Aras Innovator 31

Scheduler Service

Document #: *D-008127*

Last Modified: *1/23/2024*

Copyright Information

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Phone: 978-691-8900

E-mail: support@aras.com

Website: https://www.aras.com/

Table of Contents

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

Email:
TechDocs@aras.com
Subject: Aras Product Documentation

Or,

Postal service:
Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Attention: Aras Technical Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit https://www.aras.com/support

Document Conventions

The following table highlights the document conventions used in the document:

Convention	Description
Bold	This shows the names of menu items, dialog boxes, dialog box elements, and commands. Example: Click **OK**.
`Code`	Code examples appear in `courier` font. It may represent text you type or data you read.
`Yellow highlight`	Code highlighted in yellow draws attention to the code that is being indicated in the content.
`Yellow highlight with red text`	Red text highlighted in yellow indicates the code parameter that needs to be changed or replaced.
Italics	Reference to other documents.
Note:	Notes contain additional useful information.
Warning	Warnings contain important information. Pay special attention to information highlighted this way.
Successive menu choices	Successive menu choices may appear with a greater than sign (-->) between the items that you will select consecutively. Example: Navigate to **File** --> **Save** --> **OK**.

1 Overview

The Aras Innovator Service is a Windows executable program that runs as a Windows Service (see screen shot). Use it to automatically run Innovator Methods on a periodic schedule, as specified in the InnovatorServiceConfig.xml file.

You can configure a single running instance of the Aras Innovator Service to periodically connect to any number of Aras Innovator Servers, and for each Aras Innovator Server, run any number of Methods. Use the standard Windows Event Viewer for logging program status and error messages.

Examples of when to use the Aras Innovator Service are:

1) Running a Method that creates and sends Workflow reminder emails.

2) Running a Method nightly that creates a report and mails it to a set of users.

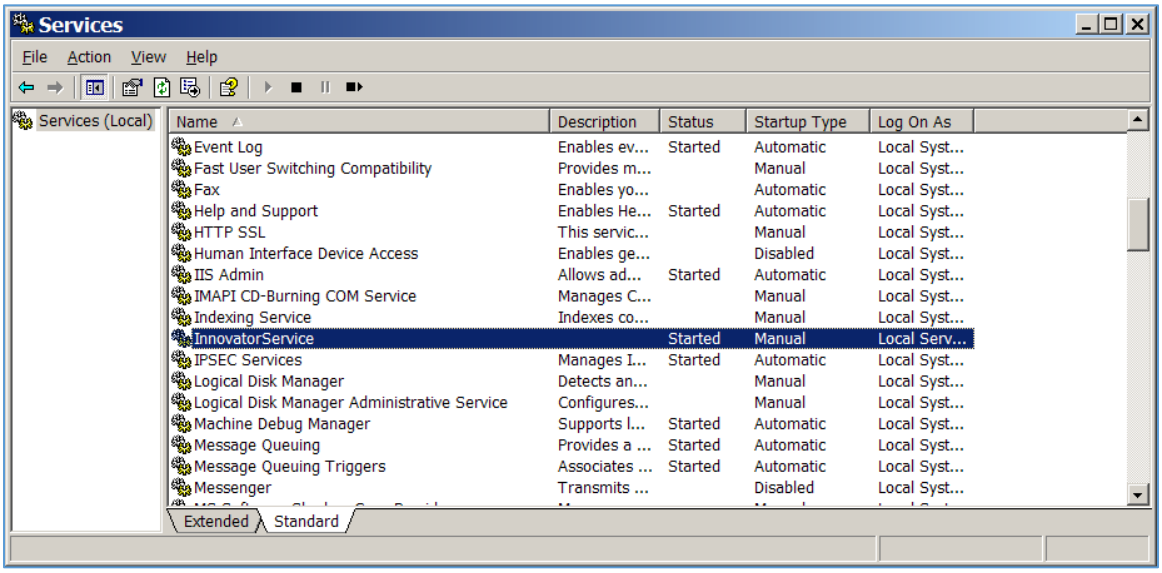

Figure 1.

1.1 Components

- `InnovatorService.exe`: a Windows-executable program
- `InstallService.bat`: a batch file to install the Innovator Service
- `rserv.bat`: run by `InstallService.bat`
- `UnInstallService.bat`: a batch file to un-install the Innovator Service
- `InnovatorServiceConfig.xml`: a configuration file
- `InnovatorService.exe.config`: a .NET configuration file

2 Installation and Configuration on Premise

2.1 Installation

The components listed in section 1.1 Components can be found on the Aras Innovator CD in the Scheduler folder.

1. Copy the InnovatorService.exe file and all files into any folder.
2. Copy the InnovatorServiceConfig.xml file into the same folder.
3. In the Folder selected in step 2, run the BAT file InstallService.bat. This program runs the standard .NET utility named INSTALLUTIL. This program leaves a set of .NET log files in the working directory which should be checked for error messages, and then deleted. You need to specify the .NET framework you are installing against in the command. This may need to be run as an administrator.

   (i.e., C:\InnovatorService\InstallService)
4. INSTALLUTIL prompts you for a windows authentication username and password for the machine on which you are installing the service. Use a fully qualified username like domain_name\user_name. If the username and password are not valid, the installation fails.
5. Verify that the Windows Service named 'Innovator Service' was added to the management console (see previous screen shot). Specify whether the service should be started automatically or manually. You can edit the username and password for the service from the LogOn tab in the service Properties.
6. Edit the following configuration file and then start the service.

Note: Use either the Windows Management console or the DOS command window to start and stop the InnovatorService.

```
NET START INNOVATORSERVICE
NET STOP INNOVATORSERVICE
```

2.2 Configuration

The InnovatorServiceConfig.xml file is analogous to a UNIX CronTab file, specifying which processes to run, and when they should be run.

Example configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<innovators>
   <innovator>
      <server>SERVER </server>
      <database>DATABASE</database>
      <username>USERNAME</username>
      <password>PASSWORD</password>
      <http_timeout_seconds>600</http_timeout_seconds>
      <certificate_path>CERTIFICATE_PATH</certificate_path>
      <certificate_password>PASSWORD</certificate_password>
      <certificate_store>STORE</certificate_store>
```

```
      <certificate_store_location>LOCATION</certificate_store_location>
      <certificate_find_type>TYPE</certificate_find_type>
      <certificate_find_value>VALUE</certificate_find_value>
      <certificate_valid_only>VALID_ONLY</certificate_valid_only>
      <client_id>CLIENT_ID</client_id>
      <assertion_token_lifetime>3600</assertion_token_lifetime>
      <job>
         <method>METHOD NAME</method>
         <months>MONTH STRING </months>
         <days>DAY STRING </days>
         <hours>HOUR STRING</hours>
         <minutes>MINUTE STRING</minutes>
      </job>
   </innovator>
   <eventLoggingLevel>LOGGING LEVEL</eventLoggingLevel>
   <intervalMinutes>INTERVAL</intervalMinutes>
</innovators>
```

2.2.1 Aras Innovator Server Connection Properties

- `<innovator>` tag specifies an Aras Innovator server to connect to. There can be multiple `<innovator>` tags in the configuration file.
- `<server>` is the HTTP URL of the server. This is the same URL that end-users use to run the standard Aras Innovator client.

  Example: `<server>http://localhost/InnovatorServer</server>`
- `<database>` is the ID of the database in the InnovatorServerConfig.xml. Note that this string is case sensitive.

  Example: `<database>InnovatorSolutions</database>`
- `<username>` is the login account that should be used for running the Methods.

  Example: `<username>admin</username>`
- `<password>` is the password for the login account, in plain text. The value can be encrypted by RsaCrypt tool.

  Example: `<password>innovator</password>`
- `<certificate_path>` is the path to an OAuth .pfx certificate file used for client connection.

  Example: `<certificate_path>D:\Certs\Scheduler.pfx</certificate_path>`
- `<certificate_password>` is the password for an OAuth .pfx certificate used for client connection. The value can be encrypted by RsaCrypt tool.

  Example: `<certificate_password>innovator</certificate_password>`
- `<certificate_store>` is the name of the certificates store (default is My).

  Example: `<certificate_store>My</certificate_store>`
- `<certificate_store_location>` is the location the certificates store (default is CurrentUser)

  Example:
  `<certificate_store_location>CurrentUser</certificate_store_location>`

- `<certificate_find_type>` is the type of the search criteria to find certificate in the specified store (default is FindBySubjectName).

  Example: `<certificate_find_type>FindByThumbprint</certificate_find_type>`

- `<certificate_find_value>` is the value of the search criteria to find certificate in the specified store.

  Example: `<certificate_find_value>Aras Innovator</certificate_find_value>`

- `<certificate_valid_only> indicates that only valid certificates can be selected from the store (default is False).`

  Example: `<certificate_valid_only>True</certificate_valid_only>`

- `<client_id> is ID of the OAuth client (default is Scheduler).`

  Example: `<client_id>Scheduler</client_id>`

- `<assertion_token_lifetime> lifetime of the token in seconds (default is 300).`

  Example: `<assertion_token_lifetime>300</assertion_token_lifetime>`

Note: OAuth Server side should be configured as well before using OAuth authentication. Below you can find basic examples of configuration changes.

Pair of OAuth certificates (.pfx and .cer) should be generated for Scheduler Service and an additional client registry should be added to the {InstallationDir}\OAuthServer\OAuth.config file:

```
<clientRegistry id="Scheduler" enabled="true">
  <secrets>
     <secret type="JwtBearerAssertionServerSecret">
      <certificate filePath="App_Data/Certificates/Scheduler.cer" />
     </secret>
  </secrets>
  <allowedScopes>
     <scope name="Innovator"></scope>
  </allowedScopes>
  <allowedGrantTypes>
     <grantType name="impersonate"></grantType>
  </allowedGrantTypes>
  <tokenLifetime accessTokenLifetime="3600"></tokenLifetime>
</clientRegistry>
```

2.2.2 Job Control Properties

For each `<innovator>`, there may be one or more `<job>` tags, specifying Methods that should be run. The * (asterix) is used as a wildcard in the scheduling tags.

- `<method>` is the Method name in the Innovator instance.

  Example: `<method>Workflow Reminders</method>`

- <months> is the month of the year that the Method should be run. Supports a comma delimited list from 1 through 12, where January is month=1 and December is month=12.

  Examples:

  - `<months>2,5,8,12</months>` Run only in Feb, May, Aug, and Dec
  - `<months>*</months >` Run every month

- <days> is the day of the week that the Method should be run. Supports a comma delimited list, where Sunday is day=0 and Saturday is day=6. A special value, `last`, is used to indicate the last day of the current month.

  Examples:

  - `<days>1, 5</days>` Run on Monday and Friday
  - `<days>*</days>` Run every day
  - `<days>last</days>` Run only on the last day of the month

- `<hours>` is the hours of the day that the Method should be run. Supports a comma delimited list of hour values from 0 through 23.

  Examples:

  - `<hours>6,17</hours>` Run 6am and again at 5pm
  - `<hours>*</hours>` Run every hour

- <minutes> is the minute of the hour that the Method should be run. Supports a comma delimited list of hour values from 0 through 59. A special value, `once`, is used to indicate the job should be run only once when the `<hours>` criteria is met, but on which minute is not important.

  Examples:

  - `<minutes>0,15,30,45</minutes>` Run every 15 minutes
  - `<minutes>*</minutes>` Run every minute
  - `<minutes>once</minutes>` Run only once when the `<hours>` pattern is met.

There is an implied tag of `<seconds>` with a value of `<seconds>once</seconds>`. All jobs are run only once within the minute that satisfies the scheduling criteria.

2.2.3 Aras Innovator Service Operating Properties

Two tags are used to control the behavior of the running InnovatorService process:

- `<eventLoggingLevel>` sets the level of detail for messages that are sent to the Windows Event monitor.

  Examples:

  - `<eventLoggingLevel>0</eventLoggingLevel>`—service startup announcement and error messages from Aras Innovator only.
  - `<eventLoggingLevel>1</eventLoggingLevel>`—announces the start of every Method run.
  - `<eventLoggingLevel>2</eventLoggingLevel>`—detailed reporting of every run cycle.

- `<intervalMinutes>` sets the Windows timer to control how often the InnovatorService is started and the <job> timing logic is tested. Any positive integer from 1 to 59 is supported.

  Example:

  - `<intervalMinutes>1</ intervalMinutes>`—the service is run every minute (recommended setting).

2.3 Configuration file example

The following example will run the three methods nightly:

- First, it runs the Send Email Reminders method to execute Workflow Activity Notification reminders at 00:02.
- Second, it runs the Check Escalations method to automatically escalate and designated Workflow Activities at 00:12.
- Finally, it runs the dailyUpdate method to update all active Project items in Aras Innovator at 00:22.

```
<?xml version="1.0" encoding="utf-8" ?>
<innovators>
  <innovator>
    <server>http://localhost/InnovatorServer</server>
    <database>InnovatorSolutions</database>
    <username>iservice</username>
    <password>innovator</password>
    <http_timeout_seconds>600</http_timeout_seconds>
    <job>
      <method>Send Email Reminders</method>
      <months>*</months>
      <days>*</days>
      <hours>0</hours>
      <minutes>2</minutes>
    </job>
    <job>
      <method>Check Escalations</method>
      <months>*</months>
      <days>*</days>
      <hours>0</hours>
      <minutes>12</minutes>
    </job>
    <job>
      <method>dailyUpdate</method>
      <months>*</months>
      <days>*</days>
      <hours>0</hours>
      <minutes>22</minutes>
    </job>
  </innovator>
  <eventLoggingLevel>0</eventLoggingLevel>
  <intervalMinutes>1</intervalMinutes>
</innovators>
```

2.4 Event Monitor

The standard Windows Event Monitor is used for logging. InnovatorService logs startup and shutdown messages. All error messages returned by the Aras Innovator Connection or by the Methods are also logged to the Event Monitor.

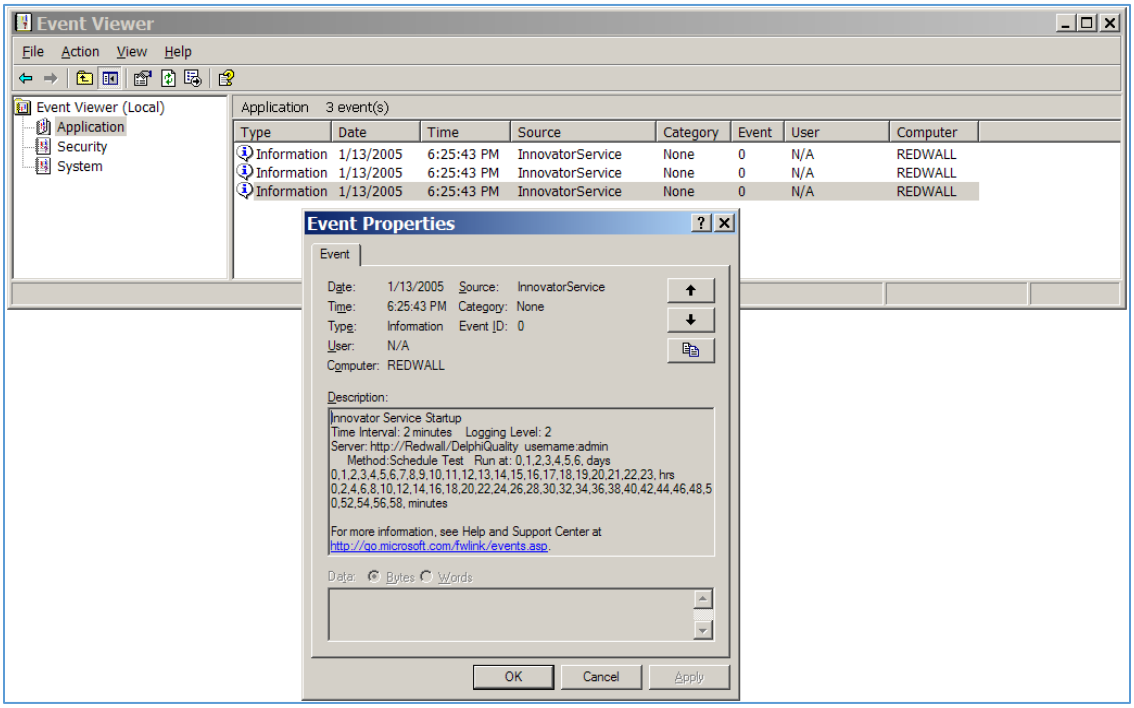

Figure 2.

The level of logging information displayed is configured in the InnovatorServiceConfig.xml file.

2.5 Scheduler Service Method Sample

The method executed by the Scheduler service must be a server-side method within Aras Innovator. Your method must return a result as shown in the following sample:

```
Innovator inn = this.getInnovator();
//set time of execution
Item schedServ = inn.newItem("variable","merge");
schedServ.setAttribute("where","[variable].name='Scheduler Execute
Time'");
schedServ.setProperty("name","Scheduler Execute Time");
schedServ.setProperty("value",DateTime.UtcNow.ToString());
schedServ=schedServ.apply();
//must return a new Result after execution of service
if(schedServ.isError())
{
   return inn.newResult("Scheduler Failed");
}
return inn.newResult("OK");
```

Note: It is recommended to make the Methods OS agnostic to use them with different operating systems. The main incompatibility issues in operating systems that need to be considered when implementing the method are related to file path case-sensitivity, file path separators, OS-specific line-endings, OS-specific code. For more information about cross-platform development please see section "`2.3 Cross-platform development`" in "`Aras Innovator 31 - Programmer's Guide`" document.