blueprism®

Robotic Process Automation Software

Remote Desktop Services

USER GUIDE

Version: 6.3
Document Revision: 1.0

blueprism®

Contents

Introduction

Blue Prism® is commonly installed and run on a virtualised infrastructure with virtual desktop Images being used for Blue Prism Runtime Resources. This remains the recommended deployment approach as it offers users flexibility and control, and provides identical user interfaces for human users, developers, and the digital workforce.

For users with specific IT infrastructure requirements or for reducing costs, it may be desirable to deploy Blue Prism on a Remote Desktop Services (RDS) environment, employing the Remote Desktop Session Host (RDSH) technology in Microsoft's Windows Server operating system (formerly known as Terminal Services).

Using RDS to host Blue Prism can dramatically increase the number of virtual workers which can be deployed per server, as no additional overhead is required for the virtual infrastructure software or for the OS on each VDI. However it has two significant disadvantages:

- High availability is difficult to achieve

- Sessions on a RDSH server all share a Windows process stack

These two issues are explored in more detail in Limitations. However, due to these reasons Blue Prism does not recommend using RDSH for business-critical environments.

Solution Overview and Configuration

This section describes the necessary steps to configure and use RDS within Blue Prism.

Limitations

As mentioned in the Introduction, using RDSH comes with two significant disadvantages:

High Availability

As numerous Runtime Resources are running on a single server, if that server experiences a fault and crashes, all the sessions are lost, with a correspondingly severe impact on the Blue Prism environment. High availability can be provided to some extent using Connection Brokers and RDS Gateways. However, typically these only ensure the RDS environment can be readily contacted - they do not prevent data loss or interruption to in-progress business processes caused by disrupted sessions.

Windows Process Stack

Applications opened in one user session are visible in the process stack of other sessions. The UI of these applications is hidden as they are Background processes, and the processes cannot typically be controlled by other users without elevating permissions due to the user context. For example, sending a "Kill Process" action in Blue Prism to Internet Explorer will kill it only in the current user context, not for all user contexts.

Although this limitation is mitigated for the reasons above, there are two major risks that still persist:

- Some applications will not behave correctly in this context. For example, they may only allow one instance to be running per machine or, if they are located elsewhere on the network, it may not be possible to make simultaneous connections to them from a single machine under different user contexts. It is expected that these cases will be rare, however all line of business applications expected to be used in an RDS context should be tested for compatibility first.

- The full process stack is visible to other processes running on the same server. In practical terms this can make it difficult to attach by Window Title. To mitigate this where possible, applications should be launched by the Runtime Resource which will be using them (in which case no attach stage is necessary), or attachment should be by Process ID. This is not always feasible however and Solution Designers should keep this context in mind when designing Blue Prism processes. The attach stage also includes a User Name option, and using this also helps; if using the User Name option it should be noted that the Process Name needs to include the ".exe" suffix.

Component Diagram
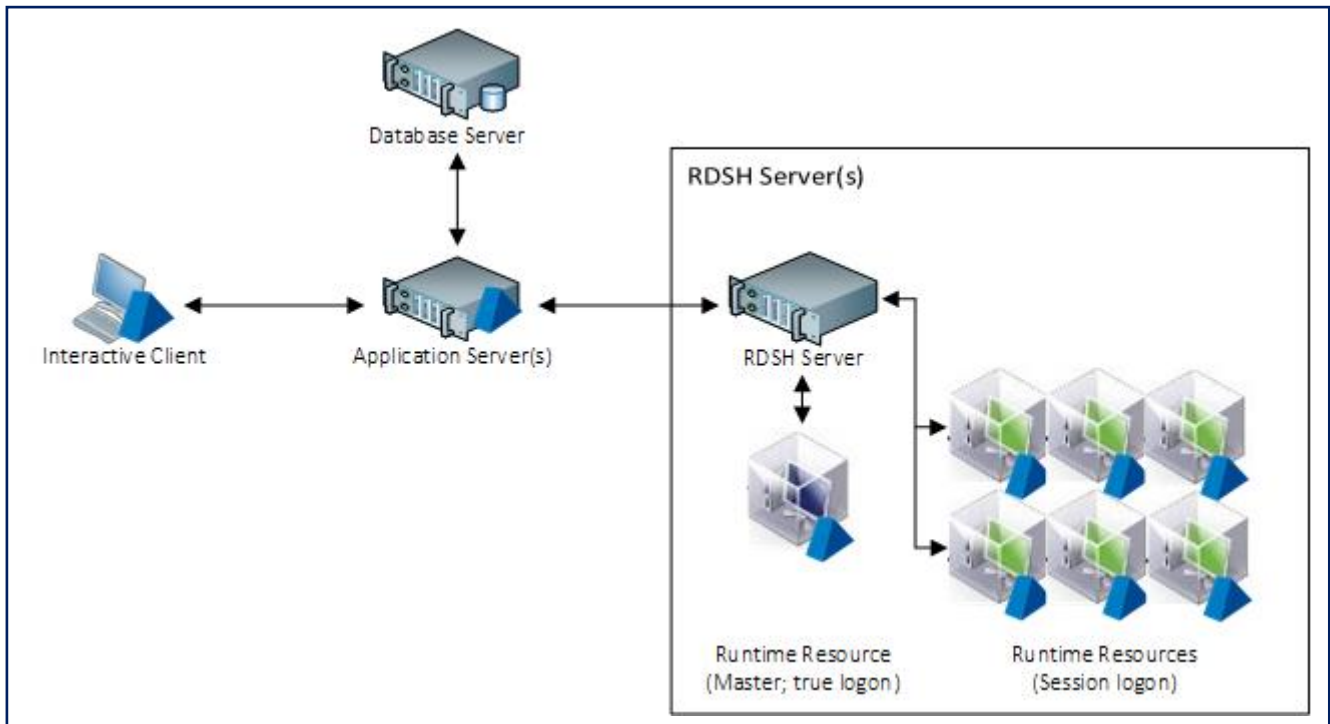
Session Creation

Creating Sessions

There are numerous ways to create user sessions on an RDS server, the simplest being to interactively log in to the server from a remote location - which is the expected model for which RDS is designed. However, this approach is not suited to unattended automation in an RPA context because:

- A user has access to the windows session while a digital worker is executing tasks, providing the potential to disrupt the process or observe sensitive information on the screen.

- If network connectivity is disrupted at either end, the automated process will be disrupted as well.

To mitigate against these risks, the Blue Prism Remote Desktop Services VBO contains an action called *Start RDP Session* which, when executed, creates a new RDS session on the local machine on which it runs. It also makes the Remote Desktop Connection window invisible. This means that a user on the RDSH server cannot view or interact with the Runtime Resources as processes are executed and, if network connectivity is disrupted, the session is maintained while the server attempts to re-establish its connection.

The VBO must be run on a Runtime Resource. To achieve this, it is recommended that the Blue Prism Login Agent be used to create a single "true logon" session on the server, with the resulting Runtime Resource listening on port 8181 if possible. Future Windows sessions can then be started by executing the *Start RDP Session* action on this runtime resource.
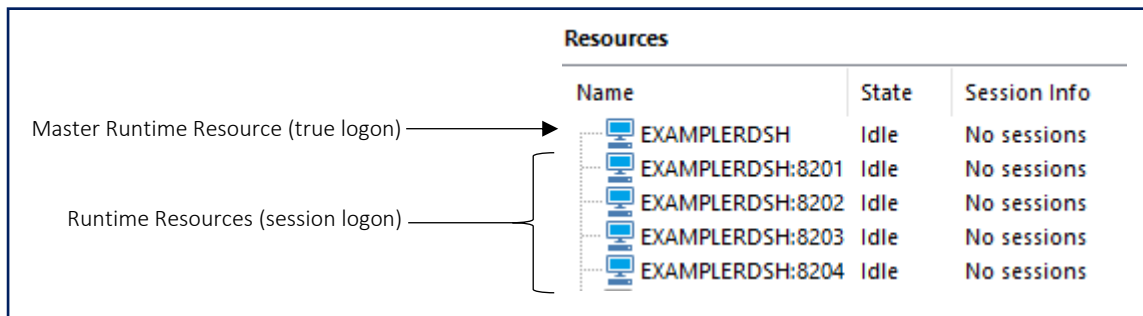
All sessions must always be started with a unique user account: it is not possible for a single user to have multiple RDS sessions on a single machine.

Starting Runtime Resources

When new Windows sessions are created using the *Start RDP Session* action, a Runtime Resource is not automatically started by default. Instead it should be started in the normal way, with a batch file, Task Scheduler task, or similar, used to start Automate.exe when a particular user logs in.

In this configuration it is necessary to specify a port for the Runtime Resource to listen on. It is therefore recommended that each user be assigned a specific port: whenever that user signs in on that RDSH server, the listening port is predictable. This facilitates the use of the scheduler and prevents port conflicts.

The resulting control room will look similar to this:

Closing Sessions

There is an important distinction in Windows between disconnecting a session and a user signing out.

When a session is disconnected, or when a user chooses to disconnect, the session stays alive on the server and the user remains logged in, but the UI is suspended. Therefore, resources are not commonly released and any Blue Prism activity related to the UI layer fails. For this reason it is strongly recommended that the disconnect feature is not used.

When a user signs out, the session is destroyed and all resources are released. The Runtime Resource running inside the session will disconnect from the application server. This is the cleanest exit from an RDS session and is the recommended way of closing sessions when not in use. The Blue Prism Remote Desktop Services VBO contains two actions which will achieve this:

- *Log Off Current* should be executed from inside the RDS session to be disconnected and will log that user out.

- *Log Off User* can be executed from any session on the RDSH server where the user has admin rights, and takes a session ID as the input. The session ID for a particular user can be obtained using the *Get Logged In Users* or *Get Current Session ID* actions.

Environmental Prerequisites and Configuration

This section provides an overview of the prerequisites and necessary configuration of the Windows platform on which RDS operates. It is expected that users wishing to deploy to an RDSH environment will have in-house expertise to provision such an environment.

RDSH Environment

There are numerous steps that must be taken to set up a single Windows Server for RDSH or to create an RDSH farm. These steps are documented in full by Microsoft, however the following steps are usually required:

- The Windows Server must have the Remote Desktop Session Host role installed.

- Microsoft RDS CAL licences are required on either a per-user or per-device basis. Microsoft provide a free 120 day trial of the RDSH role on any Windows Server, meaning it can be configured and used for 120 days without any license server or licences in place.

- Microsoft's Remote Desktop Connection software must be configured to allow saved credentials to be used for connections to "localhost", and the identity certificate permanently accepted.

- As described in Starting Runtime Resources, Task Scheduler or similar must be configured for each user to start Automate.exe on a specific port when that user signs in.

- All ports which might be used by Runtime Resources on the server must be open, without firewalls or similar blocking them.

Performance and Scaling

Idle RDS sessions with a Blue Prism Runtime Resource running consume negligible CPU and around 80mb of RAM. Therefore, it is possible to deploy dozens or even hundreds of Runtime Resources on even a modestly sized server.

Of course any line of business applications running on the server will also consume resources, as will active Blue Prism processes running on each session. Therefore, it is necessary to test the actual performance and resource consumption of processes to determine the necessary scaling of the RDSH server.

Using the Remote Desktop Services VBO

Users who are planning to run Blue Prism on an RDSH infrastructure can use the following actions in the Remote Desktop Services VBO to control the RDS sessions. In some cases these need to be run from inside RDS sessions, in other cases they should be executed from the host session.

The runmode of the VBO is "background".

Start RDP Session

Initiates a Remote Desktop Connection on the local machine.

Parameter	Direction	Data Type	Description
Username	In	Text	The username to log in with
Password	In	Password	The password to log in with
Delay	In	Number	A delay (in seconds) between starting the Remote Desktop session and deleting the cached credentials. This can prevent the credentials being deleted before the remote session is able to use them. Default is two seconds.
Use WOW64 Redirection?	In	Flag	Defaults to False. This should be set to True when running on a 64-bit server operating system if WOW64 Redirection is not already in place on the server.
Process ID	Out	Number	The Process ID for the Remote Desktop Connection

Get Logged In Users

Returns a collection containing all the users currently logged into this machine, with the session ID and current state of each.

Parameter	Direction	Data Type	Description
Users	Out	Collection	A collection containing all the users currently logged into this machine, with the session ID and current state of each

Get Current Session ID

Retrieves the session ID of the user context in which this action is executed.

Parameter	Direction	Data Type	Description
SessionID	Out	Number	The session ID of the user in whose context this action was executed

Log Off User

Logs off the user in a specified RDP Session.

Parameter	Direction	Data Type	Description
Session ID	In	Number	The session ID of the user to be logged off
Success	Out	Flag	Indicates whether the user was successfully logged out

Log Off Current

Sends a log off command in the current RDP Session. This should be run from inside the session to be logged off.

Launch With PID

Launches an application and returns its PID.

Parameter	Direction	Data Type	Description
Application	In	Text	Full file path of the application launch file (for example "C:\Program Files\Internet Explorer\iexplore.exe")
Application Parameters	In	Text	Any parameters to pass to the application as part of the launch (for example "https://www.blueprism.com/") (Optional)
PID	Out	Number	The Process ID for the launched application