RenRøros
Intelligent Automation as

Blue Prism Master Class

Stuck Pending Sessions

Have you ever found a session stuck in the pending status?

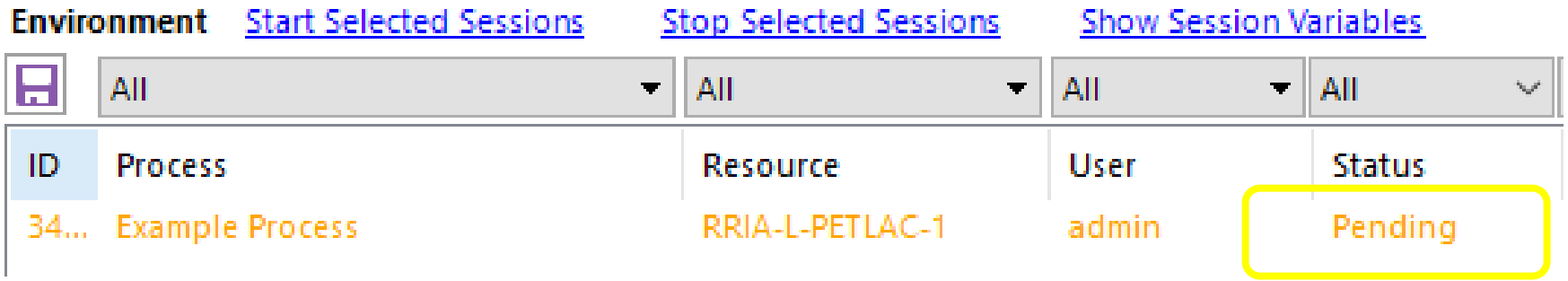
You are not alone! The Blue Prism service hosted on an application server is responsible for triggering processes on runtime resources. To successfully start a process 2 things must happen.

1. The process is registered to run on the resource – this consumes a license
2. The process is triggered to start on the resource

A stuck pending session occurs when point 2 fails. The underlying reason for point 2 failure is more difficult to pin-point but lies somewhere within the inners of your local network.

As a rule, the scheduler will not attempt to re-run a pending session therefore it is the responsibility of a controller to manually run, or delete, the session.

External monitoring for stuck pending sessions

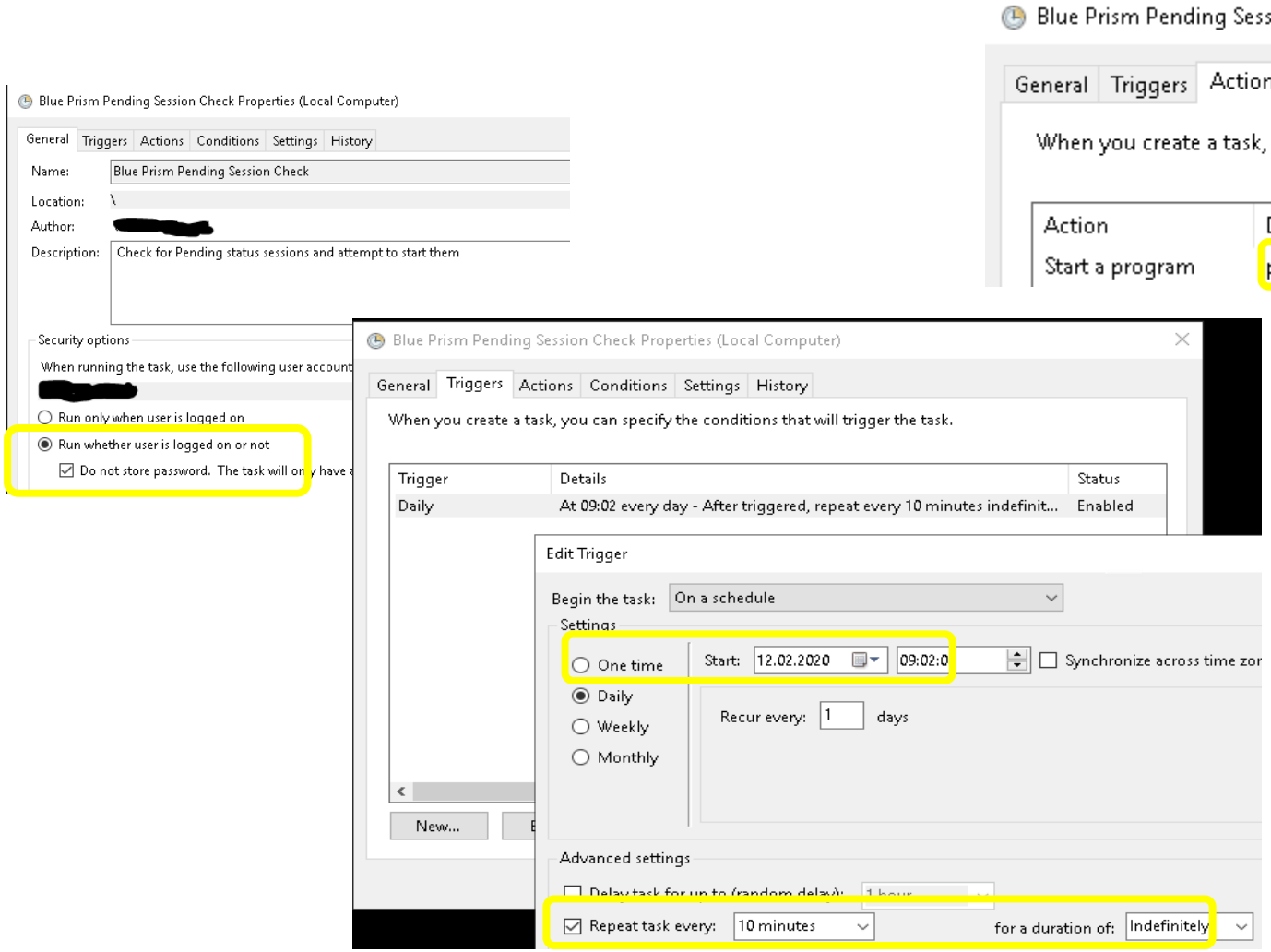


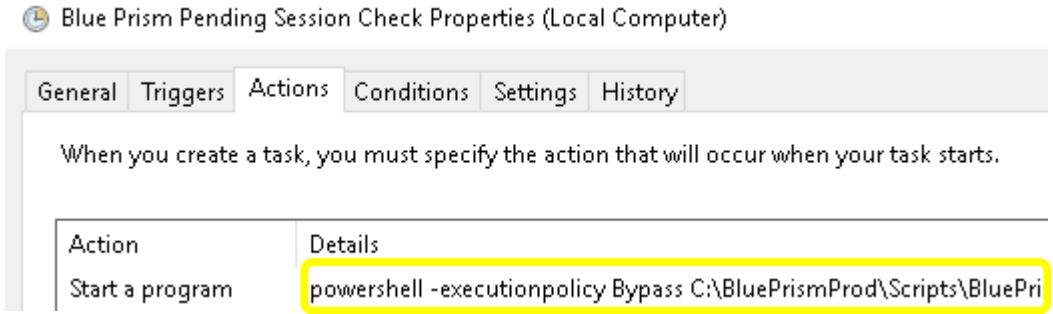
Given a pending session is consuming a valuable license it is important to ensure the situation is resolved as fast as possible. By monitoring for this scenario we can

1. **Notify that is has occured**
2. **Start the session automatically**

RenRøros

Task schedule



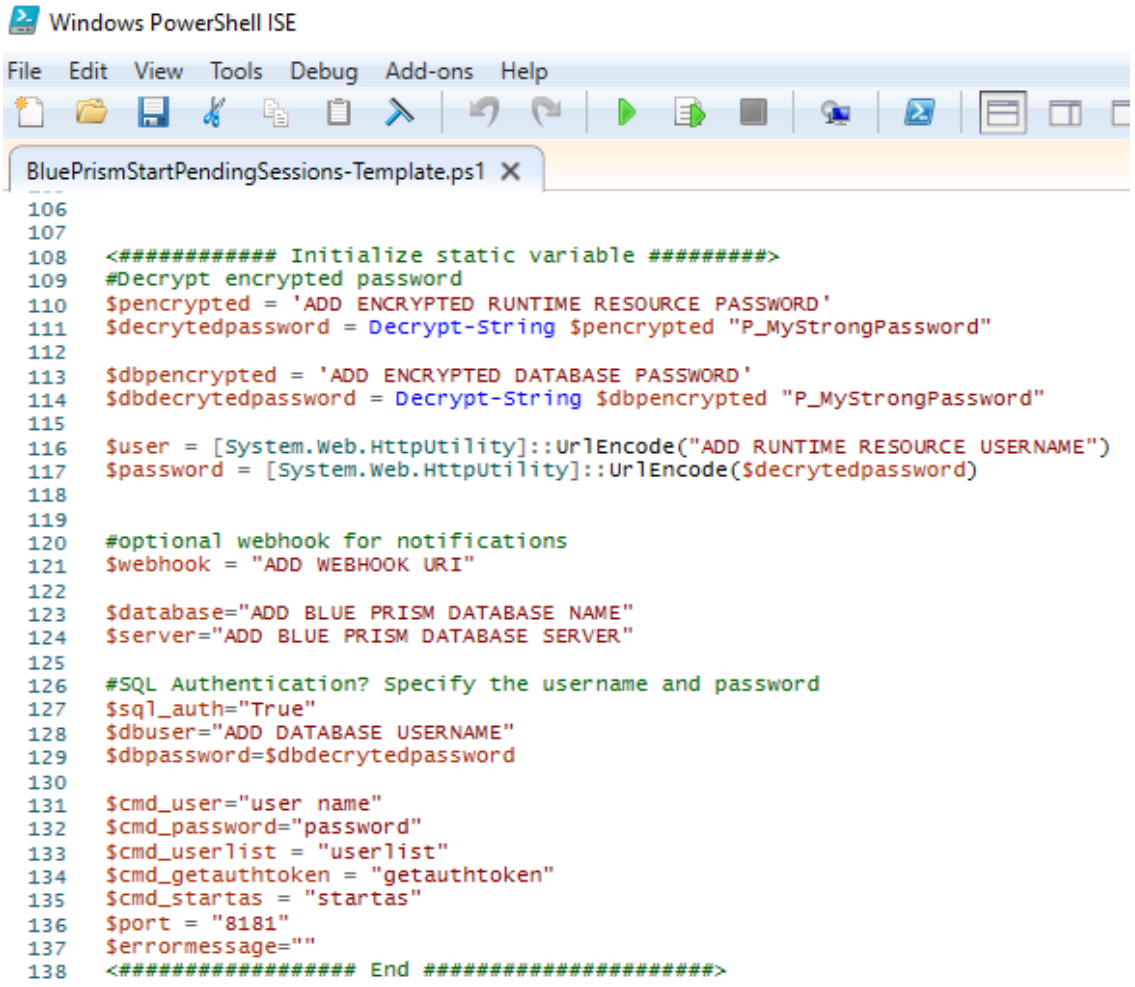


The Windows Task Scheduler is used to trigger a powershell script.

1. Configure on the App Server
2. Run as administrator
3. Run at an appropriate interval
4. Run the powershell script with an appropriate policy

RenRøros

Powershell script configuration




```
106
107
108   <############ Initialize static variable #########>
109   #Decrypt encrypted password
110   $pencrypted = 'ADD ENCRYPTED RUNTIME RESOURCE PASSWORD'
111   $decrytedpassword = Decrypt-String $pencrypted "P_MyStrongPassword"
112
113   $dbpencrypted = 'ADD ENCRYPTED DATABASE PASSWORD'
114   $dbdecrytedpassword = Decrypt-String $dbpencrypted "P_MyStrongPassword"
115
116   $user = [System.Web.HttpUtility]::UrlEncode("ADD RUNTIME RESOURCE USERNAME")
117   $password = [System.Web.HttpUtility]::UrlEncode($decrytedpassword)
118
119
120   #optional webhook for notifications
121   $webhook = "ADD WEBHOOK URI"
122
123   $database="ADD BLUE PRISM DATABASE NAME"
124   $server="ADD BLUE PRISM DATABASE SERVER"
125
126   #SQL Authentication? Specify the username and password
127   $sql_auth="True"
128   $dbuser="ADD DATABASE USERNAME"
129   $dbpassword=$dbdecrytedpassword
130
131   $cmd_user="user name"
132   $cmd_password="password"
133   $cmd_userlist = "userlist"
134   $cmd_getauthtoken = "getauthtoken"
135   $cmd_startas = "startas"
136   $port = "8181"
137   $errormessage=""
138   <################## End ####################>
```

Configuration items

1. Database username *
2. Database password **
3. Database servername
4. Database name
5. Blue Prism username
6. Blue Prism password **
7. Optional webhook for notifications

* A read-only database user must be created

**To avoid using clear-text passwords in the script an accompanying encryption-script will decrypt the password that can be pasted into the script.

RenRøros

Database function

A database function must exist in the database that will query for pending sessions. The parameter will specify the threshold for how old, in minutes, the pending session is.

The details returned will be used to automatically start the session

1. sessionid
2. processid
3. resource

RRIA_FN_GetPendingSessions.sql – Notisblokk

Fil Rediger Format Vis Hjelp

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

IF EXISTS (SELECT * FROM sys.objects WHERE type = 'TF' AND OBJECT_ID = OBJECT_ID('dbo.RRIA_FN_GetPendingSessions'))
	DROP FUNCTION [dbo].[RRIA_FN_GetPendingSessions]
GO

CREATE function [dbo].[RRIA_FN_GetPendingSessions](
	@NumMins int = 1
	)
returns @summary table(
	[sessionnumber] varchar(255),
	[sessionid] varchar(255),
	[processid] varchar(255),
	[processname] varchar(255),
	[resource] varchar(255),
	[startdatetime] datetime,
	[statusid] varchar(255)
	)
as
begin
	-- Return the aggregated information
	insert into @summary
		(sessionnumber, sessionid, processid, processname, resource, startdatetime, statusid)
	--SQL STATEMENT HERE
	select  convert(varchar(255), s.sessionnumber) as sessionnumber,
		s.sessionid as sessionid,
		s.processid as processid,
		p.name as processname,
		r.FQDN as resource,
		s.startdatetime as startdatetime,
		'Pending' as statusid
	from BPASession s, BPAProcess p, BPAResource r
	where s.processid = p.processid
	and s.runningresourceid = r.resourceid
	and s.statusid = 0 --pending
	and s.startdatetime < dateadd(minute, -@NumMins, getdate())
	return;
end
```

RenRøros

Powershell script run time

When the script executes when no pending session exists it will do nothing.

```
PS C:\BluePrismProd\Scripts> C:\BluePrismProd\Scripts\BluePrismPendingSessionCheck.ps1

GAC    Version        Location
---    -------        --------
True   v4.0.30319     C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Security\v4.0_4.
powershell script starting...
No pending sessions
powershell script complete
```

When a pending session is returned from the database function, a HTTP request is made to the resource to start the pending session.

```
PS C:\BluePrismProd\Scripts> C:\BluePrismProd\Scripts\BluePrismPendingSessionCheck.ps1

GAC    Version        Location
---    -------        --------
True   v4.0.30319     C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Security\v4.0_4
powershell script starting...
number of pending sessions found  1
sessionnumber  24712
sessionid  AF17F207-C808-4CD7-804B-7474B281F125
processid  FE175CE9-5B01-440E-8D92-A26685B7384F
processname  Check Logged In
resource  [redacted]
timestamp  13.02.2020 23:19:32
status  Pending
1
finding user id...
userid found:  74AFAC1B-74A8-4BB1-93D3-04AAF30EFF6C
result:  STARTED

1
powershell script complete
```

If configured, a webhook is notified of the stuck pending session together with the triggered results.

RenRøros

Powershell dependancies

The Powershell module SqlServer must be installed on the Application Server.

PS> **Install-Module –Name SqlServer**

Ensure the SecurityProtocol is TLS 1.2

PS>**[Net.ServicePointManager]::SecurityProtocol**

Tls12

If not then set it.

PS> **[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12**

RenRøros