Prioritizing Messages

You can prioritize messages to ensure that high-priority messages are not dropped. Prioritization is implemented at two levels in WSO2 ESB:

- **HTTP transport level** - If users would like to use the ESB as a pure router.
- **Message mediation level** - If users use ESB for heavy processing like XSLT and XQuery.

From the users perspective, key to any priority mediation is to determine the priority of an incoming message.

At the message mediation layer, this can be done using content filters. This means the full power of ESB configuration language is available to the user for determining the priority of a given message. For example, a message may contain an element called "priority" and depending on its value the priority can be determined.

At the HTTP layer, user has access to HTTP headers, HTTP parameters and URL values. By looking at these values, user can determine the priority of a given message.

The priority mediation implementation is based on **Queues and ThreadPoolExecutors**.

**ThreadPoolExecutor** accepts a **BlockingQueue** implementation. A custom blocking queue that can be used to order the jobs based on priority was implemented. **ThreadPoolExecutor** starts queuing only when all the core threads are busy. Every message should get equal priority until all the core threads are used.

Internally custom **BlockingQueue** uses multiple queues for accepting jobs with different priorities. Once jobs are put into the queue, it uses a pluggable algorithm for choosing the next job. The default algorithm chooses the jobs based on a priority-based, round-robin algorithm. For example, let's say we have two priorities, 10 and 1. This algorithm tries to fetch 10 items with priority 10 and then 1 item with the priority 1.

**Priority Executors**

**Priority executors** can be used with the **Enqueue Mediator** to execute sequences with a given priority. Priority executors are used in high-load scenarios when you want to execute different sequences for messages with different priorities. This approach allows you to control the resources allocated to executing sequences and to prevent high-priority messages from getting delayed and dropped. A priority has a valid meaning comparing to other priorities specified. For example, if there are two priorities with value 10 and 1, a message with priority 10 will get 10 times more resources than messages with priority 1.

**Priority Executor Configuration**

```
<priority-executor name="priority-executor">
    <queues isFixed="true|false" nextQueue="class implementing NextQueueAlgorithm">
        <queue [size="size of the queue"] priority="priority of the messages put in to this queue"/>
    </queues>
    <threads core="core number of threads" max="max number of threads'
    keep-alive="keep alive time"/>
</priority-executor>
```

Core priority executors' attributes:

- **queues** - Defines separate queues for different priorities in a Thread Pool Executor.
- **isFixed** - Controls the queues to have a fixed depths or un-bounded capacities.
An executor should have at least two or more queues. If only one queue is used, there is no point in using a priority executor.

- **size** - Defines a size of a queue.
- **priority** - Defines a priority of a queue.

If the queues have unlimited length, no more than core threads will be used.

- **core** - Defines a core number of Priority Executor threads. When ESB is started with the priority executor, this number of threads will be created.
- **max** - Defines the maximum number of threads this executor will have.
- **keep-alive** - Defines the keep-alive time of threads. If the number of threads in the executor exceeds the core threads, they will be in active for the keep-alive time only. After the keep-alive time, those threads will be removed from the system.

The following topics describe how to manage your priority executors:

- Adding a Priority Executor
- Editing a Priority Executor
- Deleting a Priority Executor

To see a sample of priority-based mediation, see Sample 652: Priority Based Message Mediation.