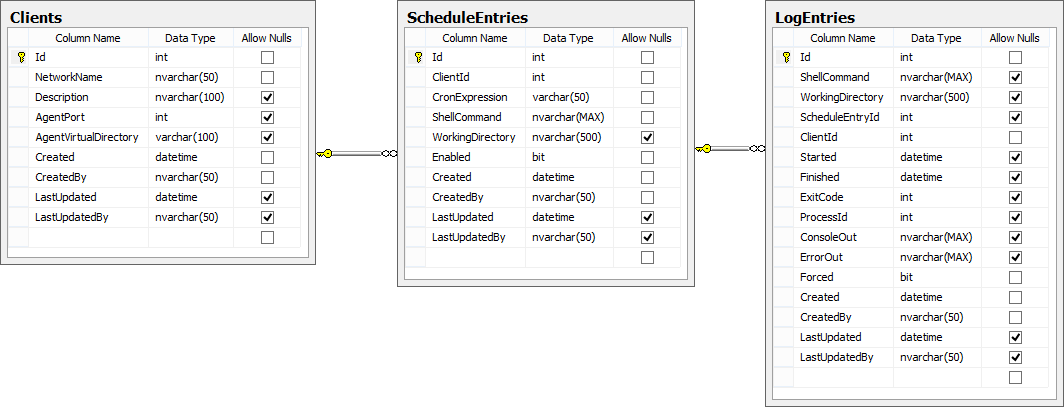
Scheduler

Design Document

# Data Model View

## Entity-Relation Diagram



## Data Dictionary

### Clients Table

The Clients table holds records of client machines in the network that the scheduler can contact and send shell commands for execution.

|  |  |
| --- | --- |
| Column Name | Description |
| Id | Table identity. Primary Key. Auto-incremental. |
| NetworkName | Name of the client machine in the network. Also known as host name. Typically a DNS or an IP address. |
| Description | Short informative text that helps the user identify the client machine. |
| AgentPort | Port number in this client machine used by the AgentService. Default value is 8733. |
| AgentVirtualDirectory | Part of the AgentService URI path. The virtual directory comes after the protocol and port, and before the service name. Must not include trailing slash. For example: http://localhost:8733/<AgentVirtualDirectory>/Scheduler/Agent.svc. |
| Created | Audit field. Date and time (UTC) when this record was created (inserted in the table). |
| CreatedBy | Audit field. Name of user that created this record. |
| LastUpdated | Audit field. Last date and time (UTC) when this record was changed. |
| LastUpdatedBy | Audit field. Name of user that made the last update to this record. |

### ScheduleEntries Table

The ScheduleEntries table holds records of shell commands that are scheduled for execution at client machines.

|  |  |
| --- | --- |
| Column Name | Description |
| Id | Table identity. Primary key. Auto-incremental. |
| ClientId | Foreign key to Clients table. Id of the target machine of this schedule entry. |
| CronExpression | A cron expression that the scheduler that defines when this shell command will be executed. Cron expressions are strings consisting of five components separated by whitespace. Each string component defines scheduling criteria for a particular date/time part. Such parts are:   * Minute (of each hour) * Hour (of each the day) * Day (of each month) * Month (of each year) * Day of the week   Each component’s value can be a positive integer, or an asterisk. Asterisks represent any and all possible values in the date part domain, for example 0 to 59 for the minute component, 0 to 23 for the hour. Weekdays range from 0 to 6, Sunday being 0, Monday 1 and so on.  At a given moment, the cron expression will be evaluated by the scheduler and will result in a positive match if all expression components match those of the current date and time.  For example: given system’s current date as Sunday, Jan 10, 2016 7:38 AM, the following cron expressions will be evaluated as follows:   |  |  |  | | --- | --- | --- | | \* \* \* \* \* | MATCH | Expression will match every time as all components are wildcards. This results in one execution every minute. | | 0 \* \* \* \* | IGNORE | Minute part does not match current time’s minute part. Expression will run once per hour, at minute zero (2:00 AM/PM, 3:00 AM/PM, etc.) | | \* 7 \* \* \* | MATCH | Expression is set up to run every minute, every day, from 7:00 AM to 7:59 AM. |   For more information about cron expressions, visit [the Wikipedia](https://en.wikipedia.org/wiki/Cron). |
| ShellCommand | The shell command to be executed by the client machine. The syntax will depend on the client machine’s operating system and the expression is executed exactly as typed. |
| WorkingDirectory | The directory in the client machine’s disk where this command will run at. |
| Enabled | A flag indicating whether this entry is to be evaluated by the scheduler for execution. |
| Created | Audit field. Date and time (UTC) when this record was created (inserted in the table). |
| CreatedBy | Audit field. Name of user that created this record. |
| LastUpdated | Audit field. Last date and time (UTC) when this record was changed. |
| LastUpdatedBy | Audit field. Name of user that made the last update to this record. |

### LogEntries Table

The LogEntries table holds detailed records of all command executions.

|  |  |
| --- | --- |
| Column Name | Description |
| Id | Table identity. Primary Key. Auto-incremental. |
| ShellCommand | Same as ShellCommand column in ScheduleEntries table. |
| WorkingDirectory | Same as WorkingDirectory column in ScheduleEntries table. |
| ScheduleEntryId | A foreign key to the ScheduleEntries table that helps identify the schedule entry that generated this execution and log record. |
| ClientId | A foreign key to the Clients table. The Id of the client machine where this command execution took place. |
| Started | Date and time (UTC) when this command execution started. A NULL value indicates either that the command has been launched but has not yet started, or that execution failed to start at the client machine. |
| Finished | Date and time (UTC) when this command execution finished. A NULL value indicates either that the command has been launched but has not yet finished, or that execution failed to start at the client machine. |
| ExitCode | An integer value that the process returned to the operating system at the client machine as a result of the command execution. Typically, an exit code of 0 (zero) indicates success. |
| ProcessId | An integer value generated by the client operating system that identifies the process in the client machine that executed this command. |
| ConsoleOut | The information sent by the process to the standard output stream. This will normally show the output of the command execution. |
| ErrorOut | The information sent by the process to the standard error stream. |
| Forced | A flag indicating whether this was a manual execution (true / checked) or a scheduled execution (false / unchecked). |
| Created | Audit field. Date and time (UTC) when this record was created (inserted in the table). |
| CreatedBy | Audit field. Name of user that created this record. |
| LastUpdated | Audit field. Last date and time (UTC) when this record was changed. |
| LastUpdatedBy | Audit field. Name of user that made the last update to this record. |

# Components View

## Components

|  |  |  |
| --- | --- | --- |
| Component name | Description | Main technologies applied |
| Scheduler.­Web | Main user interface for operators and admins. Responsibilities include:   * CRUD of client machines * CRUD of schedule entries * Manual execution of shell commands | * MVC 5 * EntityFramework |
| Scheduler.‌SchedulerService | WCF Service that handles communication with agents and cron service. Main responsibilities:   * Send agents execution orders * Update execution log * Send Cron service reload order | * WCF * EntityFramework |
| Scheduler.‌AgentService | WCF Service running in client machines. Responsibilities include:   * Command execution at client machines. * Reporting execution results back to SchedulerService | * WCF |
| Scheduler.‌CronService | Windows Service. Main responsibilities:   * Monitor system time * Evaluate schedule conditions * Send SchedulerService execution orders | * EntityFramework |
| Scheduler.‌SchedulerService.‌Cient | Helper component that handles WCF communication with SchedulerService. | * WCF |
| Scheduler.‌AgentService.‌Client | Helper component that handles WCF communication with AgentService. | * WCF |
| Scheduler.DataContracts | Assembly that holds data contracts and model entities. | * WCF * EntityFramework |
| Scheduler.ServiceContracts | Assembly that holds WCF service contracts | * WCF |
| Scheduler.DataAccess | Component that provides database access functionality. | * EntityFramework |
| Scheduler.Common | Helper methods used by most components |  |

## Dependency Diagram

The following diagram shows dependencies among the main solution components.



Arrows indicate dependency (“depends on”). Gray boxes represent external packages.

The Scheduler.Common package was omitted for brevity, as it contains auxiliary code only and almost all other packages depend on it.

# Deployment View

The solution is composed of the following main packages:



Each vertical block constitutes a cohesive and independent deployment package. Packages can be deployed to dedicated servers, or servers can play more than one role in the system.

Recommended production deployment layout for Scheduler is as follows:



The following diagram depicts a scenario where each component is deployed to a dedicated server:



Another possible deployment layout is one where all packages are deployed to a single server (suitable for testing and development workstations):



# Communication View

## Communication Protocols

The following diagram shows communication calls and protocols:



Arrows indicate communication calls and go from originators (clients) to receivers (servers). Clients send requests to servers. Arrows show a legend indicating the communication protocol used.

Scheduler is mostly based on Windows Communication Foundation (WCF) for inter-process communication. In the case of the “WCF” protocol, it actually means that the communication is based on WCF and, as such, the specific protocol for that communication can be specified using WCF configuration. By default, the solution uses HTTP/SOAP messages (BasicHttpBinding).

TDS stands for Tabular Data Stream. A Microsoft proprietary protocol for communicating with SQL Server.

RPC stands for Remote Procedure Call, a Microsoft proprietary protocol for inter-process communication in Windows. It is used by the SchedulerService component to communicate with CronService, which is a Windows Service.

## Communication Messages

Services expose operations for consumers to invoke. Clients invoke such operations sending messages to services, and get a response back from them. The following shows the messages that components send to each other:



## Message descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Originator | Receiver | Name | Description |
| User browser | Web | HTTP/S requests | Typical HTTP requests that browsers send to web servers. Normal HTML traffic is send and received. |
| Web | SchedulerService | Execute | Web application requests a manual execution.  Message properties:   |  |  |  | | --- | --- | --- | | scheduleEntry | ScheduleEntry | A ScheduleEntry entity that contains all information relevant for execution. | | forced | bool | A flag set to True, indicating this is a forced (or manual) execution as opposed to a scheduled execution. | |
| Web | SchedulerService | Reload | A schedule entry has just been modified by the user so Web notifies ScheduleService to reload schedule entries. |
| SchedulerService | AgentService | Execute | ScheduleService instructs AgentService to start a command execution immediately. This can be a consequence of a previous message from CronService, or due to a manual execution message that came from Web.  Message properties:   |  |  |  | | --- | --- | --- | | logEntryId | int | Log entry identifier. ScheduleService has created a log entry for this execution and passes its identifier to the agent so that agent can update its status. | | shellCommand | string | The shell command to execute. | | workingDirectory | string | The initial directory in the client machine where command will execute. | |
| ShedulerService | CronService | Reload | A schedule entry has just been modified by the user so Web notified ScheduleService to reload schedule entries. Now ShedulerService forwards the message to its final destination. |
| AgentService | SchedulerService | UpdateLogEntry | Command execution has finished on the client machine and now the agent requests SchedulerService to log the outcome in the database. |
| CronService | SchedulerService | ExecuteMany | CronService has just evaluated all schedule entries and determined that some (or all) of them meet the execution criteria. So it sends SchedulerService a list of all schedule entry identifiers that are to be immediately executed. |
| \* | \* | SQL | SQL queries that read or write database information. |

# Security View

Pending

## ----- o -----