

By default Blue Prism uses its own authentication mechanism. User accounts are individually created and maintained within Blue Prism and user login attempts are processed by verifying the supplied username/password combination configured in the Blue Prism database. The individual permissions and roles of users are maintained by assigning Blue Prism [user roles](#).

Single Sign-on for Blue Prism

Blue Prism supports Single Sign-on using Microsoft Active Directory Domain Services which allows users who have been authenticated by the operating system, and who are members of appropriate domain security groups, to log into Blue Prism without re-providing their credentials.

Single Sign-on benefits system administrators by giving them a single point of management and access control for large numbers of users.

Configuring Single Sign-on for Blue Prism

Blue Prism's implementation of single sign-on applies access controls to user accounts based on their Active Directory security group membership.

When configuring single sign-on authentication for Blue Prism it is necessary to specify the Active Directory domain where the security groups that will be associated with Blue Prism security roles will reside. Additionally the security group whose members will be granted System Administrator access must be selected.

Once the system administrators have been configured with access, the [mapping](#) between the other Blue Prism security roles and Active Directory security groups can take place.

Trouble shooting

If you experience trouble, see the [Single Sign-on troubleshooting page](#).

Help

Creating a new Database

1. Set Connection Information

First visit the [Database Connections Form](#) to enter details of the server on which you would like to create a database, and to enter the name that the new database should take.

2. Ensure that the database is not in use

Blue Prism will create a database with the name supplied in the connections form (or modify an existing database sharing that name, if such a database already exists). If a database with that name does already exist then it is important to ensure that it is not in use by another application. This includes other instances of Blue Prism running on other resource PCs. Please ensure that no other application is currently using the database before proceeding.

3. Choose whether to purge any existing database

If a database with the specified name already exists, you may either modify the existing instance or you may purge (ie permanently and thoroughly delete) the existing database and create a new database of the same name in its place. The latter is the recommended option.

The former option may be useful if the database username under which Blue Prism runs does not have permission to DROP and CREATE databases. If no database already

exists with the specified name then choosing the "purge" option will have no effect.

4. Choose Authentication Method

A choice of standard Blue Prism Authentication, or [Single Sign-on](#) Authentication exists. Single Sign-on is usually only used by large organisations with a well managed network. This option will require the assistance of a network administrator to configure Microsoft Active Directory options.

If unsure, choose the default option of Blue Prism Authentication.

5. Create the Database

From the main menu of Blue Prism, choose the "Create Database" option. It is necessary to be signed out of Blue Prism in order to do this. On the form that pops up, confirm the password given in the connections dialog, click "OK" and wait. Once the database has been created you will receive a confirmation message. From this point forward you may sign in to the new database as described below.

6. Sign in

If you successfully configured Single Sign-on then you should be signed in automatically. If you have trouble, check your settings (including group Active Directory group membership, etc) and try again. Note that Active Directory changes may not take effect until you log off and log on again.

If you are using Blue Prism authentication then you may sign in to the new database using the username "admin" and the password "admin". It is highly recommended that you change the password on this administrator account in order to prevent unwanted access.

One of your first tasks after signing in may be to:

- [Create user accounts](#) for other users of the system.
- [Manage user roles and permissions](#) for the newly created users. This is particularly relevant if you are using Single Sign-on.
- [Import a process/release](#) — either as a learning example or a functioning entity for use against your systems.

Help

Java Access Bridge - Setup Instructions

Overview

There are two methods of installing the [Java Access Bridge](#) (JAB): automatic and manual. The former is the preferred method and will suffice in the majority of situations; the latter is described here for reference.

The manual method may be required for use with non-standard installations of the Java Runtime Environment (JRE). Such installations are sometimes used by vendors who want to maintain control over the environment in which their software runs, or by vendors who have modified the JRE for a special purpose.

Prerequisites

The Java Runtime Environment (JRE) with which the Java Access Bridge (JAB) is to be used must be installed first.

Requirements

At the time of writing the latest version of the Java Access Bridge (JAB) is 2.0.1. However this version contains a critical fault, and is not suitable for use with Blue Prism. JAB version 2.0.0 is recommended for use with JRE version 1.2.2 and above. A copy of JAB 2.0.0 is available from Blue Prism on request. Please consult Blue Prism for advice if your JRE version is lower than 1.2.2.

Automatic Installation

To begin the automatic installation, run the installation executable and follow the on-screen instructions. When complete, reboot your PC.

Manual Installation

Only use the manual installation for non-standard JREs; the automatic installer will locate and amend any standard JREs as required.

1. Extract files

Extract the contents of the installation archive `accessbridge-2_0_0-manual_install.zip` into a temporary directory.

2. Copy system files

Copy the files `JavaAccessBridge.dll`, `JAWTAccessBridge.dll` and `WindowsAccessBridge.dll` into the windows system directory, located at:

```
c:\windows\system32\
```

Note that these files may already be present if you have run the automatic installer. There is no need to replace them in this case (unless they correspond to an early version of the Java Access Bridge such as version 1.0).

3. Locate Target JRE

First locate your JRE installation(s). You may install the Java Access Bridge (JAB) for use with more than one JRE installation at once. Indeed this may be necessary as well as desirable since products such as the official Sun Microsystems™ Java web browser plugin install a new separate JRE. Thus in order to be able to automate java applications embedded in a web page, you will need to install the JAB in this JRE (as well as any others you choose). To install into multiple JREs, repeat the instructions below for each JRE installation directory.

Standard installations of the JRE are usually found in a directory such as:


```
c:\Program Files\Java\jdk1.5.0\jre\  
c:\Program Files\Java\jre1.5.0_05
```

Non-standard installations of the JRE are usually found in a subdirectory of the directory in which the target application was installed. Eg:

```
c:\Program Files\MySpecialApplication\JRE\
```

4. Locate the extensions directory

Navigate to the folder lib\ext under your installation directory. Eg:

```
c:\Program Files\Java\jdk1.5.0\jre\lib\ext  
c:\Program Files\Java\jre1.5.0_05\lib\ext
```

5. Copy Java files

Copy the file named access-bridge.jar into the "ext" directory identified in the previous step.

Please identify the version of the target JRE installation: a value of the form 1.X. Copy into the "ext" directory the file named jaccess-1_X.jar (where X is the number identified as part of the JRE version), and then rename it to jaccess.jar

Where there is no matching file use the first available file with a lower version number instead, (eg in the case of 1.5 or 1.6, for which there is no matching file, please use the file jaccess-1.4.jar).

6. Create properties file

Check for the existence of a file named accessibility.properties in the "lib" folder (one level above the "ext" folder), and create it using a text editor if necessary. Be sure to spell its name correctly. If the file does not yet include the following line then append it to the end of the file, on its own line:

```
assistive_technologies=com.sun.java.accessibility.AccessBridge
```

It is perfectly acceptable for this line to be the only line in the file.

7. Cleanup

Delete the temporary directory into which the contents of the installation archive (accessbridge-2_0_0-manual_install.zip) were extracted.

Reboot your PC.

Notes

Each time the JRE in question is upgraded or modified (eg for a security update, issued by Sun Microsystems™, the creator of the Java Runtime Environment), the Java Access Bridge installation will have to be repeated in order to make sure that the new JRE has the correct extensions in place to work with the Access Bridge.

Each time a new JRE is installed on your machine (for example for use with a new piece of software, it may be necessary to repeat the installation of the Java Access Bridge, even for your existing JREs).

Java Automation

This feature is subject to licensing restrictions. Please see the [licensing page](#) for further information.

Overview

Java automation provides a series of specialised techniques for interfacing with applications written in the [Java programming Language](#). To take advantage of these features, you should specify that your application is of type Java when running the [Application Modeller Wizard](#). Alternatively, Java integration techniques are also available from within [browser applications](#) when the [appropriate application parameter](#) is selected in the Application Modeller Wizard.

Requirements and Installation

Java Automation requires the [Java Access Bridge](#) to be installed. At the time of writing, the latest version available separately is 2.0.2. The most recent versions of the JRE include the Access Bridge as standard.

Blue Prism supports JAB versions 2.0.0 and 2.0.2 and beyond. Version 2.0.1 has a flaw causes issues working with Blue Prism, so avoid using this version.

Please consult the [Java Access Bridge Installation Guide](#) for details of how to install the Java Access Bridge for use with Blue Prism.

Usage

Java applications are accessed in the same way as any other application in [Object Studio](#), using the spy tool from the [Application Modeller](#). See the [list of java attributes](#) returned by the spy tool for more information about identifying elements within your target application.

- Customers using a 32-bit operating system are able to use Java access bridge 2.0.0 or higher and applications can be set to launch in embedded or external 32-bit mode
- Customers wishing to utilise 64 bit operating systems must use Java access bridge 2.0.2 or higher
- Business Objects that model 64-bit Java applications must be set to "External 64-bit" mode

Limitations

Sun's Java Access Bridge has limited support for version 1.3.x of the Java Runtime Environment (JRE). In particular text cannot be written directly to an edit field. However there are

workarounds for each limitation, meaning that Java Automation continues to be a valuable tool for JRE 1.3. See also [help topic 32808](#).

The Java Access Bridge does not allow interaction with modal dialogs. This fact is documented on Sun's [mailing list](#). However, Blue Prism provides several other tools which allow interaction with modal dialogs, including bitmap recognition, font recognition, and the use of regions for basic user interaction (such as clicking of the mouse).

Troubleshooting

There are [troubleshooting tips](#) available.

Performance Tips

- When using 'descendtree' mode, significant performance gains can be achieved by using the AncestorCount, MatchIndex and MatchReverse identifiers to limit the scope of a query.

Blue Prism Help - Process Alerts

This feature is subject to licensing restrictions. Please see the [licensing page](#) for further information.

Overview

Process alerts allow users to monitor the progress of running sessions, without the need to continually monitor the display in [Control Room](#). Interested parties can receive information about sessions alerting them to changes in the session status (such as when the session begins to run, when it completes, and, when it fails).

Designers of Blue Prism Process [flow charts](#) can create custom alert notifications by adding an [Alert Stage](#) to the flow diagram. This allows alerts subscribers to receive custom notifications such as "New case batch downloaded; beginning processing on case 1 of 300". The content of such messages is based on the data available in the process, and may be changed dynamically, using a [Blue Prism Expression](#).

Alerts Customisation

Different users can choose to subscribe to alerts for different processes, according to their interests. They can also filter the types of message that they wish to receive (for example if they are only interested in knowing when a session has completed then they can choose to filter out all other alert types). Several user interface options are available, and each user may choose their preferred format independently of all other users. Users change their preferences by editing their [Alerts configuration](#).

Alert Monitoring

Process alerts can be monitored from the main Blue Prism window, or by running Blue Prism in the background. When a user logs into Blue Prism, alert monitoring will also start (if the user has subscribed to process alerts) and an icon will appear in the Windows taskbar. When the user logs out, the user will have the option to keeping monitoring running. To run Blue Prism in the background, use the command line option as described below.

Command Line

Process alert monitoring can be started from the command line using the following syntax.

```
Automate /alerts /user myname mypassword
```

This option will create a taskbar icon to indicate that Blue Prism is running in the background. The icon features a menu which allows the user to configure their alert settings and to make the main Blue Prism window visible, if desired.

Help

This feature is subject to version restrictions. Please see the [version page](#) for further information.

Exceptions

Exceptions occur during the running of a process for various reasons. There may be, for example:

- A mistake in the process itself (e.g. an invalid expression entered into a calculation)
- Failure to match an element (e.g. trying to 'press' a button that doesn't exist)
- Something wrong with the data (e.g. an account type we don't know how to work)

Usually when an exception occurs, the process stops and its status in Control Room shows as Terminated. Sometimes, it is not desirable for the process to stop when an exception is encountered. In certain contexts, we are able to handle the exception and continue with processing. For example, if a Customer System tells us we can't work a particular case because it is locked, we can trap the resulting exception, mark that case to be worked later, and then the process can carry on running and continue to work subsequent cases.

Hierarchy of Processing

The most important exception handling concept is the hierarchical nature of the processing. When an exception occurs, the following happens:

- If the current stage is within a [Block](#), and that block contains a Recovery Stage, control transfers there
- Otherwise, if the current page contains a Recovery Stage (outside a block), control transfers there
- If this page was called by another (e.g. as a page reference, an action or a subprocess call), then the exception occurs at the stage that made the original call, and the above rules apply once again.
- Ultimately if the exception goes unhandled by all the above, it must occur at a stage on the main page of the original parent process, with no Recovery Stage present. At this point, the process itself stops in an Exception state, and the process status will be marked as Terminated.

If a new exception occurs during recovery, the above process resumes from the next level up the hierarchy.

Exception Stage

An Exception Stage is used to raise an exception at any point during the process flow. An exception deliberately raised in this way behaves in the same way as any normal exception that may occur during processing.

The Exception stage has the following details associated with it:

- Exception Type - a user-defined label describing the category of exception. Users can create their own categories, such as "Data Exception" and "Timeout Exception". The Blue Prism product automatically collates and remembers all Exception Types created in all processes and business objects. This means that the same types can be reused globally in a consistent manner across all processes/objects.
- Exception Detail - A expression containing any relevant details of the exception.

Recover Stage

A [Recover Stage](#) provides a means of recovering from an exception. If an exception occurs on a page, and that page contains a Recover Stage, the process flow continues there. Once the process flow has transferred to a Recover Stage, the process is in Recovery Mode, and remains so until either a Resume Stage is encountered, or until a further exception occurs. If another exception occurs while still in recovery mode, the exception is not caught by the same recovery stage, but instead 'bubbles up' to the next level.

Resume Stage

A Resume Stage is used to signal that recovery has finished, and normal processing is continuing. Typically, a Recover Stage will be followed by a number of decision/choice stages that lead to different types of cleanup (depending on the type of exception). These stages can then be linked back into the main flow, which must be done via a Resume Stage. Without a Resume Stage, any further exception which occurred would not be handled by the same Recover Stage, but would bubble up to the next level, as if there was no exception handling in place. Once the Resume Stage has been passed, normal exception handling will take place again.

Blocks

A block is a rectangular region that groups together all the stages that fall completely within it. A block can contain its own Recover Stage, which will handle exceptions that occur within that block.

Visually, a block appears as a rectangle, with the internal area shaded. The block also has its name in a label in the top left-hand corner. This is the only field available on the block's properties form.

When a block is selected from the toolbar a new block can be created by dragging with the mouse to the desired size. Blocks can be resized by dragging one of their four corners. It is an error for blocks to overlap, if you have any overlapping blocks they will be shown in [Process Validation](#).

Exception Functions

The expression editor functions treeview includes an extra branch called Exceptions, containing the three exception functions ExceptionText(), ExceptionDetail() and ExceptionStage(). These functions can be used from a calculation or decision when the process is in recovery mode.

Additionally, a third item 'Types' is present, which can be expanded, and contains all the Exception types previously used. Dragging these items from the treeview pastes a string constant e.g. "UserNameTooLong" into the expression view.

Exception message dialogs

Exception message dialogs that usually pop up during debugging do not appear if the exception is handled by a Recovery Stage. Instead the process flow will jump to the recovery stage. In addition the message that used to pop up in a dialog will now be shown above the status bar of Process Studio.

Log History Viewer

The log history viewer can be accessed by clicking the button on the right of the status bar. Clicking this button pops up a textbox showing the last ten lines of log output. This is useful when you need to find out what caused the exception, or where the exception came from when the exception has been raised by a subpage.

Blue Prism Server

Introduction

Overview

Blue Prism can be operated in two different network architectures :-

Direct Access

Blue Prism clients, whether running the full Blue Prism client software, or simply running as a Runtime Resource, have direct access to the database which governs the Blue Prism environment.

This requires each client to store the authentication details for the database, and, depending on the authentication mode utilised, may give the user full access to the Blue Prism database, which is not desirable.

Blue Prism Server

The Blue Prism clients and resources connect to the BP Server service running on a central machine. This acts as a proxy for the database, and becomes the only place within the system which needs the authentication data to access and modify the data.

The [Blue Prism Scheduler](#) also runs within a BP Server service and it is possible to retain a 'Direct Access' architecture with a BP Server instance running primarily for the purpose of running the scheduler.

Configuring BP Server

Server Configuration Overview

A Blue Prism Server service is installed as part of the Blue Prism software installation process. It can be found in the **Installed Services** list in the Start Menu, reachable under: **All Programs : Administrative Tools : Services**.

The server is configured by running `BPServer.exe`, which is installed in the Blue Prism installation directory (usually `C:\Program Files\Blue Prism Limited\Blue Prism Automate`).

The dialog box which opens can be used to specify multiple configurations of Blue Prism Server, which can connect to disparate databases and listen to multiple clients.

This dialog is no longer used to configure the [Blue Prism Scheduler](#). This can now be configured on an environment-wide basis in [System Manager](#).



Server Configurations - Click for a larger image

A comparison of the Direct Access and Blue Prism Server architectures



Configuring BP Server

Editing an existing configuration

On installation, a default configuration for Blue Prism Server is created with the name `Default`. On opening the Server configuration dialog, it is this configuration which will be selected in the **Configuration** drop down box.

To edit a configuration, ensure that it is selected in the **Configuration** drop down and click on the **Edit...** button.

This will open the [Server Configuration Details](#) dialog with the current details stored for the selected configuration.

Note: The Blue Prism Server service which is installed by default when installing Blue Prism attempts to load a server configuration named 'Default' - if that doesn't exist, it will not start up. See [Running Multiple BP Servers](#) for guidance on how to configure a service to load a different configuration.

Creating a new configuration

Clicking on the 'New...' button will open the [Server Configuration Details](#) dialog where the configuration can be specified.

Deleting a configuration

To delete a configuration, ensure that it is selected in the **Configuration** drop down and click on the **Delete** button.

Note that the configuration will be deleted and the changes saved immediately.

Server Configuration Details

Within the 'Server configuration details' dialog, you can specify:-

Details tab

- **Name**

The name of the configuration. This is largely documentary, but is also used to indicate which server configuration a service should use when defining [multiple BP Server services](#). Note that in that case, a simple name without spaces or punctuation is recommended in order to simplify the installation of the corresponding service.

- **Connection**

The connection settings that the server configuration should use to connect to the Blue Prism database it is serving. The available connections are drawn from the connections configured within the Blue Prism Client's [Connections](#) dialog, available from the login page.



Configuring a BP Server Instance

- **Listening port**

The TCP/IP port on which it should listen for connections from Blue Prism clients. If running multiple servers simultaneously, these must be different for each server configuration or the server may not be able to start.

- **Connection Mode**

There following connection modes can be used for connections between a Blue Prism client and the Blue Prism server:-

- **WCF: SOAP with Message Encryption & Windows Authentication**

- Requires trust relationship between devices: Yes
- Blue Prism Authentication Modes: Blue Prism Native / Single Sign-on
- Requires server-side certificate: No
- Transport: SOAP over HTTP

Only the message content is encrypted. The SOAP and HTTP headers remain unencrypted which assists complex routing, load balancers, proxies etc. Client and server identity is validated via Windows / Active Directory.

- **WCF: SOAP with Transport Encryption & Windows Authentication**

- Requires trust relationship between devices: Yes
- Blue Prism Authentication Modes: Blue Prism Native / Single Sign-on
- Requires server-side certificate: Yes
- Transport: SOAP over HTTPS

The transport including SOAP headers are encrypted using certificate-based encryption. Client and server identity is validated via Windows / Active Directory.

- **WCF: SOAP with Transport Encryption**

- Requires trust relationship between devices: No
- Blue Prism Authentication Modes: Blue Prism Native
- Requires server-side certificate: Yes
- Transport: SOAP over HTTPS

The transport including SOAP headers are encrypted using certificate-based encryption. Server identity is validated using certificates.

- **.NET Remoting: Secure**

- Requires trust relationship between devices: Yes
- Blue Prism Authentication Modes: Blue Prism Native / Single Sign-on
- Requires server-side certificate: Yes
- Transport: TcpChannel over SChannel

Provided for backwards compatibility. Encryption is negotiated between the client and server. Client and server identity is validated via Windows / Active Directory.

- **.NET Remoting: Insecure**

- Requires trust relationship between devices: No
- Blue Prism Authentication Modes: Blue Prism Native
- Requires server-side certificate: No
- Transport: TcpChannel

Not recommended - provided for backwards compatibility. Connection security will need to be provided entirely by third-party solutions.

- **WCF: Insecure**

- Requires trust relationship between devices: Yes
- Blue Prism Authentication Modes: Blue Prism Native / Single Sign-on
- Requires server-side certificate: Yes
- Transport: SOAP over HTTP

Not recommended. Connection security will need to be provided entirely by third-party solutions.

- **Host Name or IP Address / IP Address**

When a server has multiple network interfaces, you can specify which IP address the server should use to listen on for connections. When using WCF connection modes, a host name can be used as an alternative to an IP address.

If this field is left empty, then Blue Prism Server will accept requests on any address.

- **Disable scheduler**

With this option checked the server using this configuration will not attempt to run any scheduled tasks. This might be useful in multiple server environments where a single server is required to be responsible for running schedules.

- **Enable published dashboards**

When checked the server will periodically retrieve data for any configured [Published Dashboards](#) and write the results to the Windows Event Log in JSON format for consumption by external applications.

Certificate Tab (WCF connection modes only)

- **Certificate Listing**

Any certificate bindings found on the machine for the current port will be listed on this panel. All bindings for the port are listed so that the user can determine whether they apply to the selected host name or IP Address.

New certificate bindings can be added from this section. Existing certificate bindings can be viewed or removed.

Add New Certificate Binding Window (WCF connection modes only)

New certificate bindings can be added from the "Certificate" tab.

- **Certificate Binding Address**

This section allows you to select whether to add your certificate binding for the current port using a "wildcard" IP address, a specific IP address or a host name. Options will vary according to the IP address or hostname specified for the server.

Note that operating systems prior to Windows 8 do not support bindings using host names. A warning will be displayed and you will need to add a binding using the "Any IP address" option instead.

- **Local Machine Certificate Store**

This allows you to specify the store to select certificates from. In most cases you should use the default "My (Personal)" option. You will be asked to select a certificate from this store after clicking the OK button.

Key Store tab

- **Encryption keys**

Any [Encryption Schemes](#) defined within Blue Prism that do not have their associated key held in the database, should have their key added to the Server Key Store along with the name of the scheme it is associated with.

Once in place, and the Server started, connected Blue Prism clients will be able to use the keys to encrypt and decrypt credentials and queue item data for this connection.

- **Store keys separately in individual files**

When this option is selected the keys themselves are held separately from the main server configuration, allowing access to them to be controlled via operating system permissions and encryption policies where required. Each key will be written to a separate file in the specified folder and named according to its associated [Encryption Scheme](#) name.

It is recommended that the specified folder is only used to hold Blue Prism encryption key files. Note that multiple server configurations cannot use the same folder.

In the event that the user modifying the server configuration does not have read access to one or more of the key files, this option will be disabled to prevent accidental key loss.

Server Services tab

- **Windows Services Listing**

A list of Blue Prism Server Windows services that have been detected on the local machine are displayed here. The list only displays services that are associated with the configuration being edited. The list includes information about the service setup and status.

In addition, it also includes details of whether the user account that runs the service has permission to listen on the address and port specified for the server configuration. Clicking the "Manage Permissions" link will open a window where these permissions can be managed.

If there are no Windows Services detected on the local machine for configuration currently being edited then you can create a new service by clicking the [Create Service](#) button or by running the example command text displayed on the tab from an elevated command prompt.

Manage URL Permissions Window (WCF connection modes only)

- **URL Permissions Listing**

Any URL permissions found on the machine for the configuration's address and port will be listed on this panel.

Note that URL permissions set up for an http URL will not work with a server that uses https and vice versa. It is also not possible to set up separate URL permissions for both http and https URLs (where the URLs are identical other than the http / https part). The listing will include permissions that use both http and https URLs so that the user can see any possible conflicts.

New URL permissions can be added from this section. Existing permissions can be edited or removed.

Add / Edit URL Permissions Window (WCF connection modes only)

URL permissions can be added or edited from the "Manage URL Permissions" window.

- **URL**

This section allows you to select whether to add your URL permission using a "wildcard" IP address or a specific IP address or host used by the server. Available options will vary according to the address specified for the server.

- **Users**

User accounts that are set to run the Windows services detected on the machine (together with other accounts that belong to an existing URL permission when editing) will be available to select here.

Logging Tab

- **Send Service status messages to Event Log**

Set checked to log status messages to the windows event log for this configuration.

- **Log verbose messages**

With this option checked all 'marshaled' and 'disconnected' messages will be logged to the Windows event log in addition to status messages.

- **Log traffic detail**

When enabled, all calls from clients to remote object instances on the server are logged. This option is provided for diagnostic purposes only.

The [OK](#) button will commit your changes, and [Cancel](#) will discard them. Note that the configuration will be saved permanently when you click [OK](#).

Running BP server

The server can be executed directly from the configuration program by selecting the desired server configuration details in the [Configuration](#) drop down and clicking the [Start](#) button. The server will run until the program is closed, or the [Stop](#) button is clicked.

Alternatively, it can be executed in the form of a Windows Service. Open the [Services](#) administrative tool and locate the [Blue Prism Server](#) entry.

From here, the service can be started or stopped, it can be made to start up automatically whenever Windows opens, and it can be configured to restart if any errors occur. Note that this will, by default, use the configuration named [Default](#). If this has been renamed or removed, the service will fail to start.

Running multiple BP Servers

You can run multiple Blue Prism Server instances on the same machine concurrently, potentially connecting to different databases, and serving different clients.

If running from the server configuration program, you can just open it multiple times, select different configurations to use and start each one independently.

If running as a service, then multiple services must be set up in order to enable this.

A service can be configured using the 'Service Control' program from Microsoft. See the [Microsoft Knowledge Base article Q251192](#) for further details regarding the Service Control (SC) program.

Once the required configuration has been set up in the configuration dialog, a service can be registered specifically for that configuration by running the command :-

```
sc create {SERVICENAME} binPath= "C:\Program Files\Blue Prism Limited\Blue Prism Automate\BPServerService.exe {CONFIGURATIONNAME}"
```

where {SERVICENAME} represents the name of the service to be used, and {CONFIGURATIONNAME} is the name of the server configuration that the service should use.

Note that the lack of space between 'binPath' and the equals sign, and the subsequent space between the equals sign and the path are important.

If the desired service name contains space characters, it should be wrapped in quotes, eg.

```
sc create "BP Server II" binPath= ...etc...
```

The configuration name should not contain any spaces or quote characters in order to be correctly referenced by the service control program.

Connecting to a BP Server

To configure a connection which can communicate with a Blue Prism Server instance, see [the Connections dialog](#), available from the [Login](#) page of the Blue Prism Client.

Blue Prism Scheduler

Introduction

Overview

Blue Prism contains a system which can be used to execute processes at specified times and repeat their execution at various intervals. The scheduler runs as part of a configured [Blue Prism Server](#) service which has access to the Blue Prism environment with the processes held in it.

The schedules can be configured to run once, or be repeated at minutely, hourly, daily, weekly, monthly or yearly intervals. [Calendars](#) can be employed to cause the schedule to run only on working days, skipping specific certain weekdays and/or public holidays.

Schedules

A schedule represents the point of execution of a set of tasks. Each schedule is self-contained and contains various data :- name & description, timing data and a set of [tasks](#) to perform.

When a schedule is executed, its 'Initial Task' is executed first and the subsequent tasks to perform are determined based on the outcome of that task. Thus, the tasks within a schedule are executed serially - ie. one after another.

A schedule cannot be executed concurrently with itself. For example, say a schedule starts at 12:00, and is scheduled to run again at 13:00. If, at 13:00, the schedule is still running, it will not be executed again.

Schedules are created in the Control Room's [Scheduler](#) tab, using the [Schedules](#) components.

Tasks

A task represents a component of a schedule. It defines a set of sessions which are to be executed concurrently, and it provides a coalesced status outcome of those sessions so that the scheduler can determine which task to execute next.

A session is a top level process which runs on a resource PC. Each task can contain an arbitrary number of sessions. When the task is executed, each session is first created and, once they are all registered and pending on their host resources, they are executed.

By default, a task is set to 'fail fast' - that is, if any session fails to be created, all sessions will be terminated and the task will be terminated. Likewise, if any session's execution causes an error, all other sessions will be terminated and the task will be terminated. Note that this means that a task is marked as terminated if any sessions fail.

If a task is set to not fail fast, any session creation failures are logged but do not cause the task to fail. Equally, any individual session executions which cause an error are logged, but any other running sessions continue. Note that, if not failing fast, a task is marked as terminated only if *all* of its sessions fail.

Each task has an 'On Success' and an 'On Exception' value, which determines the next task to be executed after the current task has executed and has, respectively, completed successfully, or has been terminated due to an exception when creating/executing its sessions.

Tasks are configured in the Scheduler tab using the [Tasks](#) components.

The Scheduler

The scheduler runs within a Blue Prism Server service. It is a background process which waits for the next schedule activation time and then executes any schedules due to run at that time.

The scheduler interprets the schedule activation times using the time zone configured on the server and therefore this should be considered when configuring the schedules. Additionally where there are multiple Blue Prism Servers with the scheduler enabled it is necessary to configure all servers to use a common time zone.

The scheduler is configured for a particular connection within [System Manager](#).

It can be configured to check to see if has missed any schedules when it starts up.

It checks the database every 30 seconds to see if it needs to refresh its current schedule data, so any changes to schedule timing will take at most 30 seconds to be recognised by the scheduler service itself.

If schedules are running when the scheduler is stopped or closed, they will attempt to terminate any active sessions cleanly and mark them as terminated in any schedule or session



Example Schedule - Click for a larger image

Example Schedule and its tasks

logs.

Process Features

Active Queues

Overview

[Work Queues](#) are available in the Blue Prism product to enable cases to be queued and worked in a predefined order within a session.

Using the traditional session management model, sessions are started on resources which poll the work queue for cases to work. These sessions are started in Control Room manually, or via a [scheduler](#) service which is running on a [Blue Prism Server](#) instance.

Active queues introduce an alternative mechanism for managing the sessions which work the queues, made possible by creating a closer association between work queues and sessions.

Instead of creating sessions separately in Control Room and then moving to the queue management page to see the results, active queues allow you to set a target number of resources which should be working the queue, Blue Prism uses the active queue configuration to determine how to achieve that target.

Active Queue Requirements

In order for Blue Prism to determine how to start and stop sessions on behalf of a queue, it needs to know two things about the queue:

- Which process works the queue?
- Which resources should be used to work the queue?

This information is set in the [Work Queue Configuration](#) page in the [System](#) section of the Blue Prism client, by associating a published process and a resource group with the queue.

With this information, Blue Prism can find an available resource and create a process on it which will then work the queue.

In order to stop a session which is running on behalf of an active queue, a 'stop request' is made to the session, allowing it to ensure it does not stop in the middle of a transaction. See [the process requirements](#) for more detail.

Process Requirements

In order to operate on behalf of an active queue, a process must meet some basic requirements.

- The process must be published before it can be set as the assigned process in an active queue.
- Assuming that the process contains a main loop, the function `IsStopRequested()` must be queried using a decision stage within that loop in order to ascertain if a safe stop has been requested.

This is the mechanism used by the active queue controller to stop sessions running on behalf of active queues that it is controlling.

How It Works

When the [Active Queues](#) section of Control Room is entered, an [Active Queue Controller](#) is created for each of the queues being displayed. These controllers remain in memory until the user logs out of the client.

Each controller monitors the sessions running on the resources assigned to their queue and polls the database for the latest statistics for the queue that they are responsible for.

When a target resource value is set, the controller determines whether it needs to create new sessions or stop existing sessions to reach that target. If new sessions are required, it will create and start the new sessions; if existing sessions must be stopped, it will send a stop request to the appropriate number of sessions to reach the target value.

Creating new sessions

If the queue controller determines that new sessions must be created in order to reach the target, it will create and start new sessions on the available resources in the resource group assigned to its queue.

It will create sessions on the least busy resource first, eg. if the group has three available resources and two of them are running a background process, the first session it creates will be on the resource which is not running any sessions.

If any sessions fail to be created or started for some reason, it will retry on other resources. After three attempts, it will abandon the session and decrement the target value to indicate that it will no longer be attempting to create the session.

If any errors occur which cause a session to be abandoned by the controller, they will be recorded in the Blue Prism event log and will be visible using the Windows Event Viewer.

Stopping existing sessions

If the queue controller decides that existing sessions must be stopped in order to reach the target, it will iterate through the sessions and choose the oldest sessions (ie. the sessions that were started at the earliest times) to send a stop request to.

Once a controller sends a stop request to a session, the session is marked as [Stopping](#). In this state, it is still 'running' - ie. it is contributing to the 'Active Sessions' count in the queue, and it is not available to run further sessions on.

In order to stop correctly, the worker process must check the `IsStopRequested()` function regularly, cleaning up its environment and exiting the process when the function indicates that a stop has been requested.

Session completions / terminations



Active Queues - Click for a larger image

Active Queues

When a session which is running on behalf of an active queue completes or terminates, it is taken out of the 'Active Sessions' count for the queue and becomes available again for further sessions.

If the completion / termination occurs after a stop request has been sent by a queue controller, the target resource value is unchanged (ie. the session was expected to stop in order for the target value to be reached).

If the completion / termination occurs before any stop request has been sent by a queue controller, the target resource value is decremented - ie. no other session is created to take its place.

Active Queues and Resource Pools

To a certain extent, [Resource Pools](#) are trying to solve the same problem as the assigned resource group in an active queue - that of finding an available resource and executing a process on it.

As such, it is not recommended to use resource pools within active queues - the active queue controller assumes that each resource represents a single slot for running sessions on behalf of its queue. Resource pools are transparent in that they appear to be single resources which can run multiple exclusive sessions and the queue controller can't drill down into the member resources to ascertain availability.

If you do include a resource pool in a resource group assigned to an active queue, it will be treated as a single resource in the 'Available Resources' count and it will only have a single session assigned to it, at which point it will be taken out of consideration for that queue.

Blue Prism Release Manager

Introduction

Overview

Blue Prism contains a mechanism by which components making up the configuration of Blue Prism can be transferred between different Blue Prism environments

The **Release Manager** allows bundles of components to be defined as a 'package'. The created package can then be exported into a file as a 'release'.

Packages

A package provides a mechanism for collecting a list of elements which should make up a release. It is created within the [Release Manager](#) in the Blue Prism client.

A package consists of a name and description, and its contents. The contents of a package can be made up of any or all of the following components :-

- Processes
- Visual Business Objects
- Web Service Definitions
- Process Groups
- Environment Variables
- Credentials
- Work Queues
- Schedules
- Fonts

Once a package has been created, you can create a [release](#) from it.

Packages can be modified or deleted from within Release Manager.

If a package contains a component which is deleted, the reference to that component is removed from the release

Releases

A release represents a package at a particular point in time - in more practical terms, a release is a file with all the contents of a package saved into it.

It is made up of a name, some release notes, and the detailed contents of the packages - ie. where the package contains the list of components, the release contains the components' data.

Creating

When a release is created, the file is saved to the hard drive, and an entry is saved to the database containing details about the release, such as its name, release notes, user, date/time created and its contents at the time that the release was created. Note that if the package contents change or if components within the package are deleted, the release's contents on the system remain the same.

A release is saved to a file, usually with a `.bprelease` extension. This can then be imported into another Blue Prism 4.1+ environment. When a release is imported into an environment, an entry is recorded on the database for that release too. This records similar details to the above releases, except that instead of the creating user and date/time created, the details will include the importing user and date/time imported.

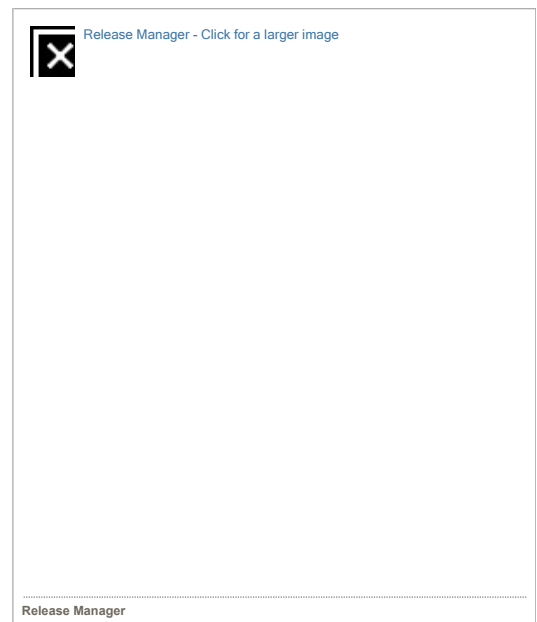
Importing

A release can be imported from within Release Manager, or by choosing **Import** from the **File** menu within the Blue Prism client.

The different components use different rules to determine how they should react to conflicts in the target system - most components require some form of user input to indicate the action that should be taken if such a conflict should occur.

Import Rules

The rules defining how each supported component is handled are given below:-



Processes

- If the target environment does not contain a process/VBO with the same ID or name as the incoming process, it is imported without any further user intervention.
 - If the target environment contains a process/VBO with the same ID as the incoming process, the user is prompted to either :-
 - Overwrite the existing process (Default) -or-
 - Save the process with a new ID.
 - If the latter is chosen and the existing process/VBO has the same name as the incoming process, the user is prompted for a new name for the incoming process.
 - If the target environment contains a process/VBO with the same name as the incoming process, the user is prompted to either :-
 - Rename the incoming process (Default) -or-
 - Overwrite the existing process/VBO -or-
 - Rename the existing process.
- If a rename is chosen, the user is prompted for a new name to rename the process / VBO to.

Visual Business Objects

- VBOs are handled identically to processes.

Web Service Definitions

- If the target environment does not contain a web service with the same name as the incoming service, it is imported without any further user intervention.
 - If the target environment contains a web service with the same name as the incoming service, the user is prompted to either :-
 - Overwrite the existing web service (Default)
 - Choose a new name for the incoming web service
 - Choose a new name for the existing web service
- If a rename is chosen, the user is prompted for a new name for the web service in question.

Process Groups

- Incoming process groups are merged with existing process groups with the same name in the target environment. If the incoming group contains processes being imported which are not part of the existing group, they are added once the incoming processes have been imported.

Environment Variables

- If no environment variable exists in the target system with the same name, the variable and its value in the source environment at the time of creating the release are set in the target environment.
- If an environment variable is found in the target environment with the same name, it is updated with the incoming description and datatype. If the existing value cannot be converted to the incoming data type, the incoming value is set in the target environment, otherwise the current value is retained.

Credentials

- If no credential exists in the target environment with the same name, the user is prompted for a username and password to set in the credential, and it is created.
- If a credential already exists in the target environment with the same name, the incoming credential is merged with it, such that the incoming description is set in the existing credential, and any incoming processes associated with the credential are associated on the target system.

The existing username and password are not affected by importing an identically named credential.

- Regardless of the existence or otherwise of the credential, the user has the option of not importing the credential as part of the import process.

Note: The default encryption scheme must be in place on the target environment before any credentials can be imported, otherwise an error will occur and the import operation will fail.

Work Queues

- If no work queue with the same name exists in the target environment, one is created with the incoming details.
- If a work queue with the same name exists in the target environment, its details (key field and max attempts) are overwritten by the incoming queue. Any items in the queue will be unaffected by these changes.

Schedules

- If no schedule exists with the same name as the imported schedule, it is created, with no scheduled sessions, in the target environment.
 - If a schedule with the same name exists within the target environment, its description, timing data and task structure is overwritten with the incoming schedule definition.
- Note that :-
- Note that if there exist any tasks with the same name as those being imported with the schedule, their descriptions and their "On Success" and "On Exception" settings are overwritten, but the list of scheduled sessions will be retained.
 - Also note that if the existing schedule is "retired" it will not be rejuvenated by importing a schedule over the top of it, although its data will change - it will have to be 'unretired' manually if required.

Calendars

Although calendars can't be explicitly added to a package, if a schedule requires a calendar to function, it is included along with the schedule in any generated release.

- If no calendar with the same name exists within the target environment, it is created there
- If a calendar with the same name exists within the target environment, it is overwritten with the incoming calendar data.

Fonts

- If no font with the same name exists on the target system, it is created with the incoming details.
- If a font with the same name exists, it is overwritten with the incoming font.

Verifying

There exists a mechanism with which a release can be verified against the environment from which it came - this checks to see if there have been any changes in the environment since the release was created.

Legacy Import/Export

Before the release manager, Blue Prism supported the import and export of single processes or visual business objects. Files created from these earlier versions of Blue Prism can still be imported using the same mechanism as importing a release - just change the file filter in the **File Open** dialog to look for *.xml files rather than *.bprelease files.

If you need to export a process / visual business object with a view to importing it in an older version of Blue Prism, you can still do so by opening the process / object in Process Studio / Object Studio and choosing **File | Export | This Process** or **File | Export | This Page** to export the process or the page as before.

Blue Prism Configuration

Windows Vista / 7 User Account Control

Overview

Windows Vista introduced 'User Account Control' to the Windows environment - a mechanism by which changes to the system which affect all users can be controlled by requesting elevated privileges when such changes are made.

How this affects the Blue Prism software

From a Blue Prism perspective, this affects any of the following configuration changes :-

- The list of database / server connections which the client and resource PCs connect to.
- Whether a resource machine runs when the Blue Prism client application is started up on a machine
- The external business objects registered on a machine
- The Blue Prism server configurations defined on a machine

If Blue Prism is running as an administrator (ie. the program is configured to always 'Run As Administrator'), then it will save the configuration with no further prompting from the user.

If the application is running under normal privileges, the user will receive an elevation prompt the first time the machine configuration needs to be saved. If accepted, a separate program called `AutomateConfig` will be run and will remain running as long as the Blue Prism client is running.

You can tell when AutomateConfig is running by looking in the notification area (the lower right of the screen). There will be a Blue Prism icon with a small cog overlaid on it.

If the program remains open longer than it should, it can be closed by right-clicking the icon and choosing `Exit`.

Help

Optical Character Recognition (OCR)

An OCR engine is available within Blue Prism for situations where it is not appropriate to use the native character recognition engine to interact with on-screen text. Commonly this will include scenarios such as where smoothed-text is enforced; or for interacting with scanned or otherwise-restricted copies of electronic documents.

The Blue Prism capability uses an embedded Tesseract OCR engine to recognise text using pattern matching and complex, language-based text recognition.

In order to maximise the effectiveness of the recognition a minimum of 300 dots-per-inch (dpi) is required. For images, such as on-screen text, where the dpi is lower than this, a scale parameter will artificially increase the size of the captured region before passing it to the engine. Generally setting the scale factor to 4 or 5 will provide successful results.

The OCR engine is leveraged through a `Read` stage when used against a previously captured Application Modeller region and includes the options to read text, lists and grids. It is also possible to output the pre-worked images to a specific diagnostics location to allow verification that the scaling being applied is sufficient for the selected region.

Language Packs

Language packs for use with Tesseract can be obtained from the internet. Blue Prism works with Tesseract version 3.05.01 and it is imperative that the correct major version of the language files are used with it. Currently, the version 3.05 language files can be downloaded from the [Tesseract website](#).

To add support for another language, download the appropriate files and copy them to the Tesseract\tessdata folder (usually `C:\Program Files\Blue Prism Limited\Blue Prism Automate\Tesseract\tessdata`).

The language files are prefixed with a language code e.g fra (French), deu (German), jpn (Japanese), chi-tra (Traditional Chinese) etc. Once installed on each of the required devices, this code can be specified in the Language parameter of the "Read Text with OCR" action within a `Read` stage, to instruct the engine to use the required pack.

Page Segmentation Mode

The "Read Text with OCR" action within a `Read` stage has an optional text parameter Page Segmentation Mode, allowing a Tesseract-defined value to be specified. The values which can be entered in this parameter are shown below, along with a brief description of their action.

If no value is entered for the Page Segmentation Mode, then the default value of Auto will be used.

OSD	Orientation and script detection (OSD) only
AutoWithOSD	Automatic page segmentation with OSD.
AutoNoOCR	Automatic page segmentation, but no OSD, or OCR.
Auto	Fully automatic page segmentation, but no OSD. (Default)
Column	Assume a single column of text of variable sizes
VerticalBlock	Assume a single uniform block of vertically aligned text
Block	Assume a single uniform block of text
Line	Treat the image as a single text line
Word	Treat the image as a single word
CircledWord	Treat the image as a single word in a circle
Character	Treat the image as a single character
SparseText	Find as much text as possible in no particular order.
SparseTextWithOSD	Sparse text with OSD.
RawLine	Treat the image as a single text line, bypassing workarounds that are Tesseract-specific.

For further information on segmentation modes please consult the official documentation provided by Tesseract on their website.

Help

This feature is subject to licensing restrictions. Please see the [licensing page](#) for further information.

Web Services - Overview

Blue Prism provides support for web services in two contexts: web services may be called as part of a process in the same way that other business objects are called and Blue Prism can expose processes as web services in their own right. Web services exposed by Blue Prism can be called remotely.

Note: In order to expose a process as a web service, it must be published, and its name must contain only alphanumeric characters - ie. no punctuation or space characters.

Web services should be thought of as a type of [Business Object](#). Indeed, users in Process Studio will not be able to tell the difference between a business object implemented as a web service and a business object implemented as an external library. The distinction is drawn in System Manager merely because the configuration options available for the two types of Business Object are different.

Web Services - Configuration

Web services to be made available as business objects may be configured in [System Manager](#). Existing web services are listed in the white area; new services may be added by following the [add a web service wizard](#). Once listed, such services will be visible as business objects from [Process Studio](#) and may be called using an [action stage](#).

Services may be enabled or disabled within Blue Prism by adding or removing (respectively) the checkmark found by the web service's entry in the list. When disabled, the web service of interest will still be visible from Process Studio, but the process will fail to run (be it in debug mode or from [Control Room](#)) because Blue Prism will deny access to the service.

Web Services - Commandline Configuration

Web services may also be registered via the commandline, using the [regwebservice](#) command.

Web Services - Glossary

- **WSDL** - Web Service Description Language. A WSDL document describes the capabilities provided by a web service, including what information is expected and what information will be returned.
- **URL** - A Uniform Resource Locator (URL) is a string of characters conforming to a standardized format, which refers to a resource on the Internet (such as a document or an image) by its location. A URL will be important to Blue Prism to locate a web service or a WSDL document.

Login Agent






Blue Prism Login Agent is a component that extends the capabilities of a Blue Prism Runtime Resource by providing a mechanism that can execute a secure log in on a Windows device.

Login Agent essentially allows a Runtime Resource:

- to receive instructions prior to the device being logged into to Windows (e.g. when the device is in a pre-logged in state)
- to send authorisation information to the operating system and to log into the device using either local or network credentials
- to carry out additional actions such as password resets

Login Agent creates and configures a Windows Service which is responsible for starting a dedicated Login Agent Runtime Resource which is able to communicate with the operating system to achieve a Windows log on.

The Login Agent Runtime Resource is shown using a dedicated icon within Control Room.

Available Resources	
Name	State
 WIN10CLIENT	Connected
 WIN7CLIENT	Connected
 WIN81CLIENT	Offline
 WIN-DG3LMS9017D	Connected
 WIN-DG3LMS9017D:8182	Connected

When appropriately configured, the Login Agent Runtime Resource is started whenever the machine is in a pre-logged in state, and remains active until the device has been logged on and a conventional Blue Prism Runtime Resource has been started. The Login Agent Runtime Resource is automatically shut-down by the start-up of a Blue Prism Runtime Resource.

The Login Agent Runtime Resource operates with a limited-access account and it will not have full access to the local machine. Therefore it can only be used for achieving Log in actions.

Installing Login Agent

Prerequisites

Windows Desktop OS: XP, Vista, 7, 8.1, 10 (32-bit and 64-bit)

Windows Server OS: 2008, 2008 R2, 2012, 2012 R2 (32-bit and 64-bit)

Administrator access is required to install Login Agent.

It is essential that the following security policies are applied to the target device.

- There must not be a requirement to press ctrl + alt + del prior to the username and password fields being presented
- There must not be a requirement to traverse an on-screen message such as a usage acceptance policy as part of the login process
- There must not be a requirement to traverse a lock screen (Windows 8.1 and Windows 10)

The install of Login Agent will apply these to the local policies on the device however within corporate networks the changes may need applying centrally.

Installing Login Agent

The installers for Login Agent can be found within the Blue Prism installation directory. It is necessary to select the installation file that matches the CPU architecture of the target device.

[Blue Prism Location]\Installers\LoginAgent_x86.msi for 32-bit operating systems

[Blue Prism Location]\Installers\LoginAgent_x64.msi for 64-bit operating systems

As part of the installation procedure it is necessary to provide a Connection Name which must be an exact match of the name of an existing Blue Prism connection on the local device.

A list of the currently configured connections on the device can be found within the Blue Prism client (pictured)

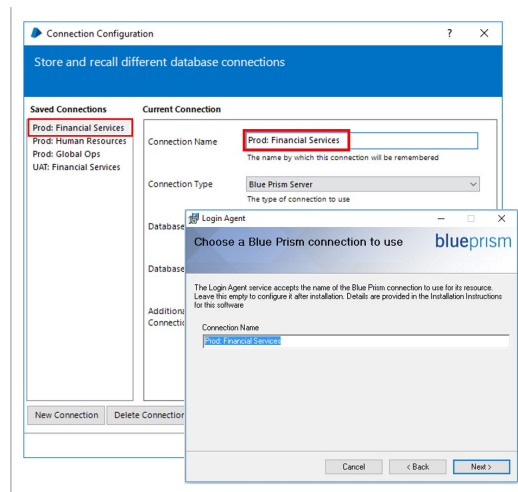
Login Agent Sample Process and Business Object

Once Login Agent has been deployed on the required devices, the Login Agent Release Package (`Login Agent Release.bprelease` within the Login Agent Install Directory) can be imported into the environment. This is achieved using the File -> Import menu option on any single device. The data is copied into the database so it only needs to be completed once for each relevant Blue Prism environment.

Two of the provided sample processes (Login and Change Password) require that a Credential record is created for each device. These credential records need to be created using the default naming format `Windows Login: [MachineName]`. E.g. if the Runtime Resource is configured on robot0001 on port 8190, the default credential name should be `Windows Login: robot0001`.

This VBO provides a set of example actions that can be used to achieve common authentication actions with the operating system such as Log In, Is Logged In, Log Out, Change Password.

Information regarding the Login Agent VBO and its actions can be found in the API documentation under [Help > API Documentation](#).



When overwriting existing versions of the Login Agent VBO, it is necessary to re-verify any processes that use the provided functionality.

Advanced Topics

Updating or customising the Login Agent configuration

The configuration of Blue Prism Login Agent service, which is responsible for initialising the Login Agent Runtime Resource, is stored within a local configuration file:

```
C:\ProgramData\Blue Prism Limited\Automate V3\LoginAgentService.config
```

The `workingdirectory` element points to the installation directory for the Blue Prism software.

```
<startuparguments>
  <argument name="resourcepc" />
  <argument name="public" />
  <argument name="port">
    <value>8181</value>
  </argument>
  <argument name="dbconname">
    <value>Prod: Financial Services</value>
  </argument>
</startuparguments>
```

The `startuparguments` element gives the arguments that will be used when launching the Login Agent Runtime Resource. For example this section can be configured to update the port that the Runtime Resource will listen on, or the name of the connection that it will use to connect to the environment.

Resources configured with Certificate-based encryption

```
<argument name="dbconname">
  <value>Prod: Financial Services</value>
</argument>
<argument name="sslcert">
  <value>[Thumbprint]</value>
</argument>
```

Where the conventional Runtime Resources are configured to force encryption of incoming connections using a specified certificate (e.g. where the Runtimes are started using the `/sslcert` switch), it is necessary to manually apply the appropriate configuration to the Login Agent Runtime Resource.

The `startuparguments` element within the configuration file can be updated to include the appropriate information:

```
<argument name="dbconname">
  <value>Prod: Financial Services</value>
</argument>
<argument name="sslcert">
  <value>[Thumbprint]</value>
</argument>
```

Silent Install

A silent install can be achieved through use of `msiexec`. It is important to ensure the correct installer is selected for the CPU architecture (i.e. `x86` is for 32-bit operating systems; `x64` is for 64-bit operating systems).

It is necessary to provide the following parameters:

- **[Blue Prism Install Location]**, the location where Blue Prism has been installed. This is commonly where the Login Agent installation file is located
- **[Connection Name]** which must be an exact match of the name of an existing Blue Prism connection on the local device. A list of the currently configured connections on the device can be found within the Blue Prism client.

For example:

```
msiexec /i "[Blue Prism Install Location]\Installers\LoginAgent_x86.msi" /quiet BPCONN="[Connection Name]"
```

For example:

```
msiexec /i "c:\Program Files\Blue Prism Limited\Blue Prism Automate\Installers\LoginAgent_x86.msi" /quiet BPCONN="Prod: Financial Services"
```

Configuring Logging

Login Agent can be configured to generate diagnostic logs on a specific device by configuring the appropriate Registry key settings.

For appropriate versions of Login Agent, the keys can be found within the Registry at the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Blue Prism Limited\LoginAgent
```

- **LogFileDir**: specifies the location where the log file will be generated.
- **LogLevel**: specifies the granularity of logs. 0: Disabled (default); 1: Error messages; 2: Debug messages; 4: Trace messages. For a combination of levels, the values can be added together. E.g. a value of 7 will provide error messages, debug messages and trace messages.

Logging is only recommend while troubleshooting. It is necessary to reboot the device to apply registry setting changes.