



Background Runtime Service (BRS) Overview

For Cora SeQUENCE 9.0 and later

October 2022

Notice

All rights reserved. No part of this document (including the text, images, graphics or the selection or arrangement of content, which forms an original compilation), may be reproduced or transmitted in any material form or by any means, electronic or mechanical, including photocopying or recording in any medium (whether or not transiently) without the written permission of the copyright holder. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. The same shall apply to the export of this publication from India, and a violation of this condition will lead to civil and criminal prosecution.

Table of contents

General document information.....	4
Purpose.....	4
Prerequisites.....	4
Introduction	4
BRS Deployment topologies.....	4
Deployment and BRS config file	5
Redirector thread.....	5
EmailSender thread.....	6
workflowLateStatusModifierThread.....	7
dbListener thread	7
StatisticsCalculator thread.....	8
Subscriber thread	8
ExpiredTemporaryAttachmentsCleanerThread.....	9
ClosedInstancesSeparator thread.....	9
ClosedRecipientBreadCrumbsCleaner thread	10
ReminderMessageCreator thread.....	10
stuckSubWorkflowActivityExcutorThread	11
EmailSenders	11
Messaging.....	12
Multiple BRS Deployment.....	13
Notes on multiple BRS deployment.....	13
Troubleshooting.....	13
Use the SVC log files	14
BRS stops.....	14
BRS won't start.....	14
Security exception in reminder thread	14
Email sending issues	14

General document information

Purpose

The purpose of this document is to describe how to configure the Background Runtime Service (BRS).

Prerequisites

You should have access to the server on which the BRS is deployed.

Introduction

The Background Runtime Service (BRS) is an integral part of the functionality of Cora SeQUENCE. The BRS is installed as part of the Cora SeQUENCE installation and continues to run in the background at all times.

The BRS is a Windows service that has several threads. Each of these threads is responsible for performing one of the following actions:

- Redirect workflows
- Send emails
- Control workflow status
- Execute the Database Listener activity
- Calculate workflow statistics
- Sync work with other BRS instances
- Clean the table of temporary attachments
- Move closed process instances to separate tables
- Create reminders
- Resume workflow execution

Each of the above threads includes configuration options, such as how often and when to perform the action. The configuration options are described in the following sections.

BRS Deployment topologies

There can be several instances of the BRS. They can all be deployed on a dedicated server or divided into separate servers.

All the BRS instances work together, based on load balancing.

The registration of a new BRS instance is done automatically. The system is updated automatically when services are shut down or go offline, without any need for manual action.

Deployment and BRS config file

Name: **PNMsoft.Sequence.BRS.exe.config**.

Default location: **C:\Program Files\Genpact Digital\Cora SeQUENCE\Background Runtime Service 1**

To configure BRS options, edit the config file as described below. You can also keep the default options, if no special functionality or optimization is required.

Configuration options

The BRS config file PNMsoft.Sequence.BRS.exe.config contains several sections for each of the threads. Each section has several options that you can edit.

Redirector thread

The redirector thread redirects workflows, progressing workflows to subsequent activities in the flow. The redirector picks workflow instances that have an open link with type: **Automatic** or **All**, and whose last updated or next redirect date matches the last redirect of the BRS instance. The workflows are redirected in a FIFO (first in first out) order.

This thread is defined in this section:

```
<redirectorThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.Redirector,
  PNMsoft.Sequence.WindowsServices.BRS"
  interval="10"
  pageSize="0"
/>
```

Options

interval: The idle time of the redirector thread (in minutes) between completing a cycle and starting a new cycle. For example, if set to 10, the thread will wait 10 minutes between cycles. To disable the thread execution set its interval to 0. (Unit = minutes).

pageSize: You can use this option to redirect chunks of the workflow, especially if there are many instances and you are interested in improving performance. For example, if set to 15, the thread will redirect 15 workflow instances at a time. If set to 0, the thread will redirect all workflow instances at once. (Unit = number of workflow instances).

Note

If there is more than one BRS installed, the redirection work is divided between them, so this thread will pick only the relevant part of the work.

EmailSender thread

This thread defines configuration for emails that are sent from *Task* and *Message* activities in Cora SeSequence.

Email messages from “Email” activity are not affected by this configuration.

The protocol details need to be defined in the Administration site (under Global Settings). This thread takes the protocol settings labeled as “(default)”.

Note

This thread sends emails asynchronously to the workflow execution (that is, there’s a delay of a few minutes).

This thread is defined in this section:

```
<emailSenderThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.EmailSender,
  PNMsoft.Sequence.WindowsServices.BRS"
  interval="10"
  sendRetries="10"
  siteUrl="http://serverName/Sequence/"
  outlookWebService="http://serverName/Interfaces/"
  messageXsltFolder="Messaging\XSLT"
  messageLogosFolder="Messaging\Images"
  sendLogos="False"
  useHtmlBody="True"
  attachmentLimitationSize="2048"
/>
```

Configuration options

interval: The idle time of the thread between sending emails. (Unit = minutes).

sendRetries: How many times to try sending an email.

siteURL: The location of the Flowtime (runtime) site. This is provided as a link within each Cora SeSequence email, enabling users to easily return to the Flowtime site.

Important

After installation, you need to configure the **siteURL** as follows:

`https://<domain name>/_layouts/`

messageXsltFolder: The path to the message body XSLT folder in which the email body format is defined. This folder contains email message body templates per language.

messageLogosFolder: The path to the folder that contains the logo files for Cora SeSequence emails.

sendLogos: Defines whether or not to include logos as part of Cora SeSequence emails. (Values: True/False).

useHtmlBody: Determines whether the email body is built using HTML or plain text (Values: True/False). Set this option to false if your users cannot see HTML emails with their email client.

attachmentLimitationSize: Defines the maximum size of an attachment to an email. (Unit: Kb)

workflowLateStatusModifierThread

This thread controls the color of the “Due Date” field of a workflow instance, as displayed in the Flowtime site.

This thread is defined in this section:

```
<workflowLateStatusModifierThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.WorkflowLateStatus
  Modifier, PNMsoft.Sequence.WindowsServices.BRS"
  interval="10"
/>
```

Configuration options

interval: The idle time of the thread between synchronizing the DB Listener activities. (Unit = minutes)

dbListener thread

This thread executes DB Listener activities. It checks for changes in the database table (defined in the activity) and initiates a workflow if the DB Listener activity is at the beginning of the workflow and a new record has been added. If the DB listener is in the middle of the workflow the process resumes if the record with the matching key has changed.

For more details on the DB Listener activity, click [here](#).

This thread is defined in this section:

```
<dbListenerThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.DatabaseListener,
  PNMsoft.Sequence.WindowsServices.BRS"
  interval="10"
/>
```

Configuration options

interval: The idle time of the thread between synchronizing the DB Listener activities. (Unit = minutes)

StatisticsCalculator thread

This thread calculates and updates the value of statistics in Cora SeSequence workflows.

For more details on defining statistics, click [here](#).

This thread is defined in this section:

```
<statisticsCalculatorThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.StatisticsCalculator, PNMsoft.Sequence.WindowsServices.BRS"
  interval="10"
/>
```

Configuration options

interval: The idle time of the thread between calculating statistics. (Unit = minutes).

Subscriber thread

This thread enables synchronization of multi BRS instances. By registering their status to the tblOnlineBRSInstances table this thread enables the instances to redistribute their work on workflow instances for redirect and move to close.

This thread is defined in this section:

```
<subscriberThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.Subscriber, PNMsoft.Sequence.WindowsServices.BRS"
  id="2F773C63-0D3C-46d5-A047-30E8E341B865"
  timeout="20"
/>
```

Configuration options

id: The ID of the BRS instance.

Note

Every BRS automatically gets a unique key during deployment.

timeout: If a BRS instance exceeds the timeout value (for example, 20 minutes), then the BRS concludes that this BRS instance is not active, and it redistributes work among the

other BRS instances. Also, when reaching half this time (for example, 10 minutes), the BRS instance notifies the other BRS instances that it is active. (Unit = minutes).

Note

You should NOT change any parameter of this thread or the table mentioned above, as it might prevent BRS from working correctly.

ExpiredTemporaryAttachmentsCleanerThread

When a user uploads an attachment to a form (even before hitting Submit), a temporary version of the attachment file is saved to the tblAttachmentsTemp table in the Database. This thread deletes these temporary attachments from this table.

This thread is defined in this section:

```
<expiredTemporaryAttachmentsCleanerThread
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.ExpiredTemporaryAttachmentsCleaner, PNMsoft.Sequence.WindowsServices.BRS"
  interval="60"
  expirationPeriod="04:00"
/>
```

Configuration options

interval: The idle time of the thread between cleaning up temporary attachments. (Unit = minutes).

expirationPeriod: the time after which the BRS deletes a temporary attachment on a form (Unit = hh:mm).

ClosedInstancesSeparator thread

This thread goes over all the closed and aborted workflow instances and moves all their metadata records to tables such as tblInstanceWorkflowsClosed, tblInstanceActivitiesClosed, among others.

This thread is defined in this section:

```
<closedInstancesSeparator
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.ClosedInstancesSeparator, PNMsoft.Sequence.WindowsServices.BRS"
  interval="10"
  pageSize="100"
  startAt="02:00"
  stopAt="04:00"
/>
```

Configuration options

interval: The idle time of the thread between checking for and moving closed workflow instance content. (Unit = minutes).

pageSize: This option enables the thread to work in chunks, handling x number of closed workflow instances at a time. This is relevant if you have a large number of workflow instances and want to optimize performance. (Unit: number of workflow instances).

startAt: The time of day the thread begins to work.

stopAt: The time of day the thread ends its work.

If you want the thread to work all the time, leave the startAt and stopAt options empty.

For more details on closing instances, click [here](#).

Note

It is recommended to set this thread to run at low peak times.

ClosedRecipientBreadCrumbsCleaner thread

Cora SeSequence saves breadcrumbs to track add recipient, remove recipient, and replace recipient actions on workflow tasks and messages. These breadcrumbs are saved to Database tables. This thread cleans up the tblRecipientsBreadcrumbs and the tblReminderBreadcrumbs tables for closed messages/tasks.

This thread is defined in this section:

```
<closedRecipientBreadCrumbsCleanerThread  
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.ClosedRecipientBreadCrumbsCleaner, PNMsoft.Sequence.WindowsServices.BRS"  
  interval="120"  
>
```

Configuration options

interval: The idle time of the thread between cleaning up the breadcrumb's tables. (Unit = minutes)

ReminderMessageCreator thread

This thread creates reminder messages as defined in the message wizard or the message instance.

For more details on reminders, [click here](#).

This thread is defined in this section:

```
<reminderMessageCreator  
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.ReminderMessageCre  
  ator, PNMsoft.Sequence.WindowsServices.BRS"  
  interval="10"  
>
```

Configuration options

interval: The idle time of the thread between creating reminders. (Unit = minutes).

stuckSubWorkflowActivityExcutorThread

This thread resumes the execution of a parent workflow that didn't continue beyond a completed sub workflow activity.

This thread is defined in this section:

```
<stuckSubWorkflowActivityExcutorThread  
  type="PNMsoft.Sequence.WindowsServices.BRS.BuiltInThreads.StuckSubWorkflowAc  
  tivityExcutor, PNMsoft.Sequence.WindowsServices.BRS"  
  interval="10"  
  pageSize="0"  
>
```

Configuration options

interval: The idle time of the thread between creating reminders. (Unit = minutes).

pageSize: You can use this option to redirect chunks of the workflow, especially if there are many instances and you are interested in improving performance.

For example, if set to 15, the thread redirects 15 workflow instances at a time. If set to 0, the thread redirects all workflow instances at once. (Unit = number of workflow instances).

EmailSenders

This section defines the 'From' address of emails which Cora SeSequence sends.

This configuration is defined in this section:

```
<emailSenders>  
  <defaultEmailSender email="sequence@pnmsoft.com" />  
  <!-- Example how to add e-mail sender  
  <workflowEmailSenders>  
    <add workflowId="" email=""></add>  
  </workflowEmailSenders>
```

```
-->  
</emailSenders>
```

Configuration options

defaultEmailSender email: define the default email sender for Cora SeQUENCE emails.
You can also add email senders per workflow using the example above (in [green](#))

Messaging

In Cora SeQUENCE there are several different types of messages: Normal, Rollback, Answer, Reminder, Comment, Delegation, and Question. Cora SeQUENCE enables you to configure the sender mode for these messages.

The rules defining which sender mode is used for messages are as follows:

You can set the sender mode in the **Message wizard > More Options** tab within the App Studio. There are several options in that tab, but if you choose "System Default", then the sender mode is set according to what is defined in the BRS config file.

In the BRS config file, if the sender mode is set to "Predefined", then messages are sent according to the <emailSenders> section of the BRS config file.

Other options you can enter here are

"WorkflowInstanceCreator"/"ActivityInstanceCreator"/"MessageSender".

The messaging configuration is defined in this section:

```
<messaging>  
<messageTypes>  
<!-- workflow message -->  
<add id="00000000-0000-0000-0000-000000000000" senderMode="Predefined" />  
<!-- rollback message -->  
<add id="E993B9F3-3A3F-4DE6-B38B-A086A1729CA1" senderMode="Predefined" />  
<!-- answer message -->  
<add id="67244068-B77E-410D-B26D-E34A607514FD" senderMode="Predefined" />  
<!-- comment message -->  
<add id="F883F524-2707-4959-8713-08B4D504B238" senderMode="Predefined" />  
<!-- delegation message -->  
<add id="100215E0-AB4C-4D11-BB7C-0FC313248895" senderMode="Predefined" />  
<!-- question message -->  
<add id="E966270A-8759-45CD-B222-3CAB5A7F0DB0" senderMode="Predefined" />  
</messageTypes>  
</messaging>
```

Configuration options

id : the message type id as detailed in the tblMessageBehaviorTypes table.

senderMode: Select the sender mode value for each message type. (Values: "Predefined", "WorkflowInstanceCreator", "ActivityInstanceCreator", "MessageSender").

Multiple BRS Deployment

It's possible to deploy multiple BRS instances on different servers (or on the same server) that will automatically load balance the work on the same Cora SeQUENCE environment.

Multiple BRS topology reduces the redirection and progress time of workflow instances, which is important for large capacity installations.

To deploy multiple BRS, follow the steps below:

1. Deploy a BRS app on the first server using the PowerShell function.
2. For additional BRS on the same server:
 - a. Deploy a second BRS app using the PowerShell function. Provide an instance number using the ApplicationID parameter.
 - b. Repeat a as many times you'll need.
3. For additional BRS on a different server, repeat step 1.
4. Run the Set-CoraSeQUENCEConnectionString function (on each server) to configure all apps connection string together.

Notes on multiple BRS deployment

- Avoid a blind copy of the same config file between BRS instances.
- Each BRS config file must have a unique Id in the subscriber thread.
- The dbListener thread does **not** support multiple BRS instance, so make sure this thread is enabled only on one of the BRS instances, and the rest of the instances must have this thread disabled (interval=0).
- In case of repeating dead lock errors related to the ClosedInstancesSeparator thread, consider minimizing the overlap between the start/end time of the different BRS instances.

Troubleshooting

Here are some troubleshooting techniques for common BRS issues:

Use the SVC log files

A generic tool that can help you with the troubleshooting is the workflow execution service trace file, also called the **SVC log file**.

This log is generated during runtime for every workflow execution and can be found in the Cora SeQUENCE server. It could be very helpful when investigating issues either on the runtime level or on the job host level.

Note that you must enable the logs in the JES config file and specify their path. You locate the relevant SVC log according to the workflow execution instance number and open it with the Microsoft Service Trace Viewer.

Important

Remember to stop the log file execution once you are done with the troubleshooting process since creating the SVCLOG files consumes server resources.

For all the configuration details of the SVC log file, click [here](#).

BRS stops

If the BRS stops unexpectedly, it might indicate a host environment issue. We recommend looking at the server logs to identify the root cause.

BRS won't start

If you try to start the BRS service and it doesn't start, there may be several reasons:

- The user that is defined to run the BRS during installation doesn't have permissions to run windows services.
- The BRS can't connect to the Cora SeQUENCE Database using the connection string you provided during deployment Either the connection string is wrong, or there is a problem with the Database server.
- The password of the user that is defined to run the BRS has changed.

Security exception in reminder thread

If you have received a security exception in the redirect or reminder thread, check that the user is a valid Cora SeQUENCE user.

Email sending issues

To troubleshoot email sending failures, check the server configuration and that the port is open on the server.

Another common mistake is that Emails are not defined for the activity.