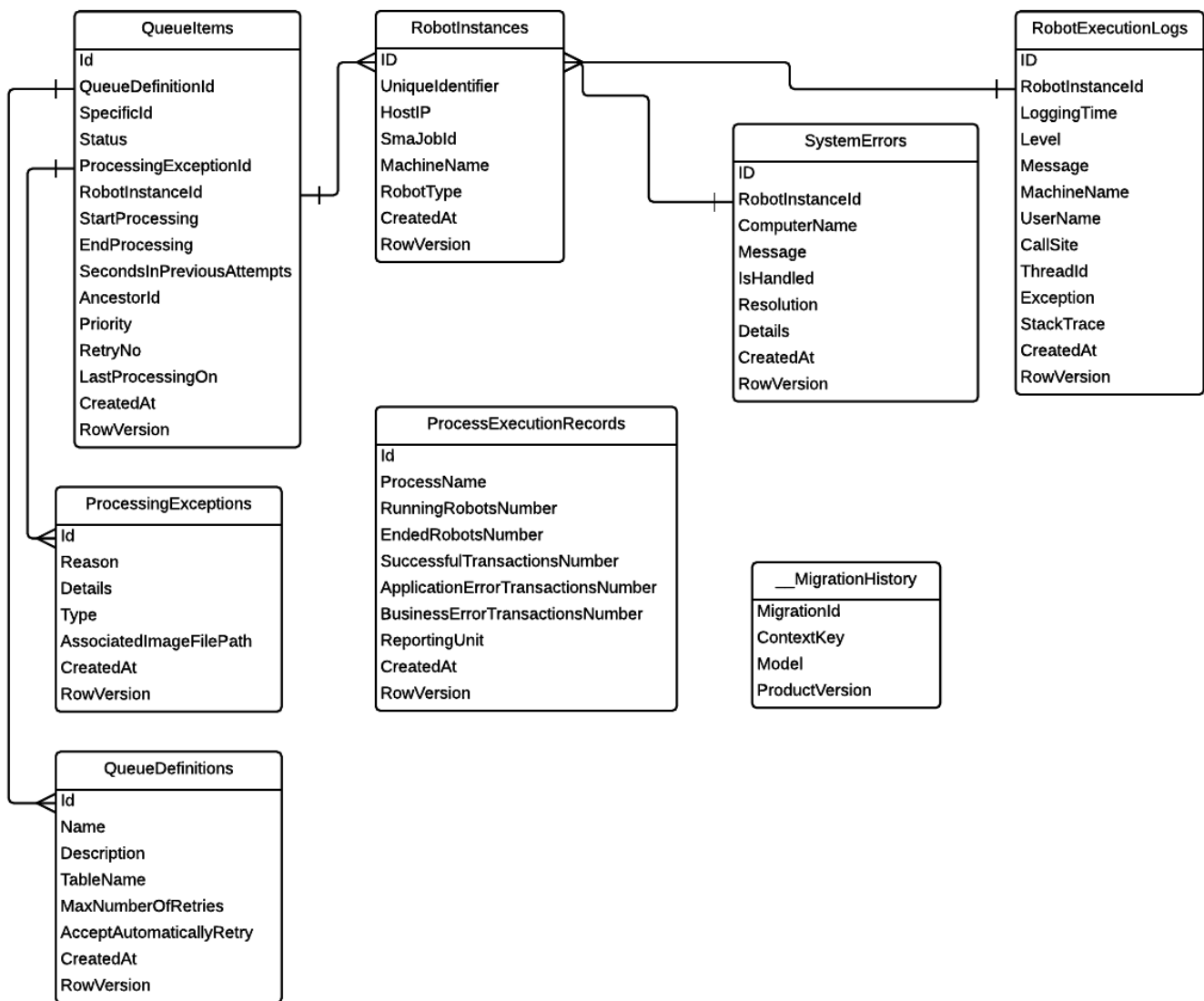


# ControlCenter Database Diagram



## ControlCenter Database Description

1. **RobotInstances** table - this table is used for registering robots when starting.
  - UniqueIdentifier - For each robot will be assigned a unique identifier, different each time (a sort of Session ID)
  - HostIP - Robot's IP address. It is provided by robot.
  - SmaJobId - This field stores job id from SMA used by Runbook when starting a robot.
  - MachineName - This field stores the computer name where the robot runs.
  - RobotType - This field stores the name of the process. This should be one of the names stored in the robotTypes setting found in the configuration file (Web.config) from Platform Services
  - CreatedAt - This field stores the date and time when the row was inserted.
  - RowVersion - This field is used for versioning. Any object from Domain Model has a version. It is used for solutioning concurrent access.

2. **RobotExecutionLogs** table - this table is used for storing messages generated by robots.
  - RobotInstaceld - Foreign Key id with RobotInstances.ID.
  - LoggingTime - It stores timestamp value received from robot.
  - Level - It stores value received from robot (a string with one of the following values: Information, Trace, Warning, Error, Verbose).
  - Message - It stores message text received from robot.
  - Machine - It stores machine name (computer name).
  - UserName - It stores user's name that was used by the server to open a Windows session on the client computer.
  - CallSite - This field is unused.
  - ThreadId - This field is unused.
  - Exception - This field stores exception (error) messages.
  - StackTrace - This field stores the .NET stack trace in case of error.
  - CreatedAt - This field stores date when row was inserted.
  - RowVersion - This field is used for versioning. Any object from Domain Model has a version. It is used for solutioning concurrent access.
  
3. **SystemErrors** table - In this table are stored errors that can occur in SMA when robot runbook starts or when robot is initialized. There is no data in this table because "Executor" or "Transactor" runbooks have not been written in such a way to access the web service from Platform Services in order to send the error.
  - RobotInstaceld - This field stores Foreign Key with RobotInstances Id field table. It can be null, also.
  - ComputerName - This field stores computer name.
  - RowVersion - This field is used for versioning.
  
4. **\_\_MigrationHistory** table - This table is created by Entity Framework when DB is created in order to keep model hash key.
  - MigrationId -
  - ContextKey -
  - Model -
  - ProductVersion - This field is used for versioning.
  
5. **Queue.QueueDefinitions** table is used to hold the definition of a queue - name, table name and a few specific parameters. Every time a user creates a new queue (materialized by a new table) they need to add a new record in the Queues.QueueDefinitions table
  - Id - Primary Key, autoincremented
  - Name - the name of the queue; should be the same name as the process which consumes data from the queue
  - Description - free text
  - TableName - should contain the name of the table that was created to store data specific to the queue. This field should contain the full name of the queue table (including the "Queues" prefix), like Queues.ClaimProcessing
  - MaxNumberOfRetries - if the queue supports retrying of failed transaction items, here should be specified the maximum number of retries before the transaction item is abandoned
  - AcceptAutomaticallyRetry - boolean field (0 or 1) to specify whether a failed transaction item should be retried or not
  - CreatedAt - This field stores date when row was inserted.

- RowVersion - This field is used for versioning. Any object from Domain Model has a version. It is used for solutioning concurrent access.

Let's assume the user wants to create a queue for the process ClaimProcessing. There is already a process ClaimProcessingExecutor which is found in Administration -> Processes in UiPath Control Center. There are 2 runbooks in SMA, ClaimProcessingExecutor and ClaimProcessingTransactor. There is a value "ClaimProcessing" in the robotTypes variable (which is a comma separated list of automation processes) in the web.config file of the PlatformServices (editable in IIS Manager). Now the user needs to create a new table to store the data for this queue and add this table name in the Queues.QueueDefinitions table.

The table that the user creates:

- needs to be called Queues.ClaimProcessing
- should have an ID field, of type BigInt, autoincrement, Primary Key
- all the other fields are added to serve the user's needs to store information about a transaction that needs to be processed by the robots.

After the table Queues.ClaimProcessing is created, a row should be added in Queues.QueueDefinitions with the following values:

- Name = ClaimProcessing
- Description = anything that is relevant
- TableName = Queues.ClaimProcessing
- MaxNumberOfRetries = positive integer (can be 0)
- AcceptAutomaticallyRetry = 0 or 1 (1 to enable other robots to retry a failed transaction item)
- CreatedAt = current date in the form "yyyy-mm-dd" (time is not necessary, but can be filled after the date plus a space, in the form "hh:mm:ss")

6. **Queue.QueueItems** table stores common information about all the transaction items from all the queues.

- Status - This field is used for storing status. Please check Status table for each status description.

Status	Description
0	New
1	In Progress
2	Failed
3	Success
4	Invalid
5	Retried

- QueueDefinitionId - Foreign Key to Queues.QueueDefinitions.ID to identify the queue.
- SpecificId - a value from the ID column of a queue table. Because this table stores information for all the queue items for all the queues, one queue item is identified by QueueDefinitionId and by the SpecificId
- ProcessingExceptionId - if the transaction item failed (the robot called the activity SetTransactionStatus with status = Failed) then the reason of the failure is inserted in ProcessingExceptions and the corresponding

ProcessingExceptions.ID is stored here. If ProcessinExceptionId is not NULL, then you should expect to find one of the following values in Status: 2 (Failed), 5 (Retried), 4 (Invalid)

- RobotInstancelId - Foreign Key to RobotInstances.ID
- StartProcessing - This field stores the time when GetTransactionItem was executed by the robot.
- EndProcessing - This field stores the time when SetTransactionStatus was executed by the robot for the current try.
  - If Transaction fails with ApplicationException, another robot will retry this Transaction and sets up on this line "SecondsInPreviousAttempts". It will then create a new row on which it will set up AncestorId and RetryNo fields, also. Before each retry a verification is made in order to see if maximum number of retries has been reached.
- AncestorId - is not NULL in case the current row is related to a QueueItem that is retried. If not null, it indicates the ID of another row from the same table that failed on the first attempt.
- Priority - This field stores priority values. It is setup when an item is added to the queue. Values for priority are: Low, Normal, High. For GetTransactionItem in UiPath Studio, priority is used only if it's activated from config (PlatformServices - > usePriority).
- RetryNo - number of retry if this row was created as a result of a failed transaction item that was retried.
- CreatedAt - This field stores date when row was inserted.
- RowVersion - This field is used for versioning. Any object from Domain Model has a version. It is used for solutioning concurrent access.

7. **ProcessingExceptions** table stores the exceptions sent by the robots when SetTransactionStatus was declared Failed.

- Reason - This field stores data sent by robot using the Reason attribute of the SetTransactionStatus activity.
- Details (optional) - This field stores data sent by robot.
- Type - This field stores error types. ( 0 = ApplicationException, 1 = BusinessException)
- AssociatedImagePath - This field stores local path where screen snapshot was saved if the "Take screenshot" activity was used by the robot. Value sent by the robot.

8. **ProcessExecutionRecords** table stores aggregated data about the running processes.

- ProcessName - This field stores process name.

In order to understand the rest of the fields, ReportingUnit should be explained first.

- ReportingUnit - This field can have 2 values
  - 0 - for the rows that are created every 1 minute (the rest of the fields contain numbers from the last 60 seconds of execution).
  - 1 - for the rows that are created every 60 minutes (the rest of the fields contain aggregated / totalized numbers from the last 60 minutes of execution).
  - At each minute process state is saved for each running process. After 120 rows are created for each running process, the oldest 60 entries will be deleted. In conclusion will be 1 entry for the previous hour and 60 entries for the last 60 minutes. After another hour, another record with totals for one hour will be created. If a process runs for 5 hours and 26 minutes, we'll find with that process name 5 records with total numbers for each of the 5 hours, plus 26 records for the 26 minutes.

- RunningRobotsNumber - This field stores maximum robots number per the respective hour or the respective minute.
- EndedRobotsNumber - This field stores how many robots stopped or completed execution per the respective hour or the respective minute.
- SuccessfulTransactionsNumber - how many successful transactions per the respective hour or the respective minute
- ApplicationErrorTransactionsNumber - how many transactions failed with ApplicationException per the respective hour or the respective minute
- BusinessErrorTransactionsNumber - how many transactions failed with BusinessException per the respective hour or the respective minute