



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Composer Help

Using URS and ORS Functions

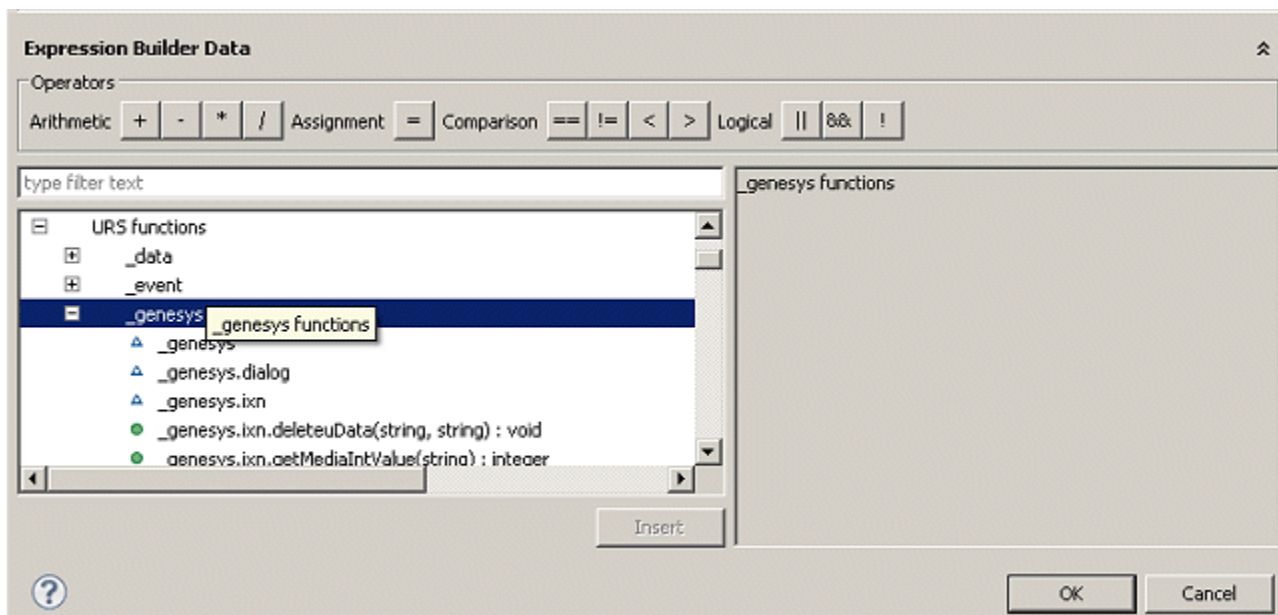
Using URS and ORS Functions

Composer's **Expression Builder** allows you to use SCXML elements and extensions as described the **Orchestration Server Developer's Guide** as well as **Universal Routing Server (Workflow) functions**.

Expression Builder opens from many blocks; to name a few:

- Assign--**Assign Data Property**
- Branching--**Conditions Property**
- ECMAScript (for workflows)--**Script Property**
- Entry--**Variables Property**
- Log--**Logging Details Property**
- Looping--**Exit Expression Property**

Using one or more of the above blocks in a workflow gives access to functions for SCXML-based strategies. The `_genesys` data subcategory is where you will find modules that access Genesys-platform related objects, properties, and functions.



Below is summary information on certain functions accessible via Expression Builder.

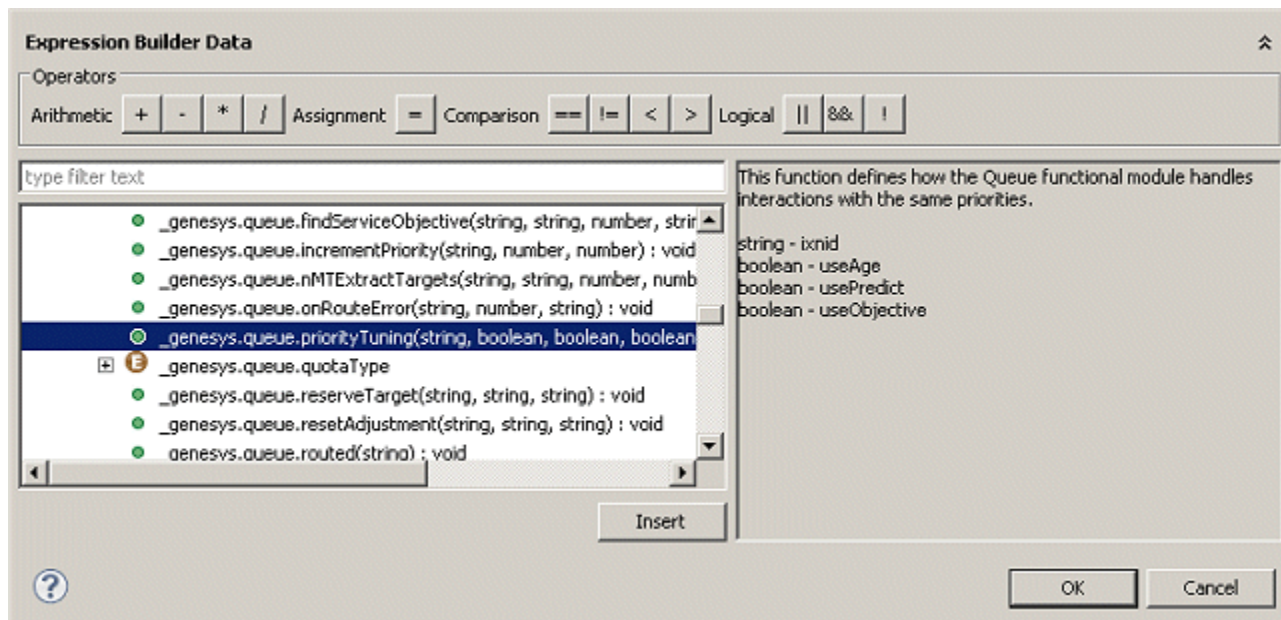
Interaction Priority

URS always puts interactions into waiting queues according to their priorities. The `priorityTuning`

function defines how URS handles interactions with the same priorities. By default, interactions with the same priority are ordered according to the time the interaction began to wait for some target.

- `useAge` parameter (true or false). If true, URS uses the time the interaction was created instead of the time interaction is placed into the waiting queue. Age of interaction is usually the time that URS starts the strategy for the interaction.
- `usePredict` parameter (true or false). If true, then URS calculates the estimated time for the interaction to be answered and will use this time instead of the time that the interaction has already waited.
- `useObjective` parameter (true or false). This parameter works with the URS `use_service_objective` option (*Universal Routing 8.1 Reference Manual*), which allows you to scale the time interval for processing different interactions. If this option set to true, URS scales the time interval that the interaction waits (or is going to wait) according to the Service Objective of the interaction

If two interactions wait the same time, such as 60 seconds, but interaction #1 has a Service Objective of 30 seconds and interaction #2 has a Service Objective of 120 seconds, then URS puts interaction #1 ahead of interaction #2 in the queue ($60/30 > 60/120$). The figure below shows how to access the `priorityTuning` function in Expression Builder.



- For more detailed information on using this function, see the *Universal Routing 8.1 Reference Manual* and the *Universal Routing 8.1.Routing Application Configuration Guide* located on the [Universal Routing Wiki](#).
- For information on Orchestration Server functions, see Orchestration Extensions and Core Extensions topics in the [Orchestration Developers Guide](#).