

Appendix

Conversion Data Model

This section describes the complete data model of the File Conversion feature.

ConversionServer

The 'ConversionServer' ItemType is where the URL for the Conversion Server is stored. In this regard, it is similar to the Vault ItemType.

Properties

- Name: Unique name for the Conversion Server.
For example, this could describe what types of Conversions are handled by this Conversion Server, i.e. 'Cad Native Server'
- URL: HTTP path to the '/ConversionServer/ConversionService.asmx' file
Impersonation User: Aras Innovator User that is used to perform the conversion. All Files created by the Conversion Server have their 'Created_by' as this User.

Important

The Impersonation User either must be the Vault Admin User or must be a member of the Administrators Identity.

Relationships

- Converters ('ConversionServerConverterType'): Relationship to 'ConverterType' Used to select which Converters are executed using this Conversion Server. The 'ConversionServerPriority' ItemType is a Relationship from the 'Vault' ItemType to the 'ConversionServer' ItemType. This Relationship specifies which Conversion Server to use based on the Vault that the File was added to or updated in.

ConverterType

The 'ConverterType' ItemType uses the name of the Converter to represent the Converter DLL inside of Aras Innovator.



Properties

- Name: This name must match the name attribute set in the 'Converter' tag of either the '\ConversionServerConfig.xml' file as follows:

```
<ConversionServer>
<Converters>
<Converter name="MyConverter" ... />
</Converters>
</ConversionServer>
```

ConversionRule

The 'ConversionRule' ItemType is where the rules and events for the conversion are defined.

Properties

- Name: Unique name to describe what the Rule does using the Conversion Server.
- Conversion Type: Points to the 'ConverterType' Item that performs the Conversion.
- Description (optional): Detailed description of the Conversion Rule
- Timeout: Number of minutes after which the actual converting of the physical file is considered as hung and the conversion task is marked as failed.
In other words this is the time allowed to the OnConvert event handler to return a result; suggested default is 60 minutes
- Delay: The delay in minutes after which a failed task can be rerun; suggested default is 5 minutes
- Cutoff: The number of hours that is given to a failed Conversion Task to finish successfully; If the Conversion Task has status Failed for the specified number of hours, the conversion task is marked as discarded and removed from the processing by the framework. The suggested default value is 24.
- Enabled: Boolean to enable or disable the Conversion Rule.

Relationships

- File Types (ConversionRuleFileType): Relationship to 'FileType'
Used to specify which types of files trigger this Conversion Rule. Each FileType can be enabled or disabled using the 'Enabled' property on the Relationship.



- Rule Events (ConversionRuleEventHandler): Relationship to 'Method'
Used to specify Server-side methods that are run prior to the creation of the Conversion Task. (Event Type = OnTaskCreating)
These methods can then be used to check certain conditions such as the File size and prevent the Conversion Task from being created.
- Task Event Templates(ConversionRuleEventTemplate) : Relationship to 'Method'
 - Used to specify Server-side methods that are run at different times while the Conversion Task is processing.
 - Conversion mode - blocking vs non-blocking (default):
Non-blocking is intended to process long-running tasks (more than 30 minutes); the Innovator Server sends a request to the Conversion Server to execute a Task, and the Conversion Server notifies the Innovator Server about completion, so the Innovator doesn't wait for completion, which avoids the possibility of a timeout.
 - Available Events (In order of execution):
 - OnStartTaskProcessing: Fired right after the conversion task is created and the processing cycle gets the task for processing
 - OnBeforeConvert: Fired right before the conversion task is ready to be sent for conversion
 - OnConvert: Overrides the conversion event
 - OnAfterConvert: Fired on the processing cycle iteration after the framework sets result to ConversionTaskResult

ConversionTask

'ConversionTask' Items are created automatically based on enabled Conversion Rules. Each 'Conversion Task' represents one File being converted by a Conversion Server. All properties and relationships are automatically populated.

Properties

- Status: Current status of the Task. Can be 'Not Started', 'In Progress', 'Succeeded', 'Failed', or 'Discarded'
- Rule: Points to the Conversion Rule that was used for the creation of this Conversion Task
- File Type: The type of File that is being converted
- File: The ID of the File being converted
- Started On: Date that the Conversion Task started
- Finished On: Date that the Conversion Task finished



- User Data: Text (most probably XML) that could be used by a specific converter to carry the converter specific data that has to be eventually passed to the process that performs actual conversion of the physical file (e.g. conversion options; list of supplementary files that have to be given to the converter in order to convert the file; etc.)
- Error: Error description in case the task status is Failed or Discarded

Relationships

- Dependencies ('ConversionTaskDependency'): Null Relationship

Displays the Type and ID of Items that can be locked by methods during the processing of the Conversion Task.

- Results ('ConversionTaskResult'): Null Relationship
Displays the File IDs that have been created as a result of the Conversion Task and an additional string 'kind' to possibly describe why it was created / what it was created for.
- Event Handlers ('ConversionTaskEventHandler'): Relationship to 'Method'
 - Events copied from the 'Task Event Templates' Relationship on the 'Conversion Rule' ItemType.
 - Displays the status of each Event during the processing of the Conversion Task with the following properties:
 - Error [...]: Error message set if the Method failed
 - User Data [...]: Used by the event handler to pass some information between different invocations of the event handler (e.g. in case the event handler failed it could save in this property some information about what it managed to finish before it failed so that the next invocation of the method does not try to do it again).
 - Started On [...]: Date when the first invocation of the method started
 - Finished On [...]: Date when the last invocation of the method finished
 - Execution Attempt: The number of times the Method has been called by this Conversion Task

Status: Current status of the Event. This can be 'Not Started', 'In Progress', 'Failed', or 'Succeeded'