

Action Reminder Service - Configuration & Email Template Guide

Version: 1.0 Date: 2025-11-14 Target Framework: .NET 10.0

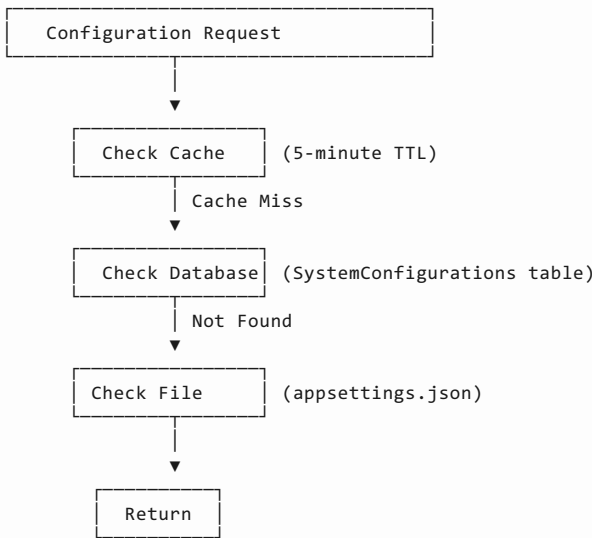
Table of Contents

- 1. [Configuration Architecture](#)
- 2. [Database-Driven Configuration](#)
- 3. [Email Template Customization](#)
- 4. [Template Placeholders Reference](#)
- 5. [Advanced Configuration](#)
- 6. [Testing Configuration](#)

Configuration Architecture

Hybrid Configuration System

The Action Reminder Service uses a **database-first, file-fallback** configuration pattern:



Key Features

- ☐ **Database Override** - Database settings take precedence over file settings
- ☐ **5-Minute Cache** - Performance optimization with automatic expiration
- ☐ **File Fallback** - Always works even if database is unavailable
- ☐ **Hot Reload** - Changes in database take effect within 5 minutes (no service restart)
- ☐ **Automatic Rollback** - Failed SMTP tests rollback all changes
- ☐ **Audit Trail** - All configuration changes are logged

Database-Driven Configuration

Overview

Database configuration allows you to modify settings without editing files or restarting the service. Changes are stored in the `SystemConfigurations` table and take effect within 5 minutes (cache TTL).

Configuration Tables

SystemConfigurations Table

Stores configuration key-value pairs:

Column	Type	Description
ConfigurationId	int	Primary key
ConfigKey	nvarchar(100)	Setting name (e.g., "SmtpHost")
ConfigSection	nvarchar(100)	Section name (e.g., "Email")
ConfigValue	nvarchar(max)	Serialized value
ValueType	nvarchar(50)	Data type (String, Int, Bool, Json, StringArray)
IsActive	bit	Enable/disable setting
IsOverride	bit	TRUE = overrides appsettings.json
CreatedBy	nvarchar(255)	Who created
CreatedDate	datetime2	When created
ModifiedBy	nvarchar(255)	Who modified
ModifiedDate	datetime2	When modified

EmailRecipientGroups Table

Manages recipient email groups:

Column	Type	Description
GroupId	int	Primary key
GroupName	nvarchar(100)	Display name
GroupKey	nvarchar(100)	Lookup key (e.g., "ActionReminderRecipients")
Description	nvarchar(500)	Purpose description
IsActive	bit	Enable/disable group

EmailRecipients Table

Individual email addresses within groups:

Column	Type	Description
RecipientId	int	Primary key
GroupId	int	Foreign key to EmailRecipientGroups
EmailAddress	nvarchar(255)	Email address
DisplayName	nvarchar(255)	Optional display name
IsActive	bit	Enable/disable recipient
SortOrder	int	Display order

EmailTemplates Table

Custom email templates:

Column	Type	Description
TemplateId	int	Primary key
TemplateName	nvarchar(100)	Display name
TemplateKey	nvarchar(100)	Lookup key (e.g., "ActionReminderDaily")
Subject	nvarchar(500)	Email subject line
HtmlBody	nvarchar(max)	HTML email body
PlainTextBody	nvarchar(max)	Plain text fallback
PlaceholderDefinitions	nvarchar(max)	JSON documentation of placeholders
Category	nvarchar(50)	Template category
IsActive	bit	Enable/disable template
IsDefault	bit	Mark as default template

Managing Recipients via Database

Add Recipients to Database:

```
-- Create or update recipient group
IF NOT EXISTS (SELECT 1 FROM EmailRecipientGroups WHERE GroupKey = 'ActionReminderDaily')
BEGIN
    INSERT INTO EmailRecipientGroups (GroupName, GroupKey, Description, IsActive)
    VALUES ('Action Reminder Recipients', 'ActionReminderRecipients', 'Recipients for Action Reminders', 1)
END

-- Get GroupId
DECLARE @GroupId INT = (SELECT GroupId FROM EmailRecipientGroups WHERE GroupKey = 'ActionReminderRecipients')

-- Add recipients (clears existing first)
DELETE FROM EmailRecipients WHERE GroupId = @GroupId

INSERT INTO EmailRecipients (GroupId, EmailAddress, DisplayName, IsActive, SortOrder)
VALUES
    (@GroupId, 'legal@ikea.com', 'Legal Team', 1, 1, 'admin', GETUTCDATE()),
    (@GroupId, 'finance@ikea.com', 'Finance Team', 1, 2, 'admin', GETUTCDATE()),
    (@GroupId, 'admin@ikea.com', 'Admin', 1, 3, 'admin', GETUTCDATE())
```

View Current Recipients:

```
SELECT
    g.GroupName,
    g.GroupKey,
    r.EmailAddress,
    r.DisplayName,
    r.SortOrder,
    r.IsActive
FROM EmailRecipientGroups g
INNER JOIN EmailRecipients r ON g.GroupId = r.GroupId
WHERE g.GroupKey = 'ActionReminderRecipients'
    AND g.IsActive = 1
    AND r.IsActive = 1
ORDER BY r.SortOrder
```

Managing SMTP Configuration via Database

Update SMTP Settings (with automatic test):

```
-- Update SMTP Host
INSERT INTO SystemConfigurations (ConfigKey, ConfigSection, ConfigValue, V
VALUES ('SmtpHost', 'Email', 'smtp.office365.com', 'String', 1, 1, 'admin'

-- Update SMTP Port
INSERT INTO SystemConfigurations (ConfigKey, ConfigSection, ConfigValue, V
VALUES ('SmtpPort', 'Email', '587', 'String', 1, 1, 'admin', GETUTCDATE())

-- Update Username
INSERT INTO SystemConfigurations (ConfigKey, ConfigSection, ConfigValue, V
VALUES ('SmtpUsername', 'Email', 'docuscan@ikea.com', 'String', 1, 1, 'adm

-- Update Password
INSERT INTO SystemConfigurations (ConfigKey, ConfigSection, ConfigValue, V
VALUES ('SmtpPassword', 'Email', 'YourPassword', 'String', 1, 1, 'admin',
```

☐ **Note:** When using the Configuration UI (web application), SMTP settings are tested before being saved. Direct SQL updates bypass this test.

Email Template Customization

Template Architecture

Email templates support: - ☐ **Placeholder Replacement** - `{{Placeholder}}` syntax - ☐ **Loop Blocks** - `{{#ActionRows}}...{{/ActionRows}}` for repeating content - ☐ **HTML Formatting** - Full HTML/CSS support - ☐ **Plain Text Fallback** - For email clients that don't support HTML - ☐ **Test Environment Indicator** - Automatic "[TEST ENVIRONMENT]" banner in debug builds

Default Email Template

The service includes a built-in default template (shown below). This is used if no database template exists.

HTML Template:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8'>
  <style>
    body { font-family: Arial, sans-serif; line-height: 1.6; color: #3
    .container { max-width: 800px; margin: 20px auto; background-color
    .header { background-color: #0051ba; color: #fff; padding: 20px; t
    .header h1 { margin: 0; font-size: 24px; }
    .content { padding: 30px; }
    .reminder-count { background-color: #fff3cd; border-left: 4px soli
    .reminder-table { width: 100%; border-collapse: collapse; margin:
    .reminder-table th { background-color: #0051ba; color: #fff; paddi
    .reminder-table td { padding: 10px; border-bottom: 1px solid #e0e0
    .reminder-table tr:hover { background-color: #f9f9f9; }
    .footer { background-color: #f5f5f5; padding: 15px; text-align: ce
  </style>
</head>
<body>
  <div class='container'>
    <div class='header'>
      <h1>Action Reminders Due Today</h1>
    </div>
    <div class='content'>
      <div class='reminder-count'>
        {{TestEnvironmentIndicator}}You have {{Count}} action remi
      </div>
      <table class='reminder-table'>
        <thead>
          <tr>
            <th>Barcode</th>
            <th>Document Type</th>
            <th>Action Date</th>
            <th>Description</th>
          </tr>
        </thead>
        <tbody>
          {{#ActionRows}}
          <tr>
            <td><strong><a href='https://localhost:44101/docum
            <td>{{DocumentType}}</td>
            <td>{{ActionDate}}</td>
            <td>{{ActionDescription}}</td>
          </tr>
          {{/ActionRows}}
        </tbody>
      </table>
      <p>Please review these documents and take the necessary action
    </div>
    <div class='footer'>
      <p>This is an automated reminder from the DocuScan System.</p>
    </div>
  </div>
</body>
</html>

```

Creating Custom Template in Database

SQL Script to Create Custom Template:

```

-- Create or update ActionReminderDaily template
DECLARE @TemplateKey NVARCHAR(100) = 'ActionReminderDaily'

-- Check if template exists
IF EXISTS (SELECT 1 FROM EmailTemplates WHERE TemplateKey = @TemplateKey)
BEGIN
  -- Update existing template
  UPDATE EmailTemplates

```

```

SET
    TemplateName = 'Daily Action Reminder',
    Subject = 'Action Reminders Due - {{Count}} Items',
    HtmlBody = N'<!DOCTYPE html>

<html>
<head>
    <meta charset='utf-8'>
    <style>
        body { font-family: Arial, sans-serif; line-height: 1.6; color: #3
        .container { max-width: 800px; margin: 20px auto; background-color
        .header { background-color: #0051ba; color: #fff; padding: 20px; t
        .content { padding: 30px; }
        .reminder-table { width: 100%; border-collapse: collapse; margin:
        .reminder-table th { background-color: #0051ba; color: #fff; paddi
        .reminder-table td { padding: 10px; border-bottom: 1px solid #e0e0
    </style>
</head>
<body>
    <div class='container'>
        <div class='header'>
            <h1>Action Reminders Due Today</h1>
        </div>
        <div class='content'>
            <p>{{TestEnvironmentIndicator}}You have <strong>{{Count}}</str
            <table class='reminder-table'>
                <thead>
                    <tr>
                        <th>Barcode</th>
                        <th>Document Type</th>
                        <th>Action Date</th>
                        <th>Description</th>
                    </tr>
                </thead>
                <tbody>
                    <{{#ActionRows}}
                    <tr>
                        <td><strong><a href='https://your-server/document
                        <td>{{DocumentType}}</td>
                        <td>{{ActionDate}}</td>
                        <td>{{ActionDescription}}</td>
                    </tr>
                    <{{/ActionRows}}
                </tbody>
            </table>
        </div>
    </body>
</html>',
    PlainTextBody = N'Action Reminders Due Today

You have {{Count}} action reminder(s) due on {{Date}}.

{{#ActionRows}}
- Barcode: {{BarCode}}
  Document Type: {{DocumentType}}
  Action Date: {{ActionDate}}
  Description: {{ActionDescription}}

{{/ActionRows}}

This is an automated reminder from DocuScan System.',
    PlaceholderDefinitions = N'{"Global":["Count","Date","TestEnvironm
    Category = 'ActionReminder',
    IsActive = 1,
    IsDefault = 1,
    ModifiedBy = 'admin',
    ModifiedDate = GETUTCDATE()
    WHERE TemplateKey = @TemplateKey
END

```

```
ELSE
BEGIN
    -- Insert new template
    INSERT INTO EmailTemplates (
        TemplateName, TemplateKey, Subject, HtmlBody, PlainTextBody,
        PlaceholderDefinitions, Category, IsActive, IsDefault, CreatedBy,
    )
    VALUES (
        'Daily Action Reminder',
        @TemplateKey,
        'Action Reminders Due - {{Count}} Items',
        N'<!DOCTYPE html>... (HTML from above) ...',
        N'Action Reminders Due Today... (Plain text from above) ...',
        N'{"Global":["Count","Date","TestEnvironmentIndicator"],"ActionRow
        'ActionReminder',
        1,
        1,
        'admin',
        GETUTCDATE()
    )
END
```

Template Placeholders Reference

Global Placeholders

These placeholders are replaced in the template body and subject:

Placeholder	Type	Description	Example Value
{{Count}}	int	Number of action reminders due	5
{{Date}}	DateTime	Current date	2025-11-14 08:00
{{TestEnvironmentIndicator}}	string	Shows “[TEST ENVIRONMENT]” in debug builds, empty in production	[TEST ENVIRONME or “

Usage Examples:

```
<!-- In email body -->
<p>You have {{Count}} action reminder(s) due on {{Date}}.</p>

<!-- In email subject -->
<Subject>Action Reminders Due - {{Count}} Items</Subject>

<!-- Test indicator -->
<div>{{TestEnvironmentIndicator}}This is a reminder notification.</div>
```

Loop Block: ActionRows

The {{#ActionRows}}...{/ActionRows} block repeats for each action reminder. All row placeholders are available inside this block.

Syntax:

```
{{#ActionRows}}
<!-- This content repeats for each reminder -->
<tr>
    <td>{{BarCode}}</td>
    <td>{{DocumentType}}</td>
</tr>
{{/ActionRows}}
```

Row Placeholders (Inside ActionRows Loop)

Placeholder	Type	Description	Example Value
{{BarCode}}	string	Document barcode	"2025001234"
{{DocumentType}}	string	Document type name	"Invoice"
{{DocumentName}}	string	Document name	"Supplier Agreement"
{{DocumentNo}}	string	Document number	"INV-2025-001"
{{CounterParty}}	string	Counter party name	"ACME Corporation"
{{CounterPartyNo}}	string	Counter party ID	"123"
{{ActionDate}}	DateTime?	When action is due	2025-11-14
{{ReceivingDate}}	DateTime?	When document was received	2025-11-01
{{ActionDescription}}	string	Description of action needed	"Review and approve"
{{Comment}}	string	Additional comments	"Urgent - deadline approaching"
{{IsOverdue}}	bool	TRUE if action date < today	true or false

Complete Example with All Row Placeholders:

```
{{#ActionRows}}
<div class='reminder-item'>
    <h3>Document: {{BarCode}}</h3>
    <table>
        <tr><td>Document Type:</td><td>{{DocumentType}}</td></tr>
        <tr><td>Document Name:</td><td>{{DocumentName}}</td></tr>
        <tr><td>Document No:</td><td>{{DocumentNo}}</td></tr>
        <tr><td>Counter Party:</td><td>{{CounterParty}} (ID: {{CounterPart
        <tr><td>Action Date:</td><td>{{ActionDate}}</td></tr>
        <tr><td>Receiving Date:</td><td>{{ReceivingDate}}</td></tr>
        <tr><td>Action Description:</td><td>{{ActionDescription}}</td></tr>
        <tr><td>Comments:</td><td>{{Comment}}</td></tr>
        <tr><td>Status:</td><td>{{IsOverdue}}</td></tr>
    </table>
</div>
{{/ActionRows}}
```

Placeholder Formatting

DateTime Placeholders:

DateTime values are formatted using default .NET formatting. To customize:

```
<!-- Default formatting -->
{{ActionDate}} <!-- Output: 11/14/2025 12:00:00 AM -->

<!-- Note: Custom formatting not directly supported in templates -->
<!-- Use CSS or JavaScript for display formatting -->
```

Boolean Placeholders:

Boolean values render as true or false:

```
{{IsOverdue}} <!-- Output: true or false -->

<!-- To show different text, use CSS classes -->
<span class='status-{{IsOverdue}}'>{{IsOverdue}}</span>
```

Conditional Display (Using CSS)

While the template engine doesn't support IF statements, you can use CSS to conditionally display content:

```
<style>
.overdue-true { color: red; font-weight: bold; }
.overdue-false { color: green; }
</style>

{{#ActionRows}}
<tr class='overdue-{{IsOverdue}}'>
  <td>{{BarCode}}</td>
  <td>{{ActionDescription}}</td>
</tr>
{{/ActionRows}}
```

Advanced Configuration

Using Web Application Configuration UI

The IkeaDocuScan web application includes a Configuration Management page (SuperUser only) for managing:

1. **System Configuration** - All settings in SystemConfigurations table
2. **Email Recipients** - Manage recipient groups
3. **Email Templates** - Visual template editor with live preview
4. **SMTP Configuration** - With automatic connection testing

Access: - Navigate to: /configuration (or Admin menu → Configuration) - Requires: SuperUser role

Features: - ☐ Visual template editor with syntax highlighting - ☐ Live preview with sample data - ☐ Placeholder validation - ☐ SMTP connection testing before saving - ☐ Automatic rollback on errors - ☐ Audit trail for all changes

Configuration Priority

When the same setting exists in multiple locations:

Priority Order (Highest to Lowest): 1. **Database** (SystemConfigurations table with IsOverride=true) 2. **Environment Variables** (e.g., ActionReminderService__ScheduleTime) 3. **appsettings.Local.json** (server-specific, not committed to source control) 4. **appsettings.{Environment}.json** (e.g., appsettings.Production.json) 5. **appsettings.json** (default configuration file)

Example:

If ScheduleTime is set to: - Database: "07:00" - appsettings.json: "08:00"

Result: Service runs at **07:00** (database wins)

Cache Management

Configuration is cached for 5 minutes. To force immediate cache refresh:

Option 1: Wait 5 Minutes Changes will automatically take effect after cache expires.

Option 2: Restart Service

```
Restart-Service -Name "IkeaDocuScanActionReminder"
```

Option 3: Clear Cache via Web UI Navigate to Configuration page → Click “Reload Configuration”

Disabling Database Configuration

To always use file-based configuration (ignoring database):

Set IsOverride=false for all configurations:

```
UPDATE SystemConfigurations
SET IsOverride = 0
WHERE IsActive = 1
```

Or disable specific configuration:

```
UPDATE SystemConfigurations
SET IsOverride = 0
WHERE ConfigKey = 'RecipientEmails'
AND ConfigSection = 'ActionReminderService'
```

Testing Configuration

Test 1: Verify Configuration Loading

Check what configuration service will use:

```
-- Check email recipients (database)
SELECT
    g.GroupKey,
    r.EmailAddress,
    r.IsActive
FROM EmailRecipientGroups g
INNER JOIN EmailRecipients r ON g.GroupId = r.GroupId
WHERE g.GroupKey = 'ActionReminderRecipients'
AND g.IsActive = 1
AND r.IsActive = 1
ORDER BY r.SortOrder

-- Check SMTP configuration (database)
SELECT
    ConfigKey,
    ConfigValue,
    IsActive,
    IsOverride,
    ModifiedBy,
    ModifiedDate
FROM SystemConfigurations
WHERE ConfigSection = 'Email'
AND IsActive = 1
ORDER BY ConfigKey
```

Test 2: Test SMTP Connection

From SQL Server:

```
-- Insert test configuration
INSERT INTO SystemConfigurations (ConfigKey, ConfigSection, ConfigValue, V
VALUES ('TestSmtpConnection', 'Email', 'true', 'Bool', 1, 1, 'admin', GETU
```

Then check service logs in Event Viewer for “SMTP connection test” messages.

From Web UI: 1. Navigate to /configuration 2. Go to SMTP Configuration section 3. Click “Test Connection” 4. Result displayed immediately

Test 3: Test Email Template Rendering

Create test data:

```
-- Temporarily set DaysAhead to include some documents
UPDATE SystemConfigurations
SET ConfigValue = '7' -- Look ahead 7 days
WHERE ConfigKey = 'DaysAhead'
    AND ConfigSection = 'ActionReminderService'

-- Or add a test document with ActionDate = today
INSERT INTO Documents (Barcode, DocumentNo, ActionDate, ReceivingDate, Act
VALUES (9999999, 'TEST-001', CAST(GETDATE() AS DATE), CAST(GETDATE() AS DA
```

Trigger service immediately:

Change schedule time to 1 minute from now, restart service, and monitor Event Viewer.

Test 4: Verify Template Placeholders

SQL query to test placeholders:

```
-- Check what data would be sent in email
SELECT
    Barcode as '{{Barcode}}',
    ISNULL(dt.DtName, '') as '{{DocumentType}}',
    ISNULL(dn.Name, '') as '{{DocumentName}}',
    DocumentNo as '{{DocumentNo}}',
    ISNULL(cp.Name, '') as '{{CounterParty}}',
    CAST(cp.CounterPartyId AS NVARCHAR) as '{{CounterPartyNo}}',
    ActionDate as '{{ActionDate}}',
    ReceivingDate as '{{ReceivingDate}}',
    ActionDescription as '{{ActionDescription}}',
    Comment as '{{Comment}}',
    CASE WHEN ActionDate < CAST(GETDATE() AS DATE) THEN 'true' ELSE 'false'
FROM Documents d
LEFT JOIN DocumentTypes dt ON d.DocumentTypeId = dt.DocumentTypeId
LEFT JOIN DocumentNames dn ON d.DocumentNameId = dn.DocumentNameId
LEFT JOIN CounterParties cp ON d.CounterPartyId = cp.CounterPartyId
WHERE ActionDate = CAST(GETDATE() AS DATE)
    AND ActionDate >= ReceivingDate
ORDER BY ActionDate, Barcode
```

Common Template Customization Scenarios

Scenario 1: Add Company Logo

```
<div class='header'>
    <img src='https://your-server/images/logo.png' alt='Company Logo' styl
    <h1>Action Reminders Due Today</h1>
</div>
```

Scenario 2: Highlight Overdue Items

```

<style>
.overdue { background-color: #ffebee; }
.on-time { background-color: #e8f5e9; }
</style>

{{#ActionRows}}
<tr class='{{#if IsOverdue}}overdue{{else}}on-time{{/if}}'>
    <td>{{BarCode}}</td>
    <td>{{ActionDate}}</td>
</tr>
{{/ActionRows}}

```

Note: Since `{{#if}}` is not supported, use CSS classes:

```

<tr class='overdue-{{IsOverdue}}'>

```

And define CSS for both states:

```

.overdue-true { background-color: #ffebee; }
.overdue-false { background-color: #e8f5e9; }

```

Scenario 3: Group by Document Type

While grouping isn't directly supported in templates, you can sort the database query:

(This requires code change in `ActionReminderEmailService.cs`)

```

.OrderBy(d => d.DocumentType) // Group by document type
.ThenBy(d => d.ActionDate)
.ThenBy(d => d.BarCode)

```

Then use section headers in template:

```

{{#ActionRows}}
<tr>
    <td colspan='4'><strong>{{DocumentType}}</strong></td>
</tr>
<tr>
    <td>{{BarCode}}</td>
    <td>{{ActionDate}}</td>
    <td>{{ActionDescription}}</td>
</tr>
{{/ActionRows}}

```

Scenario 4: Add Direct Action Links

```

{{#ActionRows}}
<tr>
    <td><a href='https://your-server/documents/edit/{{BarCode}}'>{{BarCode}}
    <td>{{DocumentType}}</td>
    <td>
        <a href='https://your-server/documents/edit/{{BarCode}}'>View</a>
        <a href='https://your-server/documents/edit/{{BarCode}}?action=app
    </td>
</tr>
{{/ActionRows}}

```

Scenario 5: Mobile-Responsive Template

```

<style>
@media only screen and (max-width: 600px) {
    .reminder-table { font-size: 12px; }
    .reminder-table th, .reminder-table td { padding: 5px; }
}
</style>

```

Summary

□ **Database-First Configuration** - Override any setting via database without file edits □ **5-Minute Cache** - Changes take effect quickly without service restart □ **Custom Email Templates** - Full HTML/CSS support with placeholder system □ **Loop Support** - Repeat content for each action reminder □ **Test Environment Indicator** - Automatic banner in test environments □ **Automatic Rollback** - Failed SMTP tests rollback all changes □ **Audit Trail** - All changes logged with who/when/why

For Windows Service installation and management, see:
ACTION_REMINDER_SERVICE_SETUP_GUIDE.md

Quick Reference

Add Email Recipients (SQL):

```
DECLARE @GroupId INT = (SELECT GroupId FROM EmailRecipientGroups WHERE Gro
INSERT INTO EmailRecipients (GroupId, EmailAddress, IsActive, SortOrder, C
VALUES (@GroupId, 'new@email.com', 1, 10, 'admin', GETUTCDATE()))
```

Update SMTP Host (SQL):

```
INSERT INTO SystemConfigurations (ConfigKey, ConfigSection, ConfigValue, V
VALUES ('SmtpHost', 'Email', 'smtp.newserver.com', 'String', 1, 1, 'admin'
```

View Current Template:

```
SELECT HtmlBody FROM EmailTemplates WHERE TemplateKey = 'ActionReminderDai
```

Force Cache Refresh:

```
Restart-Service -Name "IkeaDocuScanActionReminder"
```

For questions or issues with templates, check Event Viewer logs for template rendering errors.