

tasky_update_task

Description

Updates an existing Tasky task with new properties, status changes, or metadata modifications.

Purpose

Modify task properties including status transitions, content updates, tag changes, and metadata management. Supports partial updates - only specified fields are modified.

Parameters

Parameter	Type	Required	Description
id	string	<input checked="" type="checkbox"/>	Task ID to update
title	string	✗	New task title
description	string	✗	New task description
status	"pending" "in_progress" "completed" "cancelled"	✗	New task status
dueDate	string	✗	New due date in ISO format
tags	string[]	✗	New array of tag strings (replaces existing)
affectedFiles	string[]	✗	New array of file paths
estimatedDuration	number	✗	New estimated duration in minutes
dependencies	string[]	✗	New array of dependency task IDs
reminderEnabled	boolean	✗	Enable/disable reminder
reminderTime	string	✗	New reminder time in HH:MM format
assignedAgent	"claude" "gemini"	✗	New assigned AI agent
executionPath	string	✗	New path for task execution

UI Flow

- 1. **User Input:** "Mark task ABC123 as completed" or "Update task title to 'Fix critical bug'"
- 2. **AI Processing:** Identifies task ID and extracts update parameters

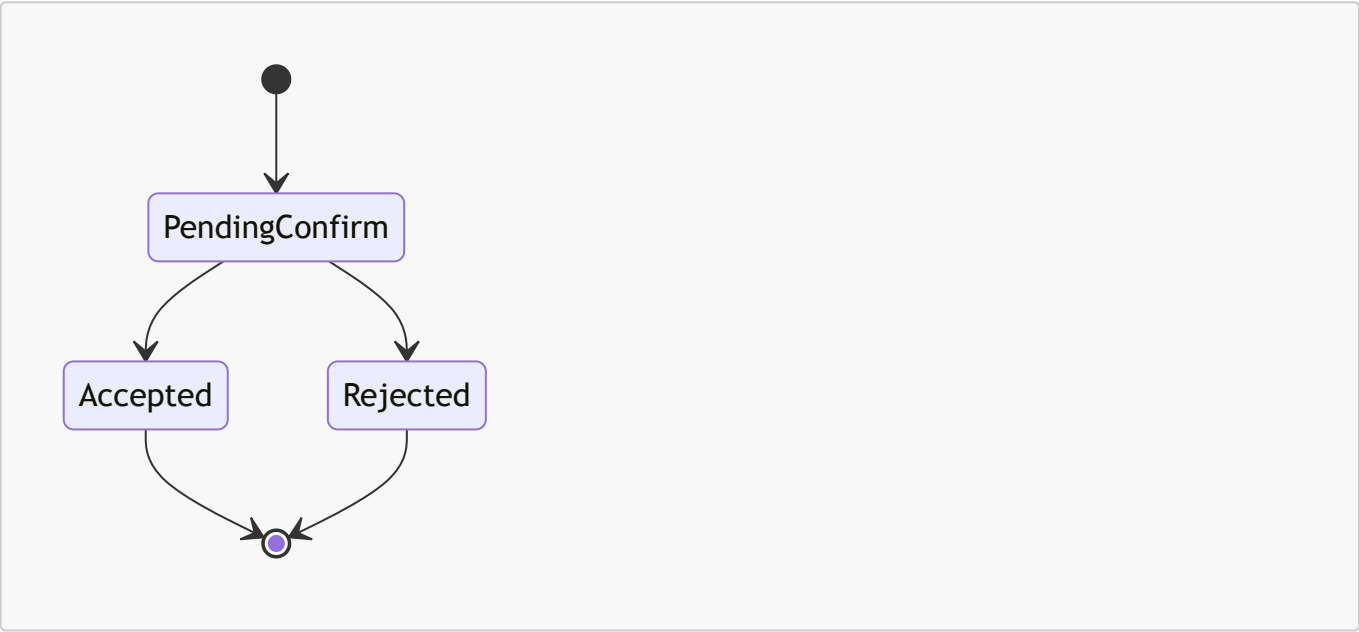
- 3. **Tool Call:** `mcpCall` invoked with task ID and updates
- 4. **Confirmation:** User sees confirmation overlay with changes
- 5. **Validation:** System verifies task exists and updates are valid
- 6. **Execution:** Upon approval, task updated in database
- 7. **Result Display:** Updated task details shown as adaptive card
- 8. **Status Tracking:** Automatic `completedAt` timestamp for completed tasks

Database Operations

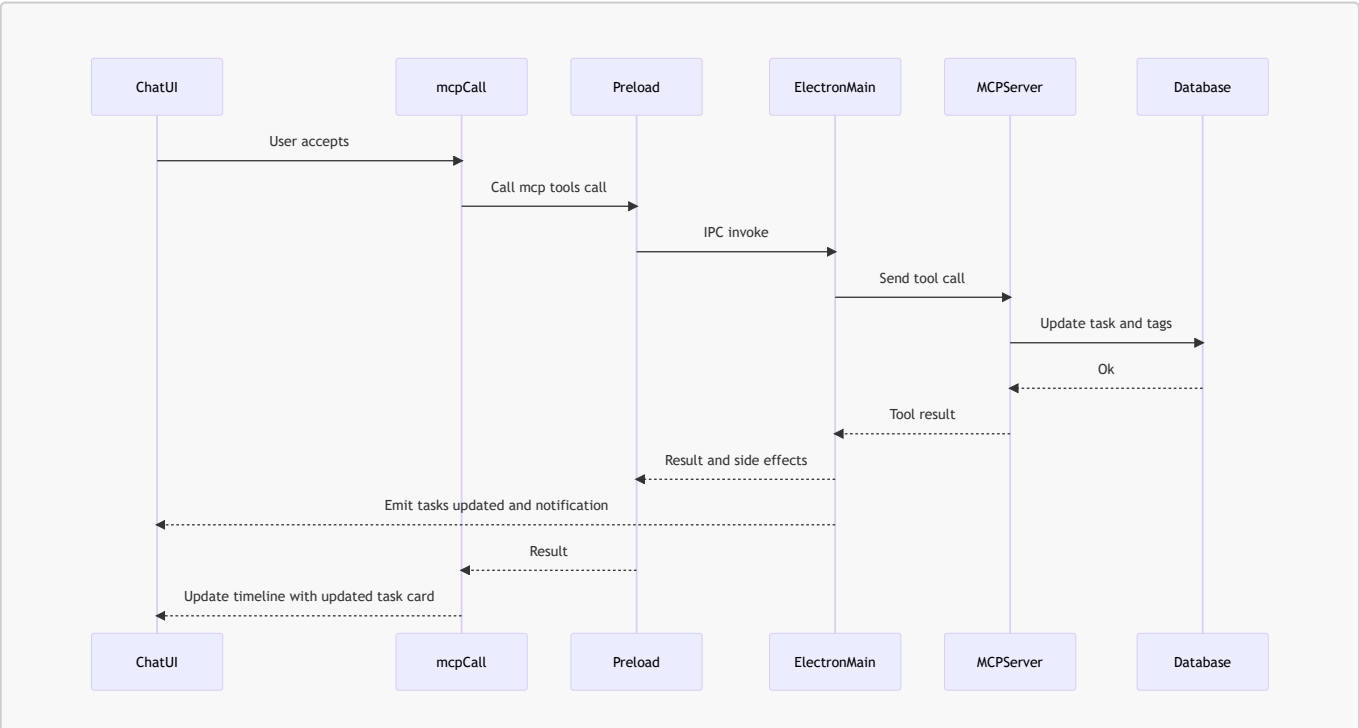
Confirmation Outcomes

This tool requires user confirmation. Auto accept is not used for updates.

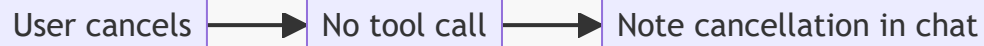
State



Accepted



Rejected



Auto accept

- Not applicable for update

Side effects on accept

- Emits `tasky:tasks-updated` event
- OS notification for updated task
- Adaptive card snapshot embedded in chat

See also: [State Management Diagrams](#)

Adaptive Card Response

Snapshot shape

```
{
  "__taskyCard": {
    "kind": "result",
    "tool": "tasky_update_task",
    "status": "success",
    "data": {
      "schema": {
        "id": "fix_login_bug_20250907_143022_abc123",
        "title": "Fix critical login bug",
        "description": "Updated scope",
        "dueDate": "2025-09-10T17:00:00.000Z",
        "tags": ["bug", "authentication", "urgent"]
      },
      "status": "IN_PROGRESS",
      "reminderEnabled": true,
      "reminderTime": "09:00"
    },
    "meta": { "operation": "update", "timestamp": "2025-09-17T16:00:00.000Z" }
  }
}
```

Error variant

```
{
  "__taskyCard": {
```

```
    "kind": "result",
    "tool": "tasky_update_task",
    "status": "error",
    "error": { "message": "Task not found", "code": "NOT_FOUND" }
  }
}
```

Renderer notes

- Success: Updated task card with status badges and tag changes.
- Error: Inline error with actionable fixes.

```
-- Fetch current task for validation
SELECT * FROM tasks WHERE id = ?;

-- Update task with new values
UPDATE tasks SET
  title = @title,
  description = @description,
  status = @status,
  updated_at = @updated_at,
  due_date = @due_date,
  reminder_enabled = @reminder_enabled,
  result = @result,
  completed_at = @completed_at,
  assigned_agent = @assigned_agent,
  execution_path = @execution_path,
  metadata = @metadata
WHERE id = @id;

-- Update tags (if provided)
DELETE FROM task_tags WHERE task_id = ?;
INSERT INTO task_tags (task_id, tag) VALUES (?, ?);
```

Status Transition Logic

From Status	To Status	Auto-Actions
Any	completed	Sets completedAt timestamp
completed	Any other	Clears completedAt timestamp
Any	in_progress	Updates updatedAt
Any	cancelled	Preserves original timestamps

MCP Request Example

```
curl -X POST http://localhost:7844/mcp \
-H "Content-Type: application/json" \
-d '{
  "jsonrpc": "2.0",
  "id": 3,
  "method": "tools/call",
  "params": {
    "name": "tasky_update_task",
    "arguments": {
      "id": "fix_login_bug_20250907_143022_abc123",
      "status": "completed",
      "description": "Fixed OAuth redirect issue - deployed to production",
      "tags": ["bug", "authentication", "completed"]
    }
  }
}
```

Response Format

```
{
  "jsonrpc": "2.0",
  "id": 3,
  "result": {
    "content": [
      {
        "type": "text",
        "text": "Task fix_login_bug_20250907_143022_abc123: Fix login bug"
      },
      {
        "type": "text",
        "text": "{\n\"schema\":\n{\n  \"id\": \"fix_login_bug_20250907_143022_abc123\", \"title\": \"Fix login\nbug\", \"description\": \"Fixed OAuth redirect issue - deployed to\nproduction\", \"createdAt\": \"2025-09-07T14:30:22.000Z\", \"updatedAt\": \"2025-09-\n07T16:45:00.000Z\", \"dueDate\": \"2025-09-08T17:00:00.000Z\", \"tags\":\n[\"bug\", \"authentication\", \"completed\"], \"assignedAgent\": \"claude\", \"status\":\n\"COMPLETED\", \"reminderEnabled\": false, \"completedAt\": \"2025-09-\n07T16:45:00.000Z\"}\n}"
      }
    ]
  }
}
```

UI Components

- **ConfirmOverlay:** Shows confirmation dialog with:
 - Original task details
 - Proposed changes highlighted

- Impact of status transitions
- **ToolCallDisplay:** Renders update operation status
- **AdaptiveCardRenderer:** Displays updated task with:
 - Status badges with color coding
 - Changed fields highlighted
 - Completion timestamps
 - Tag updates visualized

Common Update Patterns

Status Updates

- **Start Task:** `{"status": "in_progress"}`
- **Complete Task:** `{"status": "completed"}`
- **Cancel Task:** `{"status": "cancelled"}`

Content Updates

- **Rename:** `{"title": "New task title"}`
- **Add Details:** `{"description": "Additional context"}`
- **Update Tags:** `{"tags": ["new", "tag", "list"]}`

Metadata Updates

- **Reassign:** `{"assignedAgent": "gemini"}`
- **Reschedule:** `{"dueDate": "2025-12-31T23:59:59Z"}`
- **Add Reminder:** `{"reminderEnabled": true, "reminderTime": "09:00"}`

Validation Rules

Field	Validation
<code>id</code>	Must exist in database
<code>status</code>	Must be valid enum value
<code>dueDate</code>	Must be valid ISO date string
<code>tags</code>	Array of strings, duplicates removed
<code>assignedAgent</code>	Must be "claude" or "gemini"
<code>reminderTime</code>	Must be HH:MM format

Error Handling

Error	Cause	Response
Task not found	Invalid <code>id</code> parameter	<code>{"content": [{"type": "text", "text": "Task not found"}], "isError": true}</code>

Error	Cause	Response
Missing ID	<code>id</code> parameter not provided	<code>{"content": [{"type": "text", "text": "id is required"}], "isError": true}</code>
Invalid status	Unknown status value	Database constraint error
Date parsing	Malformed date string	Error during date conversion

Transaction Safety

- **Atomic Updates:** Task and tags updated in single transaction
- **Rollback:** Any failure rolls back all changes
- **Consistency:** Maintains referential integrity
- **Timestamps:** Automatic `updatedAt` management

Performance Considerations

- **Single Query:** Fetches current task for validation
- **Partial Updates:** Only modified fields updated
- **Tag Replacement:** Efficient delete-and-insert for tags
- **Indexing:** Benefits from primary key and status indexes

Audit Trail

- **Updated At:** Automatic timestamp on all updates
- **Metadata:** JSON field tracks modification history
- **Status History:** Completion timestamps preserved
- **Version Tracking:** Metadata includes version information

Related Components

- `tasky-mcp-agent/src/mcp-server.ts:111-146` - Tool definition and handler
- `tasky-mcp-agent/src/utils/task-bridge.ts:174-238` - Database update logic
- `src/components/chat/ConfirmOverlay.tsx` - Update confirmation UI
- `src/components/chat/AdaptiveCardRenderer.tsx` - Updated task display

Best Practices

1. **Incremental Updates:** Update only changed fields
2. **Status Transitions:** Follow logical task lifecycle
3. **Tag Management:** Use consistent tag naming
4. **Metadata Preservation:** Maintain creation timestamps
5. **User Confirmation:** Always confirm destructive changes