ASSET PUSH
USER GUIDE

STEP Trailblazer

Table of Contents

Asset Push Overview

Asset push allows users to export assets from STEP to a local file system, where they can be accessed by external users and systems. These assets are automatically exported whenever an object is modified or approved (depending on the asset push configuration), ensuring that the assets stored in the local file system are always updated with the latest versions of the assets from the central STEP database. Considering the potentially large size of the assets being exported, this method is favorable over manually exporting assets in bulk.

With asset push, users can maintain a single high resolution image in STEP, and convert it to all the various formats required for use in external systems. Conversion templates can be applied to transform these images from high resolution to low resolution, resize images, apply color scales to images, and more. Images can also be pushed as-is from STEP to make them available to other applications.

It is important to note that asset push does not send asset files directly to downstream systems. Instead, it makes files available for retrieval by these systems so that on-demand extractions from the intermediate file system can occur without putting any strain on the STEP system.

Asset Push Terminology

Term	Definition
Asset	Any collateral or document in electronic format, such as an image, Word document, Excel file, PDF, PowerPoint file, text file, etc.
Asset Push	A process that exports assets from STEP to a file system for different purposes. Conversion templates can be applied to transform images, for example from high resolution to low resolution, resizing, applying color scales, etc, though assets can also be sent as-is.
assetpush.properties file	Properties file that specifies parameters specific to an individual asset push, including login credentials for the local file system. One file of this type exists for each asset push event queue. Note that the ImagesFolder property is included in this file, which indicates the root directory to which the Relative Path Template is applied (e.g. [ImagesFolder][RelativePathTemplate]).
Asset Push Event Queue	Queue to listen for events on assets. Multiple asset push configurations can exist on a single event queue, though one event queue is required per sidecar. In other words, assets being pushed to the same destination system may share a single event queue with one or more configurations assigned to it.
Configurations (as	Definition of how an individual asset push will function, including the conversion,

Term	Definition
related to asset push)	relative path template, and acceptable MIME types. Any number of configurations can exist under a single Asset Push Queue in STEP.
Conversions (as related to asset push)	A series of parameters defining the way in which an image is transformed from the standard high resolution asset that exists in STEP to the required downstream format. Several conversion options are available by default, and additional conversions can be added via custom extensions. Note that conversions change assets as they are pushed, leaving the originating file in STEP unchanged. One asset push configuration is required for each required conversion, as a conversion is simply a parameter within the overall asset push configuration. Note that a Conversion is an optional parameter for each Configuration.
MIME Type	Standard identifier used to indicate the type of data that a file contains. In STEP, each asset push configuration includes an option to specify MIME Types to be used in the configuration. Leaving this entry blank means that all asset types meeting the other configuration parameters will be attempted to be converted and/or pushed as specified. Populating this entry means that conversion and/or push will only be attempted for assets meeting the specified types. For more information about MIME types, see the **MIME Types** section of the **STEP PIM Superuser / System Setup** documentation.
Relative Path Template	Used in conjunction with the asset path specified in the assetpush.properties file to tell the system where to place the pushed assets. Can be used to create a flat structure or a hierarchy structure, and will typically utilize several pre-defined STEP macros to dynamically create directories. The value entered here is used following the root directory specified in the assetpush.properties file (e.g. [ImagesFolder]/[RelativePathTemplate]). For more information about Relative Path Templates, see the **Relative Path Template Overview** section of the **Asset Push** documentation.
Sharedconfig.properties file	Properties file that specifies many system parameters, including some that define general asset push functionality across all asset push event queues on the system.
Sidecar	Optional STEP component consisting of external files installed on various machines, oftentimes on the application server or a remotely located machine. STEP can communicate with sidecars so that when a relevant activity is triggered in STEP, the sidecar is informed of the associated task and carries it out. One sidecar is required for each location to which assets must be pushed. Multiple configurations may share a sidecar if they are also sharing a root destination (as specified in the assetpush.properties file).

Enabling Asset Push

To enable asset push, the user must perform the following actions:

1. Create an image conversion configuration if one is required and none of the standard conversion configurations can be applied. See the **Creating Image Conversion Configurations** section of the **Export Manager** documentation for more information.
2. Create an asset push event queue. See the **Creating Asset Push Event Queues** section of the **Asset Push** documentation for more information.
3. Create one or more asset push configurations. See the **Creating Asset Push Configurations** section of the **Asset Push** documentation for more information.
4. Ensure that the java sidecar application is installed on the client / server where the external file structure is located. The sidecar installation is handled by Stibo Systems Technical Services. Contact your Stibo Systems account manager and/or submit a helpdesk ticket to complete this step.
5. Initiate the first asset push manually. See the **Starting the Asset Push Process** section of the **Asset Push** documentation for more information.

Additional Information

- Assets that have not been pushed or have failed to push can be found by using the 'Unpushed or Failed Assets' search criterion. See the **Unpushed or Failed Assets** topic within the **Navigation and Searches** section of the **Getting Started / STEP User Guide** documentation for more information.
- It may be necessary to monitor the activity of asset pushes, which can be especially important in the event that an asset conversion or extraction fails. See the **Monitoring Asset Push** section of the **Asset Push** documentation for more information.

Asset Push Properties

System Properties

STEP employs a variety of properties to determine some basic system settings for asset push. Many properties are best utilized in their default setting, but may need to be changed in certain situations. These settings can be found in the sharedconfig.properties file and affect all asset pushes in the system.

The tables below describe all system properties related to asset push, and recommendations for when to change them. Note that changing any property in the sharedconfig.properties file may require a restart of the system for the property to take effect.

AssetPush.AutoDetectedExtension.MimeTypes

Definition	This defines a list of mimetypes (separated by comma) of generated content where auto-detected extensions will be applied (if part of template). Otherwise the original extension will be used.
Default	image/*,application/postscript Using the default, autodetected extensions will only be applied to image and postscript files.
Additional Info / When to Change the Default Value	Change only if the autodetect macro is used as part of the relative path template to ensure that all necessary types of files can be detected. Note that the autodetect macro is used sparingly so this property generally does not need to be changed. Can also be used as a means of filtering which assets have their extensions detected as those not listed will be pushed with their original extensions.

AssetPush.Concurrency

Definition	Option to run in concurrency mode. Running in concurrency mode means that more than one sidecar can get access to events in the same queue. Running in concurrency mode has the consequence that events will be marked read immediately, i.e. before processing. If processing fails, the event will no longer be available on the queue. In that case the asset must be touched to have a new event generated.
Default	false

Additional Info / When to Change the Default Value	Typically changed only when a full download / push is being performed as it can speed up this process by allowing multiple sidecars to perform the work. Consult with Stibo Systems Technical Services before changing this or the Concurrency Level property (below).

AssetPush.Concurrency.Level

Definition	Defines the number of concurrent sidecars that will be used when the system is running in concurrency mode (see above).
Default	N/A
Additional Info / When to Change the Default Value	Only set if Concurrency is set to 'true'. See above.

AssetPush.DTPConfiguration

Definition	The default configuration to use from DTP Clients (IDS and QXP) when workspace is Main, in order to obtain pushed assets. The property should be an ID of the configuration.
Default	raw-main
Additional Info / When to Change the Default Value	Used for Print / InDesign asset pushes.

AssetPush.DTPConfiguration.Approved

Definition	The default configuration to use from DTP Clients (IDS and QXP) when workspace is Approved, in order to obtain pushed assets. The property should be an ID of the configuration.
Default	raw-approved

Additional Info / When to Change the Default Value	Used for Print / InDesign asset pushes.

AssetPush.DefaultDTPClientAssetLocation

Definition	Allows a default asset location to be set in the server for use by all DTP clients on site (if they have Use Default Asset Location checked in their preferences). Must be a string in the appropriate form for the DTP client and applicable platform.
Default	N/A (null)
Additional Info / When to Change the Default Value	Used for Print / InDesign asset pushes.

AssetPush.IgnoreClassifications

Definition	Defines a list of classifications (separated by semicolon ';') that will be ignored when an event is fired.
Default	N/A (null)
Additional Info / When to Change the Default Value	Can be set to ignore classifications for asset hierarchy folders that should not be included in asset push and/or for any non-asset classification hierarchies. Better practice is to use the 'Include Classification' setting in the configuration than to exclude via this setting. However, either is acceptable and at least one (if not both) should be populated.

AssetPush.LegalChars

Definition	Legal characters for file name and file structure on the local file system. Any illegal characters encountered (those not in this list) will be converted to underscores. Format=$char$ to indicate a range, or type individual values without using $
Default	N/A (null)

Additional Info / When to Change the Default Value	Should not be used in conjunction with Illegal Characters (use one or the other, or neither). Use when the downstream system has character limitations and allowable characters are limited (e.g. A-Z, a-z, 0-9, hyphen and underscore, indicated as $A-Z$$a-z$$0-9$-_ OR abcdefghijklmnopqrstuvwxyz01234567890-_ ABCDEFGHIJKLMNOPQRSTUVWXYZ).

AssetPush.IllegalChars

Definition	Illegal characters for file name and file structure on the local file system. Any illegal characters encountered (those in this list) will be converted to underscores. Format=$char$ to indicate a range, or type individual values without using $
Default	N/A (null)
Additional Info / When to Change the Default Value	Should not be used in conjunction with Legal Characters (use one or the other, or neither). Use when the file system has character limitations and the allowable characters are significant in number (e.g. standard alphanumerics plus all accented versions allowed but forward slash and asterisk must be omitted, indicated as *).

AssetPush.MaxHeapSize

Definition	The amount of memory to allow the Asset Push payload to use.
Default	256m
Additional Info / When to Change the Default Value	Typically left as defaulted. Consult with Stibo Systems Technical Services prior to changing.

AssetPush.RelativePathTemplate.Sample1

Definition	A pre-defined relative path template. Will be used in the AssetPushConfiguration editor in the Relative Path Template combobox.
Default	$configID$/$contentdimensionpointsID$/$IDpath$/$assetID$.$autodetected-extension$
Additional Info / When to Change the Default Value	Use to provide valid and useful macro templates for asset push administrators.

AssetPush.RelativePathTemplate.Sample2

Definition	A pre-defined relative path template. Will be used in the AssetPushConfiguration editor in the Relative Path Template combobox.
Default	$configID$/$IDpath$/$assetID$_ $contentdimensionpointsID$.$autodetected-extension$
Additional Info / When to Change the Default Value	Use to provide valid and useful macro templates for asset push administrators.

AssetPush.RelativePathTemplate.Sample3

Definition	A pre-defined relative path template. Will be used in the AssetPushConfiguration editor in the Relative Path Template combobox.
Default	$conversion$/$workspaceID$/$IDpath$/$assetID$_ $contentdimensionpointsID$.$autodetected-extension$
Additional Info / When to Change the Default Value	Use to provide valid and useful macro templates for asset push administrators.

AssetPush.RelativePathTemplate.Sample4

Definition	A pre-defined relative path template. Will be used in the AssetPushConfiguration editor in the Relative Path Template combobox.
Default	N/A (null)
Additional Info / When to Change the Default Value	Use to provide valid and useful macro templates for asset push administrators.

AssetPush.RelativePathTemplate.Sample5

Definition	A pre-defined relative path template. Will be used in the AssetPushConfiguration editor in the Relative Path Template combobox.

Default	N/A (null)
Additional Info / When to Change the Default Value	Use to provide valid and useful macro templates for asset push administrators.

AssetPush.UseSystem

Definition	Entry defining which AssetPush system to use. Legal values are: new: The new, event-queue-based one. old: The old one (pre STEP 5.1 and/or December 2009). both: Attempt the new one, and revert to the former if nothing found.
Default	new
Additional Info / When to Change the Default Value	Should always be populated with 'new' unless on a system with a STEP installation prior to 5.1 or being upgraded from a pre-5.1 system (in which case 'both' may be appropriate).

Assetpush.Email.Notification.Minutes

Definition	Time between email notifications for asset push. Must be an integer. Set in minutes.
Default	60
Additional Info / When to Change the Default Value	Update as needed

Event Queue Properties

When a sidecar is installed for an asset push, Stibo Systems Technical Services will also create and populate an assetpush.properties file. This file defines some key information for the specific event queue with which the asset push is associated. The location of the properties file will be provided by Stibo Systems Technical Services.

Note that this file only affects individual event queues, whereas system properties apply to all asset pushes in the system. Additionally, asset properties always need to be defined, whereas system properties can often be left with the default values.

The assetpush.properties file contains the following properties:

Name	Definition
UserName	Username for file system login.
Password	Password for file system login.
ImageFolder	Specifies the first part of the path and/or parent directory into which all pushed assets will be placed. Relative Path Template takes effect after this, meaning that all asset push configurations under a single event queue will share a parent folder on the local file system, but each configuration is expected to have a separate directory structure under the common parent.
Delay	Rate in seconds for sidecar to ping the file system to ensure connection. Always set to 30 seconds and should not be changed without consultation with Stibo Systems Technical Services.

Creating and Maintaining Asset Push Event Queues

Each asset push runs off of an event queue, which defines the local file system the asset is sent to. An event queue can have any number of asset push configurations running from it, each providing their own specific parameters for the asset push, including any required conversions. While the event queue may determine the destination system, assets can be placed in different hierarchies on that system (as defined by the individual configuration of each asset push).

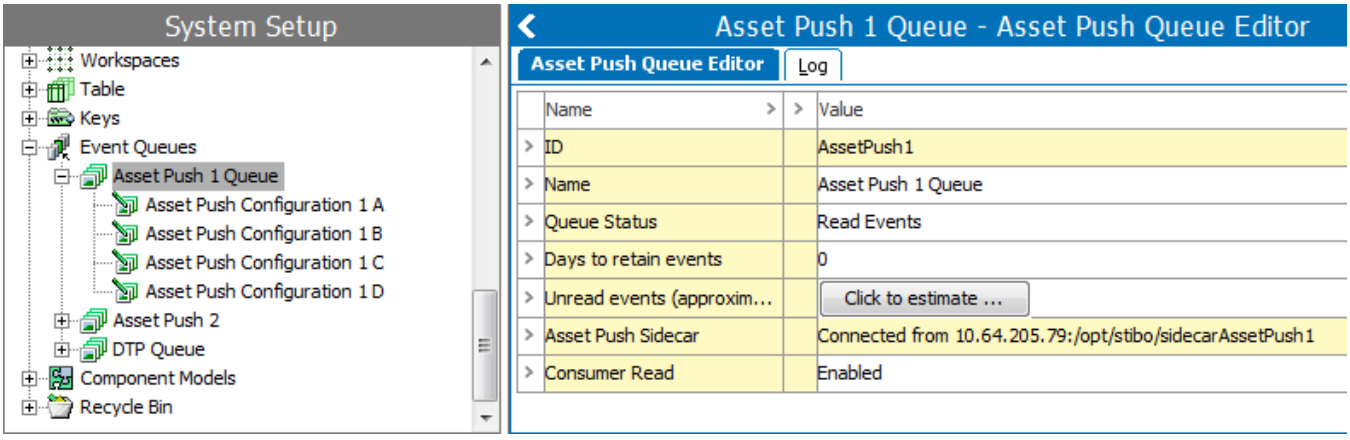
Because multiple asset push configurations can be created for each event queue, multiple queues are only needed if there is a requirement to support different target file systems. If all assets are sent to the same file system, only a single event queue should be created.

Note: The chosen file systems must be directly accessible to STEP. Typically this means pushing to some intermediate location, such as the application server, where downstream systems can access the files without putting any strain on the STEP system.

For more information on setting up asset push configurations, see the **Creating Asset Push Configurations** section of the **Asset Push** documentation.

The below section describes how to set up and maintain an asset push event queue in the STEP Workbench. Additional global configuration options are available via the system properties. For more information on configuring event queue properties, see the **Asset Push Properties** section of the **Asset Push** documentation.

1. In System Setup, right-click **Event Queues** and then select **New AssetPush Event Queue**. The **Create Asset Push Queue** dialog appears.
2. Enter an **ID** and a **Name** for the Asset Push Event Queue, and then click **Create**. A new Asset Push Event Queue node is created in **System Setup**, and the **Asset Push Queue Editor** tab opens.



3. Click the **Queue Status** field, and select either **Discard Events** or **Read Events**. The default status is Discard Events.

- **Discard Events**: Any events that occur while the specified classification folders are being monitored are not processed. **Days to retain events** setting will be ignored.
- **Read Events**: Any events that occur while the specified classification folders are being monitored are processed.

4. Double-click the **Days to retain events** field, and specify for how many days events should be saved.

   The default setting is 0, however if the user wants to reprocess an event that has already been processed, they have to specify the number of days to retain events. Events are typically reprocessed if delivered files are lost and need to be redelivered. This may be set as any integer, though it is recommended to set it between one and ten.

   For more information about rewinding events, see the **Forward, Rewind, Purge, and Republish Events** section of the **STEP Integration End Points** documentation.

5. In **Unread Events**, press **Click to estimate** to view approximately how many unread events are currently in the asset push event queue.
6. The **Asset Push Sidecar** displays the IP address of the sidecar. The address comes from the client that the Java sidecar application is installed on. If no sidecar is detected, the field displays **No activity yet**. A sidecar is an integral part of the asset push and is responsible for carrying out the tasks sent by the event queue. The sidecar communicates with both STEP and the local file system and provides the link between the two.

   Each asset push event queue requires its own sidecar.

Note: Installing a sidecar requires Stibo Systems Technical Services. Contact your Stibo Systems account manager and/or submit a helpdesk ticket to complete this step.

---

7. In the **Consumer Read** field, select either **Enabled** or **Disabled**.
   - **Disabled**: Processed events are held and are not delivered to their final destination. This setting can be used if a user wants to temporarily stop the items from queuing, for example, when the location that receives the deliveries is experiencing problems such as a full disk.
   - **Enabled**: Delivers processed events to their final destination.

The overall functionality of the endpoint is determined by a combination of the settings, with resulting functionality as follows:

- **Enabled + Read Events**: 'Active'; Use for active queue that should deliver assets to downstream systems.
- **Disabled + Read Events**: 'Paused'; Use to temporarily disable the feed, while not losing access to events being generated while disabled. Events will continue to be read and queued, but will be retained within STEP and no attempt will be made to pass them to the downstream system until the Consumer Read setting is changed to 'Enabled'.
- **Disabled + Discard Events**: 'Inactive'; Use when no new events should be processed (now or later) and no assets should be delivered downstream.
- **Enabled + Discard Events**: 'Transition'; Not commonly used but can be employed when one queue will take over from another, or prior to running a bulk update process that should not be sent downstream.

Allows the old queue to process queued events, but not generate any new ones as new events should be set to queue on the new endpoint (or discarded if bulk update is used and events should not be sent out).

Once the asset push event queue has been set up, asset push configurations can be created. For more information on setting up asset push configurations, see the **Creating Asset Push Configurations** section of the **Asset Push** documentation.

Creating and Maintaining Asset Push Configurations

Asset push configurations provide a set of parameters that determine which format to convert assets to, which workspace to extract assets from, which asset folder structure in STEP to monitor for changes, which file formats to convert (e.g. convert TIF, EPS, and JPG images in STEP but not BMP, PNG, or GIFs), and where to place the images on the local file system. Though they are primarily used for converting assets to different formats, some configurations will not have any conversion specified as the asset merely needs to be pushed as-is to the file system.

Any number of asset push configurations can exist within an asset push event queue, but only one is required. A separate configuration must be created for each conversion format desired. Typically, multiple configurations running off of the same event queue will not share a common conversion format unless: a push is needed from both the Main and Approved workspaces, the same assets need to be placed in multiple locations on the local file system for accessibility reasons, or assets from different STEP classifications need to be separated on the file system for organizational purposes.

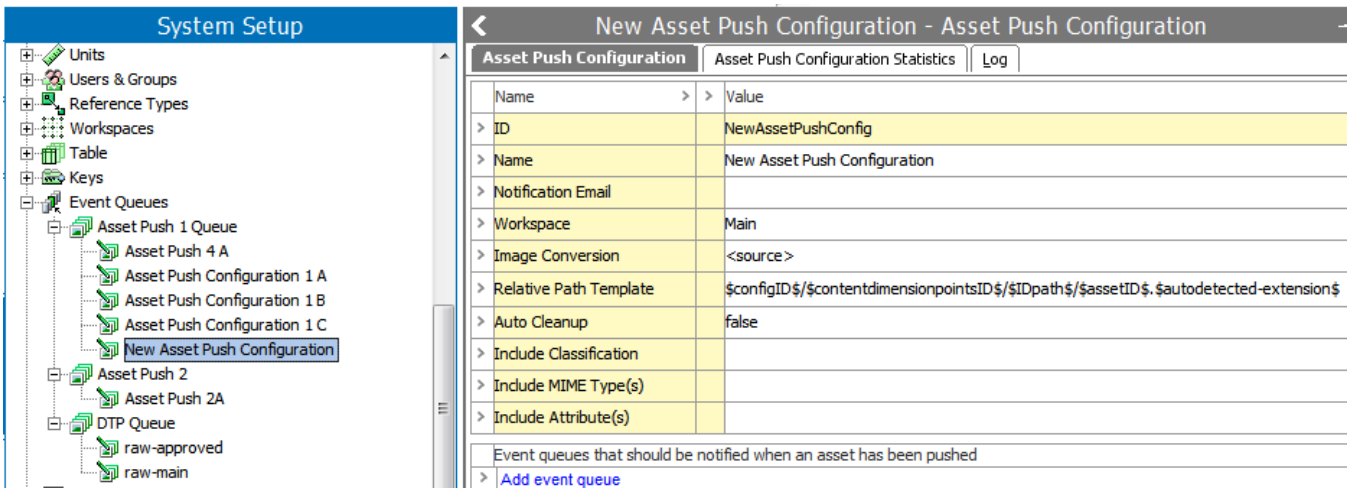
Prerequisites

Prior to creating any asset push configuration, an event queue must first have been configured. For more information on asset push event queues, see the **Creating and Maintaining Asset Push Event Queues** section of the **Asset Push** documentation.

Additionally, if image conversion is required as part of the asset push, the image conversion configuration must be completed prior to the asset push configuration, so that the conversion can be selected as part of the configuration. See the **Creating Image Conversion Configurations** section of the **Export Manager** documentation for more information.

Configuration

1. In **System Setup**, right-click an **Event Queue** and then choose **New AssetPush Configuration**.
2. Enter an **ID** and a **Name** for the asset push configuration, and then click **Create**. A new asset push configuration node is created in System Setup and the **Asset Push Configuration** tab opens.



3. (Optional) Click the **Notification Email** field, and enter a valid email address. If populated, this will result in an email being sent to the specified address in the event that a queued asset fails to be converted and / or extracted.
4. Click the **Workspace** field, and select the workspace from where the assets are to be pushed. If **Approved** is selected, asset push is triggered by asset approval. If **Main** is selected, assets are pushed when they are edited and / or replaced, and no approval is required to trigger the push.
5. If the assets need their format, size, or image color settings converted when pushed to the target system, click **Image Conversion Configuration**, and select the relevant conversion configuration from the list.

   The list contains all of the conversions that are available in the system. Configurations that have (conversion) appended to their names are custom conversions, and those without (conversion) appended are standard conversions.

   If <Source> is selected, no conversion takes place.

   See the **Creating Image Conversion Configurations** section of the **Export Manager** documentation for more information.

6. In the **Relative Path Template** field, specify how to organize the generated file structure. Typically the folder structure is based on the asset's ID, which cannot be changed and ensures the external structure is stable and free from accidental duplication.

   To specify the structure, a string of macros must be entered into this field.

   See the **Relative Path Template** section of the **Asset Push** documentation for more information on the available macros.

7. In **Setting Auto Cleanup**, the user must specify whether they want to keep or remove pushed assets from the target system. Selecting **True** removes previously pushed assets from the target system in the following situations:

   - The asset has been renamed and pushed again
   - The asset push configuration has been changed, and the asset has been pushed again

- The asset has been deleted from STEP and the deletion has been approved

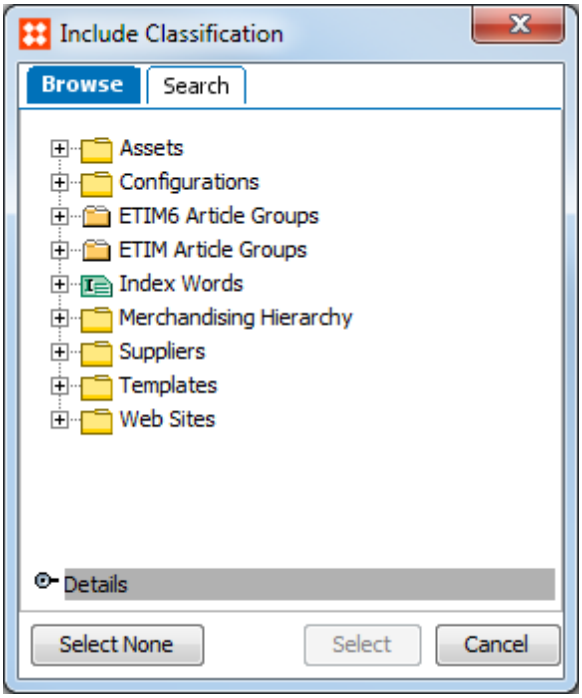
If **False** is selected, STEP will not automatically cleanup the file system.

Note: It is recommended to set Auto Cleanup to 'true' so that manual intervention is not required in maintenance of the file system.

---

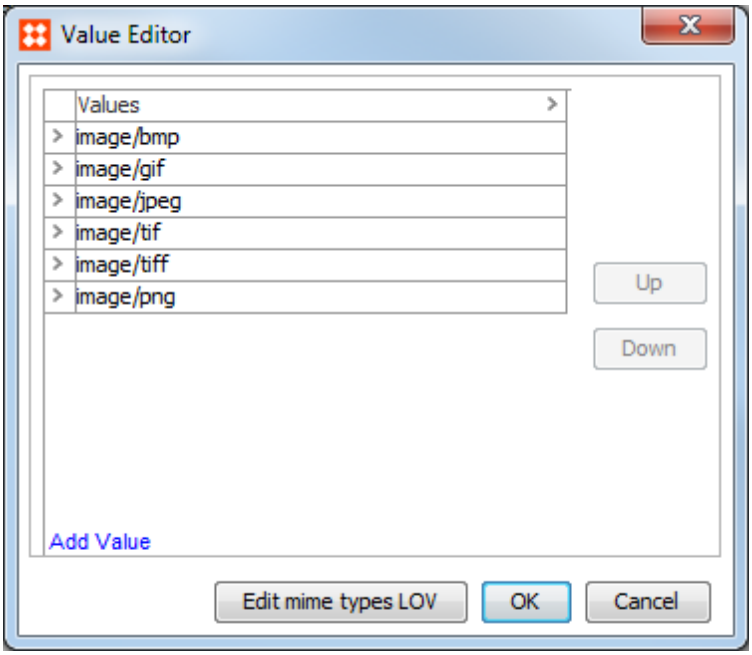
8. Click **Include Classification**, and then click the ellipsis button **(…)**. The **Include Classification** dialog appears. Select which classification node(s) asset push will monitor for changes, and then click **OK**.

Note: This field should not be left blank as that forces the system to monitor all classification folders, of which only a subset are actually valid for assets. At a minimum, the root classification folder for assets should be specified, which causes the system to monitor all child folders in the asset hierarchy.



9. Double-click the **Include MIME type(s)** field to filter which assets are included in the push via their MIME type. In the **Add Value Editor**, click **Add Value** to add the relevant MIME types, and then click **OK**.

   It is recommended that this field be filled, otherwise the asset push configuration will convert all MIME types (e.g. .wav, .xls, .exe), which is typically not required. If all standard images should be included, the image/* and application/postscript MIME types should usually be included.

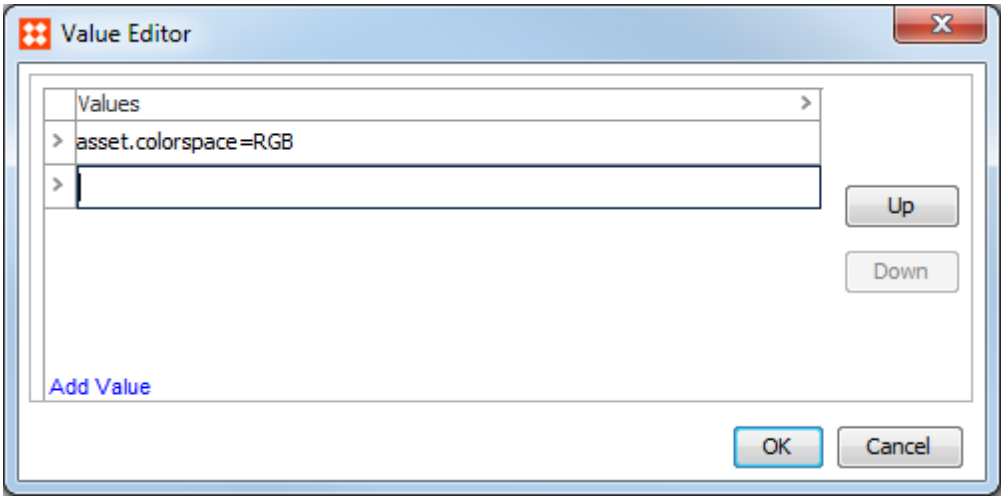


For more information about MIME types, see the **MIME Types** section of the **STEP Superuser / System Setup** documentation .

10. (Optional) Double-click the **Include Attribute(s)** field to specify if the asset push configuration is restricted to assets with specific attribute values. This setting allows users to add some basic intelligence to the conversion without requiring any system extensions. Populating this field means that only assets that have the specified value in the indicated attribute will be pushed.

    In the **Add Value Editor**, click **Add Value** to add the relevant attribute values, then click **OK**.

    Valid inputs require the following format: [Attribute ID]=[Attribute value]. The value can be specified as null by populating nothing after the equals sign (=). Populating the field with multiple attributes acts in an 'AND' fashion (asset must meet both conditions to be pushed).



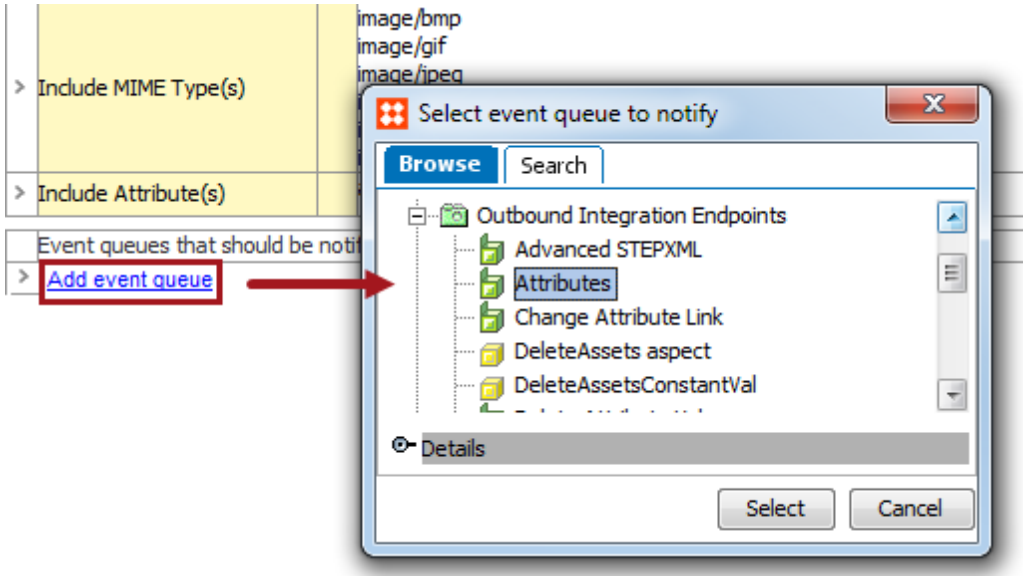
Each STEP asset has a series of metadata attributes that are automatically set upon import of the asset. Users may also create additional metadata attributes and define how they are to be set as part of the import configuration. Note that any assets with user-created metadata attributes must be pushed from the Approved worspace for the **Include Attribute(s)** setting to operate according to the value. If the asset only uses pre-existing system metadata attributes it can be pushed from either the Main or Approved workspaces.

See **System Metadata Attributes** below for a list of all metadata attributes that are automatically set upon import of the asset.

Note: When using this functionality, it is important to ensure that the selected attribute(s) are both valid on the asset types being handled by the push, and that the assets will have value(s) for the attribute(s).

---

11. (Optional) Click **Add event queue** and select an integration endpoint from the pop-up window. Populating this setting means that the listed queues will be notified when an asset has been pushed, and an event will be created on the queue if applicable.



Note that asset push sends only the actual asset files to a downstream system. Oftentimes it is useful to also provide an XML or other file type that contains the asset metadata, including the Relative Path of the asset so that users can easily find it on the downstream system.

Outbound endpoints that send asset metadata are often triggered at approval, which is typically the same trigger used for the asset push. This means that the Relative Path for any asset is not yet populated at the time of the outbound metadata file trigger. When the asset Relative Path needs to be included in the outbound metadata file, it is then necessary to notify the outbound queue following the push (using this setting). This triggers an event on the outbound file queue so that the metadata is sent (again) and includes the Relative Path.

12. When a new configuration is created, or an existing configuration is updated, it may be necessary to restart the asset push sidecar.

System Metadata Attributes

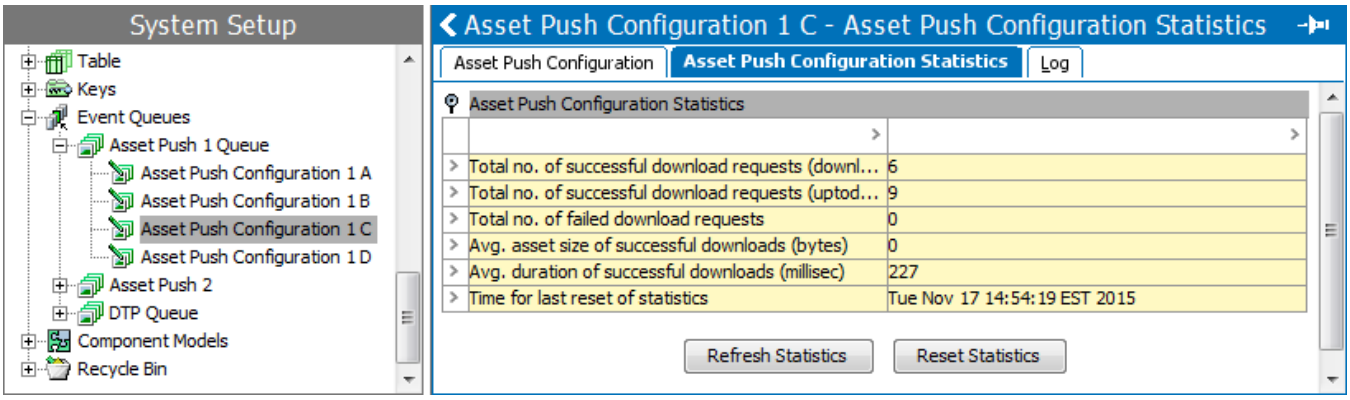
The following asset metadata attributes exist as defaults on the system and are populated automatically for any asset that contains the relevant information:

- asset.class
- asset.colors
- asset.colorspace
- asset.compression
- asset.creator
- asset.depth
- asset.dsc-conformance
- asset.extension
- asset.filename
- asset.format
- asset.format-version
- asset.height
- asset.mime-type
- asset.pages
- asset.pixel-height
- asset.pixel-width
- asset.preview
- asset.preview-format
- asset.profile
- asset.samples
- asset.size
- asset.uploaded
- asset.width
- asset.xdpi
- asset.ydpi

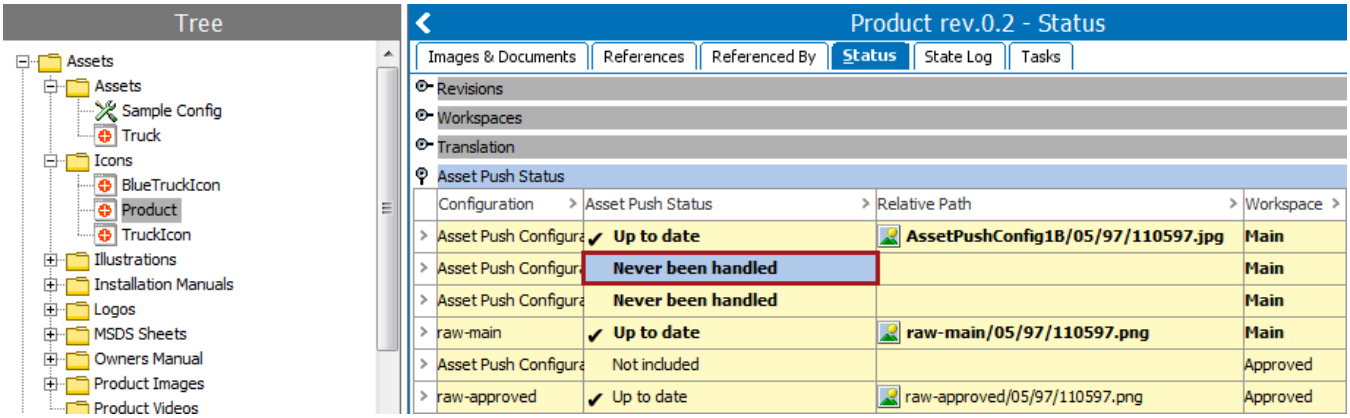
Monitoring Asset Push

It may be necessary to monitor the activity of asset pushes, which can be especially important in the event that an asset conversion or extraction fails. Listed below are four methods of monitoring the status of asset pushes within the STEP Workbench. If additional monitoring is required, it is recommended that standard system monitoring tools are utilized (e.g. designated external monitoring system). Additionally, the STEP System Administration page can be used.

- Each asset push configuration can be set up to send an email when any assets fail attempted extraction and/or conversion. See step 3 of the **Creating and Maintaining Asset Push Configurations** section of the **Asset Push** documentation for more information.
- Assets that have not been pushed, or have failed to push, can be found by using the 'Unpushed or Failed Assets' search criterion. See the **Unpushed or Failed Assets** topic within the **Navigation and Searches** section of the **Getting Started / STEP User Guide** documentation for more information.
- Statistics regarding each asset push can be viewed on the Asset Push Configuration Statistics tab.



- The status of each individual asset relative to all asset push configurations can be viewed on the Status tab of the asset, under the Asset Push Status flipper.



Relative Path Template

Rather than using the file structure in STEP, an asset push configuration specifies how to organize the generated external file structure via a Relative Path Template (RPT). Typically the folder structure is based on the asset's ID, which cannot be changed and ensures the external structure is stable and free from accidental duplication. This means that it is not possible for the same image to occur in multiple folders in the external structure (unless multiple configurations are used).

The RPT is used in conjunction with the ImagesFolder property in the assetpush.properties file to tell the system where to place the extracted assets. Specifically, the RPT dictates a path, file name, and appropriate file name extension, with the intent that all the images pushed into the structure have a unique path and/or file name. The RPT is typically populated with a series of predefined STEP macros and also allows users to control whether the assets will be pushed to a flat file structure or a folder hierarchy structure.

Note: RPT takes effect after the path specified in the ImagesFolder property of the assetpush.properties file: [ImagesFolder value]/[Relative Path Template]

The RPT can be constructed using a series of macros for dynamic creation, as well as any static text that is required. Static text can be applied in many ways, including file extensions, folder names, or anything that designates a static folder into which assets and / or child folders should be placed. The system will automatically create all folder structures dictated by the RPT on the local file system, with each slash (/) creating a new directory level.

Important: Care should be taken to ensure that the RPT is constructed in such a way that: file paths do not exceed the length limits of the target system; only legal characters for the target system are incorporated (illegal characters will be automatically replaced by an underscore); and each asset is assigned a unique file name within the target folder.

The following is a list of relevant macros:

Macro	Notes
$configID$	STEP ID of the asset push configuration. Often used as a parental folder level for the RPT.
$contentdimensionpointsID$ / $contentdimensionpointsNAME$	STEP ID / Name of the dimension point of the asset. ID macro should be used whenever possible in place of Name. Required to use a dimension macro if any assets are dimension dependent. System will not save the RPT entry if a dimension macro is not specified and the system includes dimension dependent assets. Note that if any asset types are declared dimension dependent (i.e. the asset's Name is dimension dependent) then the user cannot use the

Macro	Notes
	$assetNAME$ macro for file names, $assetID$ must be used instead.
$IDpath$	Tells the system to autogenerate external file folders using the ID of the asset. Use when a folder structure for assets is needed. Excluding this will result in a flat file structure being created on the local file system (all assets placed within the same folder). It is also possible to use $IDpath3$, $IDpath4$, or $IDpath5$ to specify how many characters to use to create the external file folders. For example, if an asset has an ID of 0123456789, then: • $IDpath$ will generate a file folder structure of 67/89/ • $IDpath3$ will generate a file folder structure of 456/789/ • $IDpath4$ will generate a file folder structure of 2345/6789/ • $IDpath5$ will generate a file folder structure of 01234/56789/ The use of more characters reduces the number of possible asset entries in each folder. Higher numbers recommended if large numbers of assets are being pushed as the user should aim to have fewer than 1,000 assets per folder to allow for navigability on the file structure.
$assetID$ / $assetNAME$	STEP ID / Name of the asset being pushed. It is required that one of these or the attribute:key macro be used to define the file name, otherwise all assets will be pushed with the same file name. ID must be used if any of the assets being pushed are dimension dependent (NAME cannot be used in this case). Best practice is to use ID (rather than NAME) whenever possible. Note that STEPXML only includes asset ID in the product references section so communication of product to asset relationships to downstream systems will require an additional section and/or feed to map asset Names and IDs if the NAME macro is utilized.
$attribute:key$	Extracts values from any Description attribute on an asset so that they can be used as the asset file name. When the asset is pushed, the value of the attribute specified by the key replaces the macro configuration in the file name. Note that this macro can also be used as part of the relative path rather than as the file name, but the intended use is for file name. See below Configure $attribute:key$ Macro section for setup instructions. Dimension dependent Attributes are not available for use as path keys. This is checked when the attribute dimension is changed for an attribute

Macro	Notes
	and when a new asset push template is created. A change in value on the attribute bound to the path key will cause the asset to be re-pushed. If the asset push configuration is set to publish from the Main workspace, any changes made to the relevant attribute value will prompt an asset push. Likewise, if the configuration is set to publish from the Approved workspace, the asset push will trigger when the value change is reflected in the Approved workspace. Changes made to externally maintained attributes will prompt an asset push from either workspace. Note that if 'Auto Cleanup' is set to 'true' in the asset push configuration, the previous version of the updated asset will be deleted.
$autodetected-extension$	Detects the type of file generated by the conversion and applies the appropriate extension. Used only when intelligent conversion is used, whether via asset metadata or a custom conversion, as intelligent conversions may produce different types of output files. Standard conversions produce only one file type per conversion, so static text or the $extension$ macro should be used, and are the preferred methods for applying the file extension. When using this, be sure to insert a text entry of "." before the $autodetected-extension$ macro so that the file name will have the usual period between the name and its extension (e.g. xxx.$autodetected-extension$).
$extension$	Applies the file type extension specified by the conversion. If the configuration does not include conversion, assets are passed through and assigned the same extension they had within STEP. When using this, be sure to insert a text entry of "." before the $extension$ macro so that the file name will have the usual period between the name and its extension (e.g. xxx.$extension$). Either this macro or static text should be used to generate the extension for all standard (non-intelligent) conversions.
$workspaceID$ / $workspaceNAME$	STEP ID / NAME of the workspace which the assets are being pushed from (e.g. Main or Approved). ID macro should be used whenever possible in place of Name.

Macro	Notes
	Typically used only if the system has multiple configurations with at least one pushing assets from each workspace.
$conversion$	Specifies the type of image conversion in the file path. This macro is not frequently used, but can be useful if it is required to keep different versions of the same asset separated by conversion type. Example ($conversion$ is bolded): AssetPushConfig1/**plain-thumbnail**/23/45/12345.jpg

RPT Example

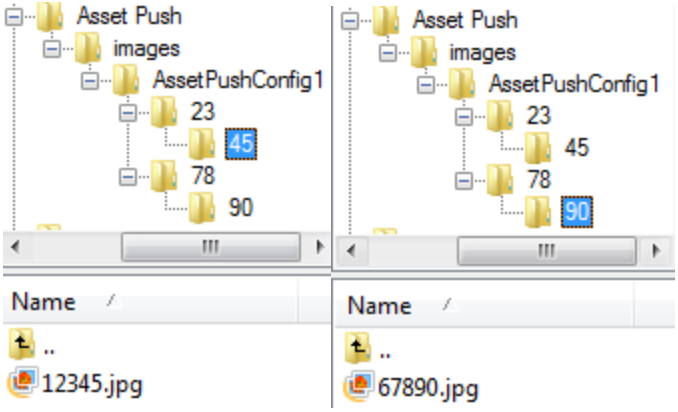
- ID of asset push configuration=AssetPushConfig1
- ImagesFolder property in assetpush.properties file set to 'images'
- RPT in configuration set to '$configID$/$IDpath$/$assetID$.jpg'
  - Assumes conversion is used with resulting file type of JPG
- Two assets are pushed, with STEP IDs of '12345.gif' and '67890.png'
- The resulting Relative Paths in the asset status tab should read:
  - AssetPushConfig1/23/45/12345.jpg

Configuration	Asset Push Status	Relative Path	Workspace
AssetPushConfig1	✔ **Downloaded**	**AssetPushConfig1/23/45/12345.jpg**	**Main**

  - AssetPushConfig1/78/90/67890.jpg

Configuration	Asset Push Status	Relative Path	Workspace
AssetPushConfig1	✔ **Up to date**	**AssetPushConfig1/78/90/67890.jpg**	**Main**

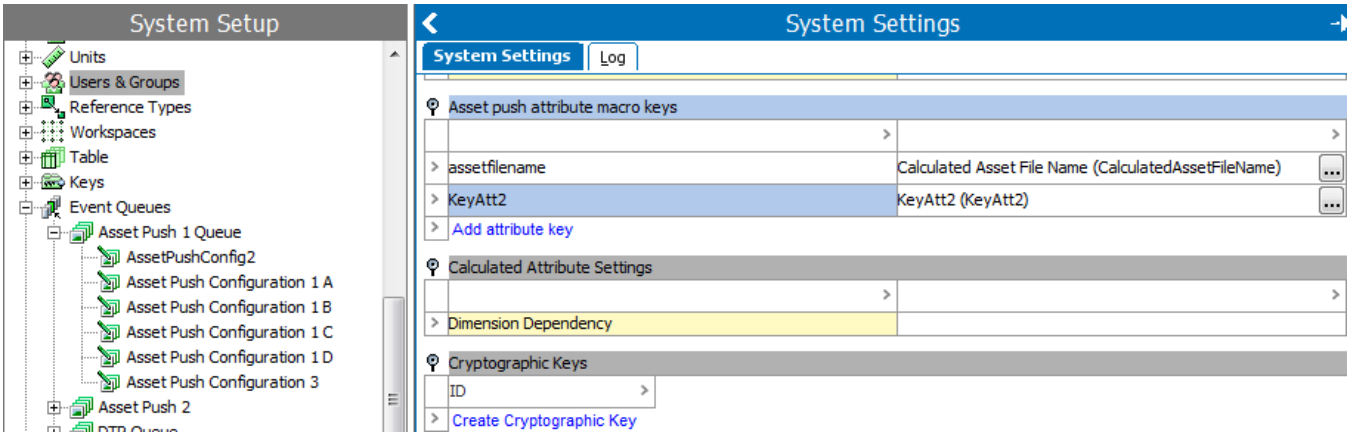
- Once on an external server, they resemble the below images:



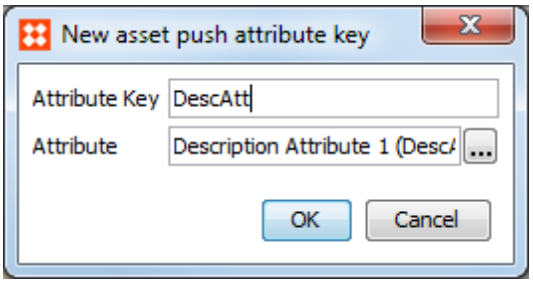
Configure $attribute:key$ Macro

Before the $attribute:key$ macro can be used the applicable attributes must be configured in **System Settings**.

1. In **System Setup**, click on the **Users & Groups** node and navigate to the 'Asset push attribute macro keys' section under the **System Settings** tab.

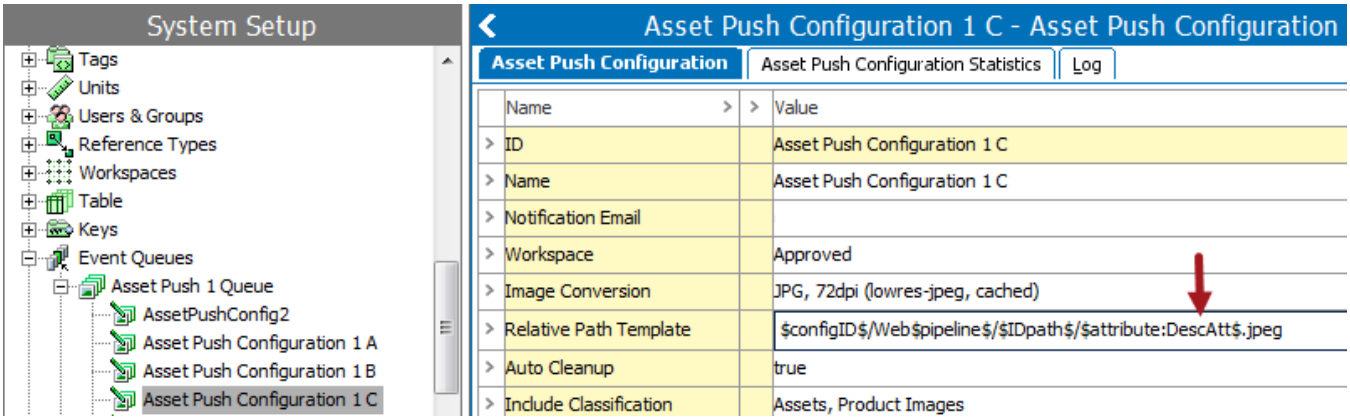


2. Click **Add attribute key**.
3. In the pop up window, enter a name for the Attribute Key. This name can differ from the selected attribute, and is used in place of 'key' in the macro. Click on the ellipsis and select the desired attribute from the list.



Note: On the **System Settings** screen, the Attribute Key appears in the first column and the attribute's STEP name appears in the second column.

4. Return to the desired asset push configuration and enter the desired Attribute Key.

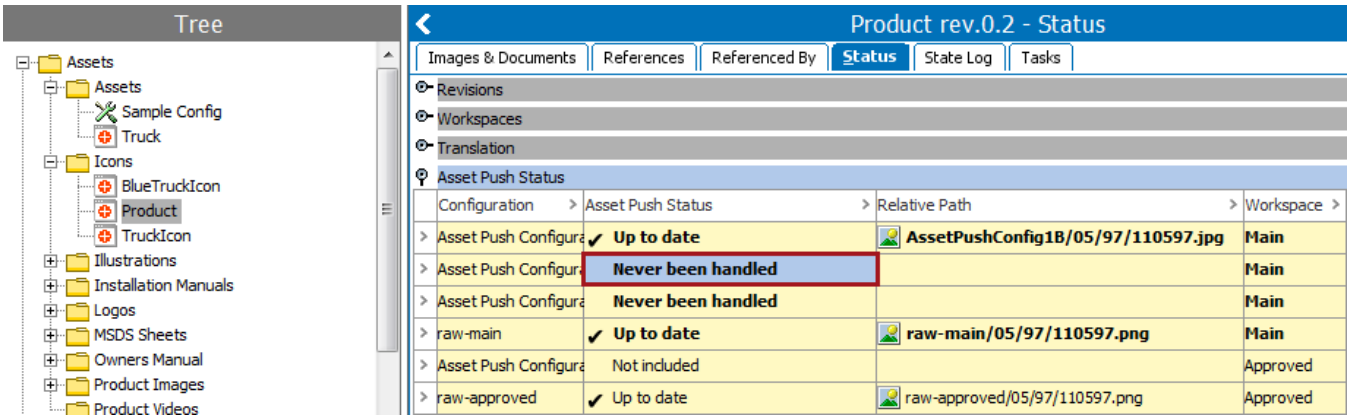


Starting the Asset Push Process

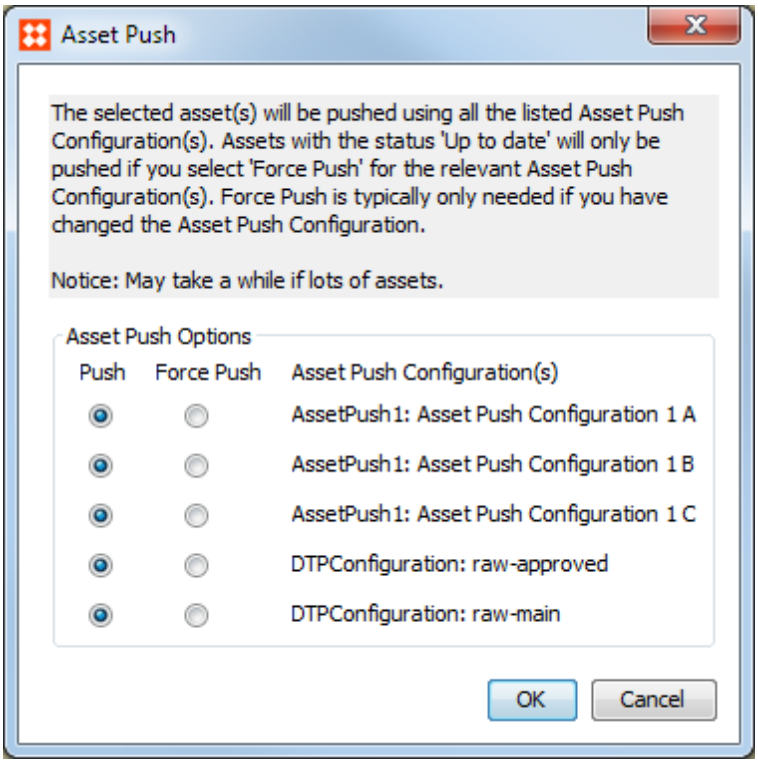
After creating or editing an asset push configuration, an initial asset push must be manually performed. This initial push ensures that the local file system has all of the initial information (assets), and receives only changes from the event queue going forward.

Note that the initial asset push can be a time consuming process so care should be given to the timing and initiation of the push.

1. In the **Tree**, navigate to one of the folders that is being monitored by the asset push configuration, and then click an image.
2. Navigate to the **Status** tab, and in the **Asset Push Status** area, the status should read "**Never been handled**", which means the image has never been pushed.



3. To verify that the process works as expected, right-click the image in **Tree** and choose **Push Asset(s)**. A window appears that lists every conversion associated with this particular folder. Click **OK**.



Note: If the user wants to push assets whose status is set to "Up to date", they have to select **Force Push**.

4. The asset push process is carried out and the **Asset Push Status** changes to **Downloaded** if changes were made or **Up to date** if the downloaded / converted file is already in place and the file-timestamp corresponds to the upload-timestamp on the asset in the database.
5. If the process works as expected, select the top level folder of the folders that are being monitored, and then repeat steps 1-3 above. This starts an initial push of all assets in the top-folder and sub-folders.