

Application Settings

Last updated by | Maksim Murashka E² | 5 Dec 2025 at 2:51 pm AEDT

Contents

- [Application Settings](#)
 - [Overview](#)
 - [How to configure Application Settings](#)
 - [Application settings](#)
 - [App Settings JSON parameters](#)
 - [Required settings](#)
 - [AppName](#)
 - [AuthorizedGroupId](#)
 - [AuthorizedGroupName](#)
 - [EmailNotificationsServiceAccount](#)
 - [TopHeaderBannerText](#)
 - [Optional settings](#)
 - [SharePoint Uploads](#)
 - [Admin Workflow Override](#)
 - [Documents Upload Drag and Drop](#)
 - [User Extra Metadata](#)
 - [Plugins Editor](#)
 - [EnablePluginEditor feature flag](#)
 - [BundleStore Kind](#)
 - [BundleStore ConnectionString](#)
 - [BundleStore ContainerName](#)
 - [BundleStore FileShareName](#)
 - [BundleStore Folder](#)
 - [Open Records in New Tab](#)
 - [Show Validation Errors](#)
 - [Email Subject Postfix for Proxy](#)
 - [Email Notifications](#)
 - [Sender Type](#)
 - [SMTP Username](#)
 - [SMTP Password](#)
 - [SMTP Server](#)
 - [SMTP Port](#)
 - [SMTP Use Secure Connection](#)
 - [SMTP Timeout](#)
 - [SMTP From Address](#)
 - [SMTP Reply-To Address](#)
 - [Allowed Upload File Extensions](#)
 - [Troubleshooting](#)
 - [Feature Flag Not Taking Effect](#)
 - [SharePoint Upload Options Not Visible](#)
 - [General Configuration Issues](#)

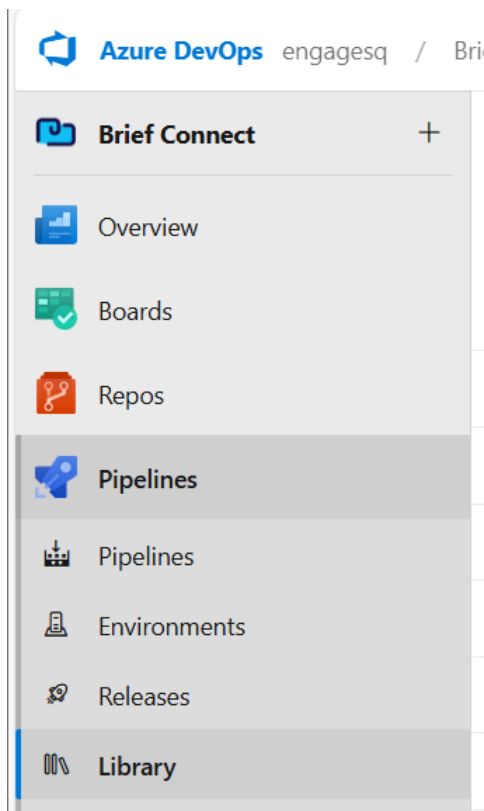
Application Settings

Overview

This page covers application configuration options for Brief Connect, including feature flags and other settings. They are used to configure system behavior. Should be managed via application deployment process. Application settings updates are pushed with every core application release (application package release).

How to configure Application Settings

1. Open Azure DevOps
2. Open Brief Connect project
3. In the left navigation menu click on "Library"



4. Open the Variable group with Environment/Stage name and add/update values

Application settings

The list below represents only required pipeline variables for successful application deployment. You can add additional environment variables if you need so.

Name	Is Required
appSettingsJson	true
applicationDomainName	true
azureFunctionAppName	true
azureFunctionDomainName	true
azureResourceGroupName	true
azureStorageAccountWebName	true
azureWebApiWebAppName	true
clientAppRegistrationClientId	true
devOpsAppRegistrationClientId	true
serverAppRegistrationClientId	true
sharePointSiteUrl	true
tenantId	true

Description:

- `appSettingsJson` : JSON object containing application settings. See the section below for available settings included in this variable.
- `applicationDomainName` : Public domain name for the Brief Connect application (UI). Example: `briefconnect.contoso.com`
- `azureFunctionAppName` : Brief Connect Backend API Azure Function App name (lowercase, no special chars). Example: `azureResourcesNamePrefix-PFA`
- `azureFunctionDomainName` : Domain name mapped to Brief Connect Backend API Azure Functions endpoints. Example: `briefconnect.contoso.com`
- `azureResourceGroupName` : Azure Resource Group name that hosts Brief Connect resources. Example: `rg-briefconnect-prod`
- `azureStorageAccountWebName` : Azure Storage account name (lowercase, no special chars) for UI web assets (bundles/static content). Example: `azureResourcesNamePrefixwsa`
- `azureWebApiWebAppName` : Brief Connect Integration API Azure Web App name (lowercase, no special chars). Example: `azureResourcesNamePrefix-API`

- `clientAppRegistrationClientId`: 'Client' App registration Application (client) ID for the client/UI app. Example: `10000000-0000-0000-0000-000000000001`
- `devOpsAppRegistrationClientId`: 'DevOps' App registration Application (client) ID used by CI/CD and DevOps automation. Example: `20000000-0000-0000-0000-000000000002`
- `serverAppRegistrationClientId`: 'Server' App registration Application (client) ID for server-side/API components. Example: `30000000-0000-0000-0000-000000000003`
- `sharePointSiteUrl`: SharePoint site URL used for storing Brief Connect documents. Example: `https://contoso.sharepoint.com/sites/BriefConnect`
- `tenantId`: Entra ID (Azure AD) tenant GUID. Example: `11111111-2222-3333-4444-555555555555`

App Settings JSON parameters

This variable contains a JSON object with multiple application settings. Example: `{"Key1": "Value1", "Key2": "Value2"}` **Must be a valid JSON format.** Please validate final JSON in any JSON editor/validator on your choice.

Below the list of application settings that can be set in App Settings JSON object.

Required settings

The settings below must be represent for every environment.

AppName

- Key: `AppName`
- Example (in JSON): `"AppName": "Brief Connect"`
- Description: The name of the application displayed in the UI.

AuthorizedGroupId

- Key: `AuthorizedGroupId`
- Example (in JSON): `"AuthorizedGroupId": "140fa905-7baa-499b-b0b0-088920c7a9ea"`
- Description: Microsoft Entra "Brief Connect All users" Group Id (Object ID) used for users authorization to access the application.

AuthorizedGroupName

- Key: `AuthorizedGroupName`
- Example (in JSON): `"AuthorizedGroupName": "BriefConnect-ABC-Users-G"`
- Description: Microsoft Entra "Brief Connect All users" Group name used for users authorization to access the application.

EmailNotificationsServiceAccount

- Key: `EmailNotificationsServiceAccount`
- Example (in JSON): `"EmailNotificationsServiceAccount": "BriefConnect-Admin@engageau.onmicrosoft.com"`
- Description: Service account UPN used for sending email notifications.

TopHeaderBannerText

- Key: `TopHeaderBannerText`
- Example (in JSON): `"TopHeaderBannerText": "AlphaDev environment with always latest DEV changes"` Or `"TopHeaderBannerText": ""` as empty value.
- Description: Top header banner text shown in the UI; use empty string if not required to show on UI.

Optional settings

SharePoint Uploads

- Key: `FeatureFlags__SharePointUploadsEnabled`
- Example (in JSON): `"FeatureFlags__SharePointUploadsEnabled": "true"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`
- Description: Enables "Select from SharePoint" options in document upload interfaces. When disabled, only local file uploads are available.

Admin Workflow Override

- Key: `FeatureFlags__WorkflowAdminOverrideEnabled`
- Example (in JSON): `"FeatureFlags__WorkflowAdminOverrideEnabled": "true"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`
- Description: Enables admin "Workflow Override" actions to progress records stuck in workflow due to task/role issues. When disabled, only standard workflow actions are available.

Documents Upload Drag and Drop

- Key: `FeatureFlags__DocumentsUploadDragAndDropEnabled`
- Example (in JSON): `"FeatureFlags__DocumentsUploadDragAndDropEnabled": "true"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`

- Description: Shows drag-and-drop upload zones across document upload interfaces. When disabled, users use the standard file picker only.

User Extra Metadata

- Key: `FeatureFlags__UserExtraMetadataEnabled`
- Example (in JSON): `"FeatureFlags__UserExtraMetadataEnabled": "false"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`
- Description: Includes extra endorser metadata (position, group, acting capacity snapshots) in API responses and PDFs. When disabled, this metadata and related UI/PDF sections are omitted.

Plugins Editor

EnablePluginEditor feature flag

- Key: `FeatureFlags__EnablePluginEditor`
- Example (in JSON): `"FeatureFlags__EnablePluginEditor": "true"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`
- Description: Enables Plugin Editor in Admin Panel and plugin bundle generation/injection based on BundleStore configuration. When disabled, Plugin Editor is hidden and no bundle is generated. Requires valid Bundle Store settings (see below)
- Dependencies: When enabled, required correct BundleStore configuration

BundleStore Kind

- Key: `BundleStore__BundleStoreKind`
- Example (in JSON): `"BundleStore__BundleStoreKind": "Blob"`
- Allowed Values: `"Blob"`, `"FileShare"`, `"FileSystem"` (for local dev only)
- Description: Sets the storage type for plugin bundles.
- Dependencies:
 - Blob: requires `BundleStore__ConnectionString` and `BundleStore__ContainerName`; `BundleStore__Folder` optional.
 - FileShare: requires `BundleStore__ConnectionString` and `BundleStore__FileShareName`; `BundleStore__Folder` optional.
 - FileSystem (local dev only): requires `BundleStore__Folder` only.

BundleStore ConnectionString

- Key: `BundleStore__ConnectionString`
- Example (in JSON): is not supported in JSON, see [Infrastructure Settings](#) page
- Description: Connection string to the target Azure Storage account.

BundleStore ContainerName

- Key: `BundleStore__ContainerName`
- Example (in JSON): `"BundleStore__ContainerName": "plugin-bundles"`
- Description: Name of the Blob container that stores bundles.
- Dependencies:
 - Required only when `BundleStore__BundleStoreKind` is `Blob`.

BundleStore FileShareName

- Key: `BundleStore__FileShareName`
- Example (in JSON): `"BundleStore__FileShareName": "plugin-bundles-share"`
- Description: Name of the Azure File Share that stores bundles.
- Dependencies:
 - Required only when `BundleStore__BundleStoreKind` is `FileShare`.

BundleStore Folder

- Key: `BundleStore__Folder`
- Example (in JSON): `"BundleStore__Folder": "bundles/v1"` (Blob/FileShare) or `"BundleStore__Folder": "c:\\DEV\\git\\bc\\client\\public\\"` (FileSystem)
- Description: Folder path within the selected store to organize bundles.
- Dependencies:
 - Optional for `Blob` and `FileShare`. If not set, then `"plugins-static"` value is used.
 - Required full local path when `BundleStore__BundleStoreKind` is `FileSystem`.

Open Records in New Tab

- Key: `OpenDashboardRecordsInNewTab`
- Example (in JSON): `"OpenDashboardRecordsInNewTab": "true"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`
- Description: Controls whether clicking records on the dashboard opens them in a new browser tab instead of the current tab.

Show Validation Errors

- Key: `ShowRecordValidationErrors`
- Example (in JSON): `"ShowRecordValidationErrors": "true"`
- Default value (if not provided): `"false"`
- Allowed Values: `"true"`, `"false"`
- Description: Shows detailed validation error messages to users when record operations fail.

Email Subject Postfix for Proxy

- Key: `EmailSubjectPostfixForProxy`
- Example (in JSON): `"EmailSubjectPostfixForProxy": " (sent on behalf)"`
- Default value (if not provided): `" "`
- Description: Adds a custom postfix to email subjects when emails are sent on behalf of proxy users.

Email Notifications

Sender Type

- Key: `Email__SenderType`
- Example (in JSON): `"Email__SenderType": "SMTP"`
- Allowed Values: `"GraphServiceAccount"`, `"SMTP"`
- Default value (if not provided): `"GraphServiceAccount"`
- Description: Selects the email delivery mechanism: Microsoft Graph using a service account, or direct SMTP.
- Dependencies:
 - When `"SMTP"` is selected, SMTP configuration keys below are required.
 - When empty or `"GraphServiceAccount"` is selected, `EmailNotificationsServiceAccount` is required.

SMTP Username

- Key: `Email__SmtpUserName`
- Example (in JSON): `"Email__SmtpUserName": "smtp-user@contoso.com"`
- Description: Username for SMTP authentication.
- Dependencies:
 - Required when `Email__SenderType` is `"SMTP"`.

SMTP Password

- Key: `Email__SmtpPassword`
- Example (in JSON): `"Email__SmtpPassword": "<secure-password-or-secret-ref>"`
- Description: Password for SMTP authentication.
- Dependencies:
 - Required when `Email__SenderType` is `"SMTP"`.

SMTP Server

- Key: `Email__SmtpServer`
- Example (in JSON): `"Email__SmtpServer": "smtp.contoso.com"`
- Description: Hostname or IP of the SMTP server.
- Dependencies:
 - Required when `Email__SenderType` is `"SMTP"`.

SMTP Port

- Key: `Email__SmtpPort`
- Example (in JSON): `"Email__SmtpPort": "587"`
- Default value (if not provided): `"25"`
- Description: Port number for SMTP server.

SMTP Use Secure Connection

- Key: `Email__SmtpUseSecureConnection`
- Example (in JSON): `"Email__SmtpUseSecureConnection": "true"`
- Default value (if not provided): `"true"`
- Allowed Values: `"true"`, `"false"`
- Description: Sets `EnableSsl` on the SMTP client.

SMTP Timeout

- Key: `Email__SmtpTimeout`
- Example (in JSON): `"Email__SmtpTimeout": "60000"`
- Default value (if not provided): `"100000"`
- Description: SMTP send timeout (milliseconds).

SMTP From Address

- Key: `Email__SmtpFrom`
- Example (in JSON): `"Email__SmtpFrom": "no-reply@contoso.com"`
- Description: From address used in emails.
- Dependencies:
 - Required when `Email__SenderType` is `"SMTP"`;

SMTP Reply-To Address

- Key: `Email__SmtpReplyTo`
- Example (in JSON): `"Email__SmtpReplyTo": "support@contoso.com"`
- Description: Reply-to address used in emails when `Email__SenderType` is `"SMTP"`.

Allowed Upload File Extensions

- Key: `AllowedUploadFileExtensions`
- Example (in JSON): `"AllowedUploadFileExtensions": ".docx,.pdf,.png"`
- Default value (if not provided or empty):
`".docx,.doc,.xlsx,.xls,.pptx,.ppt,.pdf,.jpg,.jpeg,.png,.gif,.tiff,.bmp,.txt,.csv,.mp4,.mov,.avi,.heic,.zip,.msg,.eml"`
- Allowed Values: Comma-separated list of file extensions (e.g., `".docx,.pdf,.png"`), or `"all"` to allow any file type (not safe).
- Description: Limits uploads to the specified file extensions.

Troubleshooting

Feature Flag Not Taking Effect

If a feature flag change doesn't appear to work:

1. **Check Setting Name:** Verify the environment variable name is spelled correctly (case-sensitive)
2. **Verify Value:** Ensure the value is exactly `true` or `false` (lowercase, case-sensitive)
3. **Restart Required:** Restart the Azure Functions App to ensure configuration is reloaded
4. **Check Environment:** Verify you're testing in the correct environment
5. **Check Logs:** Review application logs for any configuration errors

SharePoint Upload Options Not Visible

If users report that SharePoint upload options are missing:

1. Check that `FeatureFlags__SharePointUploadsEnabled` is set correctly
2. Clear browser cache and refresh the application
3. Check in browser Dev Tools that SharePoint connectivity is working properly (no errors when control sends requests to Graph API)

General Configuration Issues

1. **Case Sensitivity:** All setting names and boolean values are case-sensitive
2. **Update variables before the release is created:** When release is created, it takes snapshot of all pipeline variables. You need to re-create release **after** you perform any changes to pipeline variables.