# ERLANG ADD-IN USER MANUAL

Version 4.0

CCMATH

NEXT-LEVEL WFM

## TABLE OF CONTENTS

# 1. ABOUT THIS ADD-IN

CCmath is a highly specialized software company that offers solutions for the most challenging contact-center setups. Please contact us at info@ccmath.com or visit our website at www.ccmath.com.

CCmath's Erlang Calculator is a one-of-a-kind Excel add-in for workforce management professionals. You can calculate your metrics in Erlang C, Erlang X, Blending, and Chat models, all within your own worksheets using our custom functions — providing a full capacity calculation suite with all the functions and parameters needed. CCmath is the only supplier of the Erlang Blending and Chat add-ins, and the first supplier of Erlang X, worldwide.

The add-in contains functions written in the very fast C++ programming language, related to resource planning for contact centers. This manual covers version 4.0 of the Erlang engine.

# 2. ERLANG MODELS AND FUNCTION SETS

Each function set was developed by specialists in academia based on theoretical Erlang models:

- The **Erlang C** function set is the classic model for inbound call center staffing: customers queue and wait until an agent becomes available (infinite patience assumed). In this add-in, Erlang C calculations are performed by calling the Erlang X functions without providing the Patience parameter — the model then falls back to standard Erlang C behaviour automatically.
- The **Erlang X** extends the Erlang C model with parameters for customer patience, abandonment, and redials, providing much higher accuracy for real-world contact centers.
- The **Erlang Blending** function set calculates staffing when agents handle both inbound and outbound work. A configurable threshold protects inbound accessibility.
- The **Erlang Chat** function set extends the Erlang model for digital channels where agents can handle multiple conversations in parallel simultaneously.

# 3. INSTALLATION METHODS

The offline version of the Erlang add-in operates directly on your local computer, eliminating the need for data storage on external servers. As a result, no data will be stored or transmitted to CCmath when using the offline version. All computations and processing occur locally, ensuring that your data remains within your control and providing an additional layer of privacy and security. With the offline version, you can confidently work with your Erlang functions without any data leaving your local environment.

Both installation methods below give you the same add-in with identical functionality. The difference is only in how your license is delivered and how the files are placed on your machine.

### Self-registration (licenses.ccmath.com)

Purchase and manage your license at licenses.ccmath.com. During registration you select how many machines your license covers. Once registered, download the installer (`.exe`) from the portal. Run it — the installer automatically detects your Windows bit version (32-bit or 64-bit), downloads the correct files, and places them in the right locations. No manual file handling is required.

### Manual license (via CCmath support)

If the `.exe` installer does not work on your machine — for example due to company firewall policies, restricted user directories, or IT security settings that block executable downloads — please contact CCmath at info@ccmath.com. We will supply the required files directly. This option is also available for special enterprise or custom licensing arrangements.

You will receive:

- The `.xla` add-in file
- The DLL file
- A license file
- A demo sheet
- This user manual

Place the DLL file and the license file in the `C:\Windows` directory, or in any folder listed in the system's PATH environment variable. If you do not have administrator rights to do this, see Problem 4 in the Troubleshooting section.

## 4. EXCEL SETUP AND MACROS

Once the add-in files are installed, some Excel settings must be configured. Go to *Office button* → *Excel Options* → *Trust Center* → *Trust Center Settings* → *Macro Settings*, select **Disable all macros with notification**, and press **OK**.

There are two methods for integrating the Erlang functions into your own workbook:

### Method 1 — Open the demo sheet

Open the demo sheet, delete all example sheets, and insert your own data. The advantage is that function autocomplete works. The disadvantage is that you cannot use your own VBA code because the sheet is protected.

### Method 2 — Load the add-in into an existing workbook

Start from an existing document and load the `.xla` file as an add-in. The steps depend on your version of Excel:

- **Excel 2003:** Go to *Tools* → *Add-Ins* → *Browse* to the `.xla` file location, enable it, and press OK.

- **Excel 2007 and higher:** Go to *Start* → *Excel Options* → *Add-Ins* → *Manage* → *Go*, browse to the file, select it, and enable it.

The disadvantage of Method 2 is that function autocomplete does not work. To insert a function, use the *Insert Function* button, select *User Defined Functions*, and choose the desired function. The main advantage is that you can use your own VBA code.

## 5. FUNCTION PARAMETERS AND CONVENTIONS

All function parameters are described below. **The time unit of all parameters must be the same.** For example, if Volume is expressed per minute then AHT, TTA, Patience, and ASA must all also be expressed in minutes.

> **Optional parameters** are marked with an asterisk (*). When an optional parameter is left empty, all parameters to its right must also be left empty.
>
> **Error returns:** If input parameters are invalid, the cell will display `"Invalid parameters"`. If there is a license problem, the cell will display `"Invalid license"`.

| Parameter | Description |
|---|---|
| `Volume` | The average number of arrivals per unit of time. (Volume ≥ 0) |
| `AHT` | The average handling time of a contact. (AHT > 0) |
| `Agents` | The number of agents. Can be a real (non-integer) number. (Agents ≥ 0) |
| `TTA` | The *time to answer* — the maximum allowed waiting time used to define the service level. The service level is the percentage of contacts served within the TTA. The time unit is the same as the other parameters and is not necessarily in seconds. (TTA ≥ 0) |
| `Patience` | The average time a customer is willing to wait in the queue before abandoning. A practical estimate: total waiting time (including abandoned calls) divided by the number of abandonments; filter out extreme values in advance. (Patience ≥ 0) |
| `RetrialPercentage` | The probability that a customer who abandons will redial. (0 ≤ RetrialPercentage ≤ 1) |
| `ServiceLevel` | The expected service level — the fraction of contacts answered within the TTA. (0 < ServiceLevel < 1) |
| `ASA` | The average speed of answer — the average waiting time an arbitrary customer with infinite patience would incur. (ASA > 0) |
| `Occupancy` | The fraction of time agents spend handling contacts. Used as an input target in AGENTS_OCCUPANCY functions. (0 < Occupancy < 1) |
| `Abandonment` | The probability that an arbitrary customer will abandon the queue. (0 < Abandonment < 1) |

| Parameter | Description |
|---|---|
| `BlockingProbability` | The expected probability that a customer is blocked (all lines busy). (0 < BlockingProbability < 1) |
| `SLDefinition` | A flag to switch between service level calculation modes. 0 = virtual service level; 1 = answered; 2 = offered (default). |
| `Variance` | The variance of the forecast error — the difference between actual arrivals and the forecast. (Variance ≥ Volume) |
| `Threshold` | In the blending model: the number of agents kept idle before outbound calls are taken. When Threshold equals Agents, all capacity is reserved for inbound. (Threshold ≤ Agents) |
| `Outbound` | In the blending model: the average number of outbound calls per unit of time, used when finding required staffing. (Outbound ≥ 0) |
| `OutboundAHT` | In the blending model: the average handling time of outbound calls. (OutboundAHT > 0) |
| `Parallel` | In the chat model: the maximum number of chats one agent can handle simultaneously. Must be a whole number. (Parallel ≥ 1) |

# 6. ERLANG C FUNCTIONS

The classic Erlang C model assumes customers have infinite patience — they wait indefinitely until an agent becomes available. It is the standard baseline model for inbound call center staffing.

There are no separate Erlang C functions in this add-in. Erlang C is accessed by calling the Erlang X functions and **leaving the Patience parameter empty**. The model automatically switches to the Erlang C algorithm when Patience is omitted.

The functions are organised in two groups:

1. **Performance metrics** — given a known number of agents, calculate outcomes such as service level, waiting time, or occupancy.
2. **Finding required agents** — given a performance target, calculate the number of agents needed to achieve it.

## Performance metrics (Agents known)

```
X_SLA(Volume, AHT, Agents, TTA)
```

Calculates the expected service level.

```
X_ASA(Volume, AHT, Agents)
```

Calculates the average speed of answer (average waiting time).

```
X_OCCUPANCY(Volume, AHT, Agents)
```

Calculates the occupancy of the agents.

```
X_VOLUME(ServiceLevel, AHT, Agents, TTA)
```

Calculates the maximum average number of arrivals per unit of time such that the service-level objective is satisfied.

### Finding required agents

```
X_AGENTS_SLA(ServiceLevel, Volume, AHT, TTA)
```

Calculates the optimal number of agents such that the service-level objective is satisfied.

```
X_AGENTS_ASA(ASA, Volume, AHT)
```

Calculates the optimal number of agents such that the average-waiting-time objective is satisfied.

```
X_AGENTS_OCCUPANCY(Occupancy, Volume, AHT)
```

Calculates the optimal number of agents such that the occupancy objective is satisfied.

> These are the same functions as in the Erlang X section. Patience is simply left empty, which tells the model to use the Erlang C algorithm. When Patience is filled in, the same function becomes Erlang X. Functions that calculate abandonment or retrial rates (`X_ABANDON`, `X_AGENTS_ABANDON`, `X_RETRIALRATE`) always require Patience and have no Erlang C equivalent.

## 7. ERLANG X FUNCTIONS

The Erlang X model extends Erlang C by modelling impatient customers who may abandon and potentially redial. Optional parameters are marked with *. When Patience is omitted in functions that support it, the calculation falls back to Erlang C (see Erlang C above).

The functions are organised in two groups:

1. **Performance metrics** — given a known number of agents, calculate outcomes such as service level, waiting time, occupancy, abandonment rate, or blocking probability.
2. **Finding required agents** — given a performance target, calculate the number of agents needed to achieve it.

### Performance metrics (Agents known)

```
X_SLA(Volume, AHT, Agents, TTA, Patience*, RetrialPercentage*, SLDefinition*, Variance*)
```

Calculates the expected service level.

```
X_ASA(Volume, AHT, Agents, Patience*, RetrialPercentage*)
```

Calculates the average speed of answer (average waiting time).

```
X_OCCUPANCY(Volume, AHT, Agents, Patience*, RetrialPercentage*)
```

Calculates the occupancy of the agents.

```
X_ABANDON(Volume, AHT, Agents, Patience, RetrialPercentage*, Variance*)
```

Calculates the fraction of customers that abandon the queue. Patience is required.

```
X_RETRIALRATE(Volume, AHT, Agents, Patience, RetrialPercentage*)
```

Calculates the average number of customers that retrial per unit of time. Patience is required.

```
X_BLOCKING(Volume, AHT, Agents)
```

Calculates the fraction of customers that are blocked (Erlang B model). No patience parameter.

```
X_VOLUME(ServiceLevel, AHT, Agents, TTA, Patience*, RetrialPercentage*, SLDefinition*)
```

Calculates the maximum average number of arrivals per unit of time such that the service-level objective is satisfied.

## Finding required agents

```
X_AGENTS_SLA(ServiceLevel, Volume, AHT, TTA, Patience*, RetrialPercentage*,
SLDefinition*, Variance*)
```

Calculates the optimal number of agents such that the service-level objective is satisfied.

```
X_AGENTS_ASA(ASA, Volume, AHT, Patience*, RetrialPercentage*)
```

Calculates the optimal number of agents such that the average-waiting-time objective is satisfied.

```
X_AGENTS_OCCUPANCY(Occupancy, Volume, AHT, Patience*, RetrialPercentage*)
```

Calculates the optimal number of agents such that the occupancy objective is satisfied.

```
X_AGENTS_ABANDON(Abandonment, Volume, AHT, Patience, RetrialPercentage*, Variance*)
```

Calculates the optimal number of agents such that the abandonment objective is satisfied. Patience is required.

```
X_AGENTS_BLOCKING(BlockingProbability, Volume, AHT)
```

Calculates the optimal number of agents such that the blocking objective is satisfied (Erlang B model). No patience parameter.

# 8. ERLANG BLENDING FUNCTIONS

In the blending model, agents handle both inbound calls and outbound work. When an agent becomes free, inbound calls take priority. An agent will only take outbound work when at least *Threshold* other agents are idle. When Threshold equals Agents, all capacity is reserved for inbound calls.

**Patience is required** for all blending functions — the model always accounts for customer abandonment. All parameters are required unless stated otherwise.

The blending functions are organised in three groups:

1. **Performance metrics** — given known agents and threshold, calculate outcomes.
2. **Agent-finding** — given a performance target and outbound workload, find required agents.
3. **Threshold-finding** — given known agents and a performance target, find the required threshold.

## Performance metrics (Agents and Threshold known)

```
BL_SLA(Volume, AHT, Agents, TTA, Patience, Threshold)
```

Calculates the expected inbound service level in a blending model.

```
BL_ASA(Volume, AHT, Agents, Patience, Threshold)
```

Calculates the average speed of answer in a blending model.

```
BL_OCCUPANCY(Volume, AHT, Agents, Patience, Threshold)
```

Calculates the occupancy of the agents in a blending model.

```
BL_OUTBOUND(Volume, AHT, Agents, Patience, Threshold, OutboundAHT)
```

Calculates the average number of outbound calls handled per unit of time in a blending model.

```
BL_ABANDON(Volume, AHT, Agents, Patience, Threshold)
```

Calculates the fraction of inbound customers that abandon in a blending model.

## Agent-finding (performance target and Outbound workload known)

```
BL_AGENTS_SLA(ServiceLevel, Volume, AHT, TTA, Patience, Outbound, OutboundAHT)
```

Calculates the optimal number of agents such that the service-level objective is satisfied, given a known outbound workload.

---

```
BL_AGENTS_ASA(ASA, Volume, AHT, Patience, Outbound, OutboundAHT)
```

Calculates the optimal number of agents such that the average-waiting-time objective is satisfied, given a known outbound workload.

---

```
BL_AGENTS_ABANDON(Abandonment, Volume, AHT, Patience, Outbound, OutboundAHT)
```

Calculates the optimal number of agents such that the abandonment objective is satisfied, given a known outbound workload.

---

```
BL_AGENTS_OCCUPANCY(Occupancy, Volume, AHT, Patience, Outbound, OutboundAHT)
```

Calculates the optimal number of agents such that the occupancy objective is satisfied, given a known outbound workload.

### Threshold-finding (Agents and performance target known)

```
BL_THRESHOLD_SLA(Volume, AHT, Agents, Patience, ServiceLevel, TTA)
```

Calculates the threshold such that the inbound service-level objective is satisfied for the given number of agents.

---

```
BL_THRESHOLD_ASA(Volume, AHT, Agents, Patience, ASA)
```

Calculates the threshold such that the average-waiting-time objective is satisfied for the given number of agents.

---

```
BL_THRESHOLD_ABANDON(Volume, AHT, Agents, Patience, Abandonment)
```

Calculates the threshold such that the abandonment objective is satisfied for the given number of agents.

---

```
BL_THRESHOLD_OCCUPANCY(Volume, AHT, Agents, Patience, Occupancy)
```

Calculates the threshold such that the occupancy objective is satisfied for the given number of agents.

---

```
BL_THRESHOLD_OUTBOUND(Volume, AHT, Agents, Patience, Outbound, OutboundAHT)
```

Calculates the threshold such that the outbound workload objective is satisfied for the given number of agents.

# 9. ERLANG CHAT FUNCTIONS

The Erlang Chat model is designed for digital channels — live chat and messaging — where agents handle multiple conversations simultaneously. The key distinguishing feature is the **Parallel** parameter: a whole number specifying how many chats one agent can handle at the same time. AHT is a single value (the average handling time per chat session).

> **Example:** If an agent can handle up to 3 chats simultaneously, set `Parallel = 3`. Combined with the AHT of a single chat session, the model automatically accounts for the effect of concurrency on staffing.

**Patience is required** for all chat functions. All parameters are required.

The functions are organised in two groups:

1. **Performance metrics** — given a known number of agents, calculate outcomes such as service level, waiting time, occupancy, or abandonment rate.
2. **Finding required agents** — given a performance target, calculate the number of agents needed to achieve it.

## Performance metrics (Agents known)

```
CHAT_SLA(Volume, Parallel, AHT, Agents, TTA, Patience)
```

Calculates the expected service level for a chat channel.

---

```
CHAT_ASA(Volume, Parallel, AHT, Agents, Patience)
```

Calculates the average speed of answer for a chat channel.

---

```
CHAT_OCCUPANCY(Volume, Parallel, AHT, Agents, Patience)
```

Calculates the occupancy of agents handling chats.

---

```
CHAT_ABANDON(Volume, Parallel, AHT, Agents, Patience)
```

Calculates the fraction of customers that abandon the chat queue.

## Finding required agents

```
CHAT_AGENTS_SLA(ServiceLevel, Volume, Parallel, AHT, TTA, Patience)
```

Calculates the optimal number of agents such that the service-level objective is satisfied for a chat channel.

---

```
CHAT_AGENTS_ASA(ASA, Volume, Parallel, AHT, Patience)
```

Calculates the optimal number of agents such that the average-waiting-time objective is satisfied for a chat channel.

```
CHAT_AGENTS_OCCUPANCY(Occupancy, Volume, Parallel, AHT, Patience)
```

Calculates the optimal number of agents such that the occupancy objective is satisfied for a chat channel.

```
CHAT_AGENTS_ABANDON(Abandonment, Volume, Parallel, AHT, Patience)
```

Calculates the optimal number of agents such that the abandonment objective is satisfied for a chat channel.

# 10. TROUBLESHOOTING

### Problem 1: You are unable to register

**Solutions:**

a. Please make sure you have an active internet connection.

b. Add the document as a permitted program in your firewall's or antivirus software's settings. If it is already listed, try removing it and adding it again.

### Problem 2: Cell displays "Invalid license"

*What does this mean:* Your registration has expired or you are not registered yet.

**Solutions:**

a. If the registration has expired, go to licenses.ccmath.com or contact CCmath for payment options.

b. If you have not registered yet, register using the Registration tab in the demo sheet.

### Problem 3: Cell displays "Invalid parameters"

*What does this mean:* One or more input parameters are outside the allowed range or their combination is infeasible.

**Solutions:**

a. Please check that each parameter is within its allowed range. Common causes:

- **X_SLA, X_ASA, X_OCCUPANCY and their AGENTS_ variants:** The Agents parameter must not be zero (Agents ≠ 0). For X_ASA and X_SLA, Volume × AHT must be less than Agents (rounded down) — if Volume × AHT ≥ Agents, the queue is unstable and waiting time is infinite.
- **X_ABANDON, X_AGENTS_ABANDON, X_RETRIALRATE:** Patience is required and must be greater than 0.
- **X_AGENTS_SLA:** ServiceLevel must be strictly between 0 and 1 and achievable given the input parameters.
- **BL_ functions:** Threshold must be less than or equal to Agents. Patience must be greater than 0.
- **CHAT_ functions:** Parallel must be a whole number ≥ 1. Patience must be greater than 0.

b. If you cannot resolve the error, please contact CCmath with the function name and parameter values you have used.

**Example — X_ASA returns "Invalid parameters":**

We want to calculate the average speed of answer with Volume = 4 (per minute), AHT = 3 (minutes), and Agents = 10. Because Volume × AHT = 12 > 10 (Agents), the system is overloaded and the queue never empties. Increase Agents to at least 13 (the first integer above 12) to obtain a valid result.

**Example — X_SLA returns "Invalid parameters" with Patience:**

We want to calculate the service level with Volume = 4, AHT = 3, Agents = 11, TTA = 0.333 (minutes), and Patience = 0.01 (minutes). The Patience value is extremely small — customers abandon almost instantly. Check that Patience is expressed in the same time unit as AHT (minutes here) and represents a realistic average wait tolerance.

### Problem 4: You cannot copy the DLL or license file into C:\Windows *(manual license only)*

**Solutions:**

a. Make sure you have administrator rights on the machine.

b. Ask your system administrator to install the files for you.

c. Alternatively, place the files in any other folder that is listed in the system's PATH environment variable.

If you received your license via licenses.ccmath.com, use the installer instead — it handles file placement automatically without requiring administrator rights.

# 11. CHEAT SHEET

## Parameters

| Parameter | Description |
|---|---|
| `Volume` | Avg arrivals per unit of time (≥ 0) |
| `AHT` | Avg handling time (> 0) |
| `Agents` | Number of agents, can be non-integer (≥ 0) |
| `TTA` | Time to answer — max wait for service level (≥ 0) |
| `Patience` | Avg time customer waits before abandoning (≥ 0) |
| `RetrialPercentage*` | Probability an abandoning customer redials (0–1) |
| `ServiceLevel` | Target fraction answered within TTA (0–1) |
| `ASA` | Target avg speed of answer (> 0) |
| `Occupancy` | Target fraction of time agents handle contacts (0–1) |
| `Abandonment` | Target fraction of customers abandoning (0–1) |
| `SLDefinition*` | 0 = virtual SL \| 1 = answered \| 2 = offered (default) |
| `Variance*` | Variance of forecast error (≥ Volume) |
| `Threshold` | Blending: agents kept idle before outbound (≤ Agents) |
| `Outbound` | Blending: avg outbound calls per unit of time (≥ 0) |
| `OutboundAHT` | Blending: avg handling time of outbound calls (> 0) |
| `Parallel` | Chat: max simultaneous chats per agent, integer (≥ 1) |
| `BlockingProbability` | Target blocking probability — Erlang B (0–1) |

## Erlang C and X Functions Erlang C = Erlang X without Patience

| Function | Signature | Returns |
|---|---|---|
| `X_SLA` | Volume, AHT, Agents, TTA, Patience*, RetrialPercentage*, SLDefinition*, Variance* | Expected service level |
| `X_ASA` | Volume, AHT, Agents, Patience*, RetrialPercentage* | Avg speed of answer |
| `X_OCCUPANCY` | Volume, AHT, Agents, Patience*, RetrialPercentage* | Agent occupancy |

| Function | Signature | Returns |
|---|---|---|
| `X_ABANDON` | Volume, AHT, Agents, Patience, RetrialPercentage*, Variance* | Abandonment fraction |
| `X_RETRIALRATE` | Volume, AHT, Agents, Patience, RetrialPercentage* | Retrials per unit of time |
| `X_BLOCKING` | Volume, AHT, Agents | Blocking probability (Erlang B) |
| `X_VOLUME` | ServiceLevel, AHT, Agents, TTA, Patience*, RetrialPercentage*, SLDefinition* | Max arrivals meeting SL target |
| `X_AGENTS_SLA` | ServiceLevel, Volume, AHT, TTA, Patience*, RetrialPercentage*, SLDefinition*, Variance* | Agents to meet SL target |
| `X_AGENTS_ASA` | ASA, Volume, AHT, Patience*, RetrialPercentage* | Agents to meet ASA target |
| `X_AGENTS_OCCUPANCY` | Occupancy, Volume, AHT, Patience*, RetrialPercentage* | Agents to meet occupancy target |
| `X_AGENTS_ABANDON` | Abandonment, Volume, AHT, Patience, RetrialPercentage*, Variance* | Agents to meet abandonment target |
| `X_AGENTS_BLOCKING` | BlockingProbability, Volume, AHT | Agents to meet blocking target (Erlang B) |

### Erlang Blending Functions Patience required for all

| Function | Signature | Returns |
|---|---|---|
| `BL_SLA` | Volume, AHT, Agents, TTA, Patience, Threshold | Inbound service level |
| `BL_ASA` | Volume, AHT, Agents, Patience, Threshold | Avg speed of answer |
| `BL_OCCUPANCY` | Volume, AHT, Agents, Patience, Threshold | Agent occupancy |
| `BL_OUTBOUND` | Volume, AHT, Agents, Patience, Threshold, OutboundAHT | Outbound calls per unit of time |
| `BL_ABANDON` | Volume, AHT, Agents, Patience, Threshold | Inbound abandonment fraction |
| `BL_AGENTS_SLA` | ServiceLevel, Volume, AHT, TTA, Patience, Outbound, OutboundAHT | Agents to meet SL target |
| `BL_AGENTS_ASA` | ASA, Volume, AHT, Patience, Outbound, OutboundAHT | Agents to meet ASA target |
| `BL_AGENTS_ABANDON` | Abandonment, Volume, AHT, Patience, Outbound, OutboundAHT | Agents to meet abandonment target |

| Function | Signature | Returns |
|---|---|---|
| `BL_AGENTS_OCCUPANCY` | Occupancy, Volume, AHT, Patience, Outbound, OutboundAHT | Agents to meet occupancy target |
| `BL_THRESHOLD_SLA` | Volume, AHT, Agents, Patience, ServiceLevel, TTA | Threshold to meet SL target |
| `BL_THRESHOLD_ASA` | Volume, AHT, Agents, Patience, ASA | Threshold to meet ASA target |
| `BL_THRESHOLD_ABANDON` | Volume, AHT, Agents, Patience, Abandonment | Threshold to meet abandonment target |
| `BL_THRESHOLD_OCCUPANCY` | Volume, AHT, Agents, Patience, Occupancy | Threshold to meet occupancy target |
| `BL_THRESHOLD_OUTBOUND` | Volume, AHT, Agents, Patience, Outbound, OutboundAHT | Threshold to meet outbound target |

## Erlang Chat Functions Patience and Parallel required for all

| Function | Signature | Returns |
|---|---|---|
| `CHAT_SLA` | Volume, Parallel, AHT, Agents, TTA, Patience | Expected service level |
| `CHAT_ASA` | Volume, Parallel, AHT, Agents, Patience | Avg speed of answer |
| `CHAT_OCCUPANCY` | Volume, Parallel, AHT, Agents, Patience | Agent occupancy |
| `CHAT_ABANDON` | Volume, Parallel, AHT, Agents, Patience | Abandonment fraction |
| `CHAT_AGENTS_SLA` | ServiceLevel, Volume, Parallel, AHT, TTA, Patience | Agents to meet SL target |
| `CHAT_AGENTS_ASA` | ASA, Volume, Parallel, AHT, Patience | Agents to meet ASA target |
| `CHAT_AGENTS_OCCUPANCY` | Occupancy, Volume, Parallel, AHT, Patience | Agents to meet occupancy target |
| `CHAT_AGENTS_ABANDON` | Abandonment, Volume, Parallel, AHT, Patience | Agents to meet abandonment target |

## Error Reference

| Error | Cause |
|---|---|
| `"Invalid parameters"` | One or more inputs are out of range or their combination is infeasible. Check that Volume × AHT < Agents for queue stability. |
| `"Invalid license"` | License expired or not registered. Visit licenses.ccmath.com or contact info@ccmath.com. |

* Optional parameter. When omitted, all parameters to its right must also be omitted. | All time parameters must use the same unit.

---