

The following table explains the properties of scheduled tasks. To configure tasks:

1. Open the **Scheduled tasks** application.
2. Edit (✎) the task.

Property	Description
Task display name	Sets a name for the task that appears in the administration interface.
Task name	Serves as a unique identifier for the scheduled task, for example in the API. You can leave the ( <i>automatic</i> ) option to have the system generate an appropriate code name based on the display name.
Task provider	<p>Selects the class that implements the scheduled task.</p> <ul style="list-style-type: none"> <li>• <b>Assembly name</b> - specifies the name of the assembly where the task is implemented. Choose the (<i>custom classes</i>) option for tasks implemented in the App_Code folder.</li> <li>• <b>Class</b> - specifies the exact class (including any namespaces) that defines the functionality of the scheduled task.</li> </ul>
Period Start time Every Between Days	<p>This group of properties sets the time interval between the executions of the task.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• The interval settings do not ensure that the system executes the task at the exact time, only that the task is considered ready for execution. The exact run time depends on the global scheduling settings. See also: <a href="#">Configuring scheduled task execution</a></li> <li>• The actual execution time and run duration do not affect the scheduling of the next run time. The system calculates the next run time based on the time for which the last run was originally scheduled (or the start time).</li> <li>• If a task is still being processed when the next scheduled run time occurs (for example with long-running tasks that have a short execution interval), the task skips the given run and waits until the next run time.</li> </ul>
Task data	Additional data provided to the task's class. You can access the field from the code of the task, and use the value as a parameter. The Task data allows you to modify the task's functionality without having to edit the code implementation.
Task condition	<p>Allows you to enter an additional macro condition that must be fulfilled in order for the scheduler to execute the task.</p> <p>You can write any condition according to your specific requirements. For details about available macro options and syntax, refer to the <a href="#">Macro expressions</a> chapter.</p>
Task enabled	Indicates if the system executes the task.
Delete task after last run	Indicates if the system deletes the task after its final run (applicable if the task is set to run only once).
Run task in separate thread	<p>Indicates if the scheduler executes the task in a separate thread to improve application performance.</p> <p>It is not possible to access context data (information about the current page, user, etc.) in the code of tasks that are executed in a separate thread.</p>

Use external service	<p>If enabled, the task is processed by the <a href="#">Installing the Scheduler Windows service</a> instead of the web application. If the <b>Use external service</b> setting in <b>Settings -&gt; System</b> is disabled, even tasks with this option enabled are processed by the application itself.</p> <p>Only some of the default scheduled tasks support this option. The tasks that do not have the option available in the editing interfaces must be processed by the application itself.</p> <p>You cannot define the task in the <i>App_Code</i> folder if you wish to use the external service. To run a custom task externally, you must add a new assembly to your project and then define the task class there.</p>
Run individually for each site	<p>Only available for <b>global</b> tasks. If enabled, the scheduler executes the task repeatedly as a sitespecific task, once for each running site in the system. The task automatically runs within the context of the corresponding site.</p> <p>This can be useful if you wish to manage a task in a single location instead of creating a separate one for every site.</p>
Server name	<p>Sets the name of the web farm server where the task is executed. This field is applicable only if your application is running in a <a href="#">Web farm environment</a>.</p> <p>To add a new task for all web farm servers currently registered in the system, select the <b>Create tasks for all web farm servers</b> check box below the field.</p>
Use context of user	<p>If the scheduled task needs to access data from the user context in its code (for example to check permissions), you can use this property to choose which user is provided. The scheduler always executes the task within the context of the selected user.</p> <p>In most cases, the user context does not affect the functionality of the task and you can leave the <i>(default)</i> option. The <i>(default)</i> option means either the context of a public user, or the user from the context of the request during which the system executed the task (when using the <a href="#">request-based scheduler mode</a>).</p>
Executions	<p>Shows how many times the task has been executed. You can reset this counter back to 0 by clicking <b>Reset</b>.</p>