

# DBMS\_CLOUD\_NOTIFICATION

The `DBMS_CLOUD_NOTIFICATION` package allows you to send messages or the output of a SQL query to a provider.

## DBMS\_CLOUD\_NOTIFICATION Overview

This scenario describes setup and use of the `DBMS_CLOUD_NOTIFICATION` package.

The supported providers are: Email, Microsoft Teams, and Slack.

## Summary of DBMS\_CLOUD\_NOTIFICATION Subprograms

This table summarizes the subprograms included in the package.

Subprogram	Description
<a href="#">SEND_DATA Procedure</a>	Send SQL query output to a provider.
<a href="#">SEND_MESSAGE Procedure</a>	Send a text message to a provider.

## SEND\_DATA Procedure

The `SEND_DATA` procedure sends the results of the specified query to a provider.

### Syntax

```
DBMS_CLOUD_NOTIFICATION.SEND_DATA(  
    provider          IN  VARCHAR2,  
    credential_name   IN  VARCHAR2,  
    query             IN  CLOB,  
    params            IN  CLOB  
);
```

### Parameters

Parameter	Description
provider	Specifies the provider. Valid values are: 'email', 'msteams', and 'slack' This parameter is mandatory.

Parameter	Description
<code>credential_name</code>	<p>The name of the credential object to access the provider.</p> <p>For the email provider, the credential is the name of the credential of the SMTP approved sender that contains its username and password.</p> <p>For the msteams provider, the credential is the name of the credential.</p> <p>For the Slack provider, the credential's username must be "SLACK_TOKEN" and the password is the Slack Bot Token.</p> <p>This parameter is mandatory.</p>
<code>query</code>	<p>Specifies the SQL query to run. Results are sent to the provider.</p> <p>This parameter is mandatory.</p>
<code>params</code>	<p>Specifies specific parameters for the <code>provider</code> in JSON format.</p> <p>For the provider type <code>email</code>, the valid parameters are:</p> <ul style="list-style-type: none"> <li>• <code>sender</code>. This specifies the Email ID of the approved sender in a <code>String</code> value.</li> <li>• <code>smtp_host</code>. This specifies the SMTP host name in a <code>String</code> value.</li> <li>• <code>subject</code>. This specifies the subject of the email in a <code>String</code> value. The maximum size is 100 characters.</li> <li>• <code>recipient</code>. This specifies the email IDs of recipients in a <code>String</code> value. Use a comma between email IDs when there are multiple recipients.</li> <li>• <code>to_cc</code>. This specifies the email IDs that are receiving a CC of the email. It is a <code>String</code> value. Use a comma between email IDs when there are multiple CC recipients.</li> <li>• <code>to_bcc</code>. This specifies the email IDs that are receiving a BCC of the email. It is a <code>String</code> value. Use a comma between email IDs when there are multiple BCC recipients.</li> <li>• <code>message</code>. This specifies the message text in a <code>String</code> value.</li> <li>• <code>type</code>. This specifies the output format in a <code>String</code> value as either CSV or JSON.</li> <li>• <code>title</code>. This specifies the title of the attachment of SQL output in a <code>String</code> value. The title should only contain letters, digits, underscores, hyphens, or dots as characters in its value due to it being used to generate a file name.</li> </ul> <p>For the provider type <code>msteams</code>, the valid parameters are:</p> <ul style="list-style-type: none"> <li>• <code>tenant</code>. This specifies the tenant ID in a <code>String</code> format.</li> <li>• <code>team</code>. This specifies the team ID in a <code>String</code> format.</li> <li>• <code>channel</code>. This specifies the channel ID in a <code>String</code> format.</li> <li>• <code>title</code>. This specifies the title of the file in a <code>String</code> format. The maximum size is 50 characters.</li> <li>• <code>type</code>. This specifies the output format in a <code>String</code> value. Valid values are CSV or JSON.</li> </ul> <p>For the provider type <code>slack</code>, the valid parameters are:</p> <ul style="list-style-type: none"> <li>• <code>channel</code>. This specifies the Channel ID in a <code>String</code> value. The Channel ID is a unique ID for a channel and is different from the channel name. In Slack, when you view channel details you can find the Channel ID on the "About" tab.</li> <li>• <code>type</code>. This specifies the output format in a <code>String</code> value. Valid values are CSV or JSON.</li> </ul> <p>This parameter is mandatory.</p>

## Usage Notes

- Use the procedure `DBMS_CLOUD.CREATE_CREDENTIAL(credential_name, username, password)` to create the credential object.  
See [CREATE\\_CREDENTIAL Procedure](#) for more information.
- For the `email` provider, note the following:
  - The user requires an SMTP connection endpoint for the Email Delivery server to obtain `smtp_host`. The user also requires an SMTP approved sender and its credentials to authenticate with the Email Delivery server to obtain the `credential_name`. The SMTP connection must be configured and the SMTP credentials must be generated and approved.
  - The maximum message size supported when using `DBMS_CLOUD_NOTIFICATION.SEND_DATA` with the `email` provider is 32k bytes.
  - You can only use `DBMS_CLOUD_NOTIFICATION` for mail notifications with Autonomous Database version 19.21 and above.
- For the `msteams` provider, the user requires the Microsoft Teams App and a bot configured in it. The app needs to be published to the org and installed after obtaining approval from the admin from the admin center. The user also requires access for `Files.ReadWrite.All` and `ChannelSettings.Read.All` permission for Graph API from their Azure Portal. To generate the required token, the user requires `bot_id` in the username and `bot_secret` in the password.

The maximum file size supported when using `DBMS_CLOUD_NOTIFICATION.SEND_DATA` for Microsoft Teams is 4MB.

- For the `slack` provider, the `username` value can be any valid string and the `password` is the Slack bot token.

## Examples

Send SQL output with the `email` provider:

```
BEGIN
    DBMS_CLOUD.CREATE_CREDENTIAL(
        credential_name => 'EMAIL_CRED',
        username        => 'username',
        password        => 'password');
END;
/

BEGIN
    DBMS_CLOUD_NOTIFICATION.SEND_DATA(
        provider          => 'email',
        credential_name   => 'EMAIL_CRED',
        query             => 'SELECT tablespace_name FROM dba_tablespace',
        params            => json_object('recipient' value 'mark@oracle.com',
suresh@oracle.com',
                                                'to_cc' value 'nicole@oracle.com1',
jordan@oracle.com',
                                                'to_bcc' value 'manisha@oracle.com',
                                                'subject' value 'Test subject',
                                                'type' value 'json',
                                                'title' value 'mytitle',
```

```

                                'message' value 'This is the message',
                                'smtp_host' value
'smtp.email.example.com',
                                'sender'      value
'approver_sender@oracle.com' )
    );
END;
/

```

Send SQL output with the `msteams` provider:

```

BEGIN
    DBMS_CLOUD.CREATE_CREDENTIAL(credential_name => 'TEAMS_CRED',
                                username         => 'bot_id',
                                password        => 'bot_secret');
END;
/

BEGIN
    DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE(provider => 'msteams',
    credential_name => 'TEAMS_CRED',
    query          => 'SELECT tablespace_name FROM dba_tablespaces',
    params         => json_object('tenant'value '5b743bc*****c0286',
                                'team'value '0ae401*****5d2bd',
                                'channel'value
'19%3a94be023*****%40thread.tacv2',
                                'title'value 'today',
                                'type'value 'csv'));
END;
/

```

Send SQL output with the `slack` provider:

```

BEGIN
    DBMS_CLOUD.CREATE_CREDENTIAL(
        credential_name => 'SLACK_CRED',
        username        => 'SLACK_TOKEN',
        password        => 'password');
END;
/

BEGIN
    DBMS_CLOUD_NOTIFICATION.SEND_DATA(
        provider        => 'slack',
        credential_name => 'SLACK_CRED',
        query          => 'SELECT username, account_status, expiry_date FROM
USER_USERS WHERE rownum < 5',
        params         => json_object('channel' value 'C0....08','type' value
'csv'));
END;
/

```

# SEND\_MESSAGE Procedure

The procedure sends a text message to a provider.

## Syntax

```
DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE (  
    provider          IN  VARCHAR2,  
    credential_name   IN  VARCHAR2,  
    message           IN  CLOB,  
    params            IN  CLOB  
);
```

## Parameters

Parameter	Description
provider	Specifies the provider. Valid values are: 'email', 'msteams', and 'slack' This parameter is mandatory.
credential_name	The name of the credential object to access the provider. For the email provider, the credential is the name of the credential of the SMTP approved sender that contains its username and password. For the msteams provider, the credential must contain the bot_id and the bot_secret key in both the username and the password. For the Slack provider, the credential's username must be "SLACK_TOKEN" and the password is the Slack Bot Token. This parameter is mandatory.
message	Specifies the message text. This parameter is mandatory.

Parameter	Description
<code>params</code>	<p>Specifies specific parameters for the <code>provider</code> in JSON format.</p> <p>For the provider type <code>email</code>, the valid parameters are:</p> <ul style="list-style-type: none"> <li><code>sender</code>. This specifies the Email ID of the approved sender in a <code>String</code> value.</li> <li><code>smtp_host</code>. This specifies the SMTP host name in a <code>String</code> value.</li> <li><code>subject</code>. This specifies the subject of the email in a <code>String</code> value. The maximum size is 100 characters.</li> <li><code>recipient</code>. This specifies the email IDs of recipients in a <code>String</code> value. Use a comma between email IDs when there are multiple recipients.</li> <li><code>to_cc</code>. This specifies the email IDs that are receiving a CC of the email. Use a comma between email IDs when there are multiple CC recipients.</li> <li><code>to_bcc</code>. This specifies the email IDs that are receiving a BCC of the email. Use a comma between email IDs when there are multiple BCC recipients.</li> </ul> <p>For the provider type <code>msteams</code>, the valid parameter is:</p> <ul style="list-style-type: none"> <li><code>channel</code>. This specifies the Channel ID in a <code>String</code> value.</li> </ul> <p>For the provider type <code>slack</code>, the valid parameter is:</p> <ul style="list-style-type: none"> <li><code>channel</code>. This specifies the Channel ID in a <code>String</code> value.</li> </ul> <p>The Channel ID is a unique ID for a channel and is different from the channel name. In Slack, when you view channel details you can find the Channel ID on the “About” tab.</p> <p>This parameter is mandatory.</p>

### Usage Notes

Use the procedure `DBMS_CLOUD.CREATE_CREDENTIAL(credential_name, username, password)` to create the credential object. See [CREATE\\_CREDENTIAL Procedure](#) for more information.

- For the `email` provider, the user requires an SMTP connection endpoint for the Email Delivery server to obtain `smtp_host`. The user also requires an SMTP approved sender and its credentials to authenticate with the Email Delivery server to obtain the `credential_name`. SMTP connection must be configured, and SMTP credentials must be generated and approved.
- For the `msteams` provider, the user requires the Microsoft Teams App and a bot configured in it. The app needs to be published to the org and installed after obtaining approval from the admin from the admin center. The user also requires access for `Files.ReadWrite.All` and `ChannelSettings.Read.All` permission for Graph API from their Azure Portal. To generate the required token, the user requires `bot_id` in the username and `bot_secret` in the password.
- For the `slack` provider, the `username` value can be any valid string and the `password` is the Slack bot token.

### Examples

Send a text message with the `email` provider:

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'EMAIL_CRED',
    username        => 'username',
    password        => 'password');
```

```

END;
/

BEGIN
  DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE(
    provider      => 'email',
    credential_name => 'EMAIL_CRED',
    message       => 'Subject content',
    params        => json_object('recipient' value 'mark@oracle.com,
suresh@oracle.com',
                                'to_cc' value 'nicole@oracle.com,
jordan@oracle.com',
                                'to_bcc' value 'manisha@oracle.com',
                                'subject' value 'Test subject',
                                'smtp_host' value
'smtp.email.example.com',
                                'sender' value
'approver_sender@oracle.com' )
  );
END;
/

```

Send a text message with the `msteams` provider:

```

BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(credential_name => 'TEAMS_CRED',
    username      => 'bot_id',
    password      => 'bot_secret');
END;
/

BEGIN
  DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE(
    provider      => 'msteams',
    credential_name => 'TEAMS_CRED',
    message       => 'text from new teams api',
    params        => json_object('channel' value
'C0....08'),'region' value 'india');
END;
/

```

Send a text message with the `slack` provider:

```

BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'SLACK_CRED',
    username      => 'SLACK_TOKEN',
    password      => 'password');
END;
/

BEGIN
  DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE(
    provider      => 'slack',
    credential_name => 'SLACK_CRED',

```

```
        message      => 'Send text from Autonomous Database.',  
        params       => json_object('channel' value 'C0....08');  
    END;  
/
```