DocuBrain®
Scan Agent
Guide
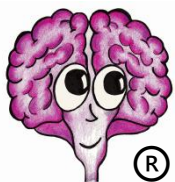
A DocuBrain® Product   By Prevo Technologies, Inc.

https://docubrain.com/   https://prevo.com/

DocuBrain® TechDoc Scan Agent Guide

By Prevo Technologies, Inc.

Table of Contents

Revision History

Revision	Date	Comments
10	02/10/2020	Initial Scan Agent 10 Release

1. Introduction

Scan Agent is a versatile tool for importing documents into the TechDoc Document Manager. Scan Agent automates the process of importing documents from Ascent Capture, Adobe Capture, and PC Folders.

2. Scan Agent Configuration

The Scan Agent configuration tool *(ConfigScanAgent)* provides a simple way to configure Scan Agent settings without having to edit INI files manually. The configuration is broken up into two main tabs "Global Settings" and "Worker Settings". The "Global Settings" tab is used to create and delete workers, and configure global email and log retention settings. The "Worker Settings" tab is used to configure various settings for created workers.

3. Global Settings

The "Global Settings" tab is where workers can be created and deleted. This tab also contains global email settings such as the SMTP gateway, and set the number of days to retain log files.

3.1. Create a New Worker

On this dialog, workers of three different types can be created, ascent, capture, and auto upload. By default, when no workers have been created yet, there will be three template workers listed, *AscentWorker*, *AutoUploadWorker*, and *CaptureWorker*. If one of the template workers is used to create a worker, is will disappear from the drop down. The drop down also contains a worker called custom. Custom workers can be given any name that ends in "Worker" and configured to be one of the three worker types.

3.2. Delete an Existing Worker

Workers can be deleted using this dialog. When deleting a worker, a message will be displayed asking if the worker's folders and INI file should be deleted as well. If the worker's folders and INI file are not deleted, they can be later restored using the "create a new worker" dialog. However, if the worker's folders and INI file are deleted, all of the settings for that worker are permanently deleted.

To restore a worker, use the "create a new worker" dialog. Select the worker if it was a template worker, or select custom and enter the name that was used by that worker and click "OK". If the worker was deleted without deleting its configuration files, a message will be displayed asking if the worker should be restored.

3.3. Global Email Config

This dialog is used to configure global email settings. All email from Scan Agent can be turned on or off here, when enabled an SMTP gateway must be specified. Scan Agent will retain email just as it does log files. Email successfully delivered and email that fails to be delivered is retained separately and both can be retained for a different number of days.

3.4. Log File Settings

Currently, the only log file setting is retention days. Retention days specifies the number of day log files to keep before purging them. For example, if retention days is set to 20, there will always be 20 log files kept, one file for the current day, and 19 for the previous 19 days.

3.5. Test All Worker Configurations

Test all workers validates the connection settings, email settings, document and keyword settings, etc., for all configured workers. Test all workers perform the same check as using the worker settings page, selecting each worker and testing it individually.

4. Worker Settings

The "Worker Settings" tab is used to configure all of the various settings of created workers. The worker to configure should be selected from the list. The worker is then configured by clicking the various buttons for each of the different groups of settings and filling out those dialogs appropriately. If a *Capture* worker is used, the set raw input attributes button will be grayed out, as they are not used by *Capture* workers.

4.1. Set Automapping Keywords

When Scan Agent goes to create or update a document in TechDoc, it looks at all of the attributes for the given document. Scan Agent then uses those attributes to fill in all of the information for the document such as *DocNumber*, *DocType*, *Owner*, etc., and the document's associations such as *AccessLists*, *AccessUsers*, etc. Any attributes that are not mapped to document metadata or document associations can optionally be treated as keywords.

When "Automapping Keywords" is enabled, any attribute that is unrecognized as a document metadata field or association will be sent with the document as a keyword. It is important to note that the keyword(s) must already exist in TechDoc. If an unrecognized attribute is auto mapped as keyword, sent to TechDoc, and does not already exist, an error will be generated. When "Automapping Keywords" is disabled, any attribute that is unrecognized as a document metadata field or association will be ignored.

Sometimes it may be necessary to use automapping keywords, but still ignore certain attributes. When "Automapping Keywords" is enabled, any attributes that are not document metadata or associations and are not desired as keywords can be added to the list of attributes to ignore. When Scan Agent encounters any of the listed attributes it will simply skip over it.

4.1.1. Add a New Attribute to Ignore

Adding an attribute to ignore can only be performed when "Automapping Keywords" is enabled. If there is an attribute that is not a document metadata field or association and is not desired as a keyword, it should be entered here.

4.1.2. Edit an Existing Attribute to Ignore

Attributes to ignore are only used when "Automapping Keywords" is enabled. Attributes to ignore are attributes that are not document metadata or associations and should not be treated as a keyword. If there was a previously entered attribute to ignore, it can be edited here.

4.2. Set Document Associations

Document associations are used to assign access and mail (comments, distribution, and notification) associations to a document when it is being created or updated. The following table describes the available associations and the valid usage for each one.

Association Attributes	Required	Usage
AccessLists	No	*list1:Read* or *list1:Read; list2:Owner; list3:Read,Modify*
AccessUsers	No	*user1:Read* or *user1:Read; user2:Owner; user3:Read,Modify*
CommentLists	No	*list1* or *list1; list2; list3*
CommentUsers	No	*user1* or *user1; user2; user3*
DistributionLists	No	*list1* or *list1; list2; list3*
DistributionUsers	No	*user1* or *user1; user2; user3*
DistributionRemoteEmails	No	*rem1@email.addr* or *rem1@email.addr; rem2@email.addr*
NotificationLists	No	*list1* or *list1; list2; list3*
NotificationUsers	No	*user1* or *user1; user2; user3*
NotificationRemoteEmails	No	*rem1@email.addr* or *rem1@email.addr; rem2@email.addr*

On access for lists, a list is separated from the access permissions by a colon (:). Multiple access permissions are separated by a comma (,). Multiple list/access permission pairs are separated by a semicolon (;) or a vertical bar (|).

On access for users, a user is separated from the access permissions by a colon (:). Multiple access permissions are separated by a comma (,). Multiple user/access permission pairs are separated by a semicolon (;) or a vertical bar (|).

The valid values for access permissions are: *None*, *Read*, *Modify*, *Delete*, *Reserve/Replace*, and *Owner*. *None* and *Owner* must always be used by themselves.

On the mail associations, they are simply one or more lists, usernames, or remote SMTP addresses separated by a semicolon (;) or a vertical bar (|).

4.2.1. Edit a Document Association

An association can be manually edited or the "Edit" button can be used for assistance in building the value. The value entered in this line must be a valid JavaScript expression that evaluates to a string. To enter a simple literal value, it must be surrounded by

quotation marks. The other common expression to use in a document association line is *input.get("NAME")* where *NAME* is the name of the input attribute to use for the value. See section 5.1.5 for more information on using the JavaScript input object.

4.2.2. Document Association Assistant

The "Document Association Assistant" dialog is used to help construct the association's value. Note that the assistant only helps with building a literal string for the value. If advanced JavaScript features are needed, the value must be manually edited on the "Edit a Document Association" dialog.

4.2.3. Lookup a Document Association

The "Lookup a Document Association" dialog is used to find a particular user, list, or remote email that exists in TechDoc. When associating access, checkboxes are available to select the desired permissions.

4.3. Set Document Attachments

Document attachment settings are used to attach files to a document or its generation when the document is being created or updated. The document attachment settings are processed differently from most other settings in that they are automatically indexed starting from 1; although the index number is not directly specified in the setting name. The following table describes the available document attachment settings and the valid usage for each one.

Setting Name	Required	Notes
AttachmentFile#	Yes	This is the full path of the file name to upload to TechDoc as the attachment. If missing or evaluates to blank, no more attachments will be process for the current document.
AttachmentDescription#	No	The optional description of the attachment.
AttachmentExternalUUID#	No	The optional external UUID to associate with the attachment. Fetch Attachment supports fetching by this UUID so it can be useful for generating URL's with this predetermined value.

Setting Name	Required	Notes
AttachmentFileName#	No	Normally not needed. This is the file name (with no device or path) for TechDoc to save as the original file name. If omitted, Scan Agent will extract the file name from the **AttachmentFile#* setting.
AttachmentMimeType#	No	Normally not needed. This would be a standard mime type such as **text/rtf**. If omitted, TechDoc will automatically determine the appropriate mime type.
AttachmentOnGeneration#	No	Determines if the attachment is placed on the generation. If not, it will be placed on the document. If you include this field, the value must be **Yes** or **No**. Otherwise, it defaults to **No**.

This is an example of what the standard document attachment settings look like:

```
[Default Attachments]
*AttachmentFile#=input.get("*AttachmentFile#" + index)
AttachmentDescription#=input.get("AttachmentDescription#" + index)
AttachmentExternalUUID#=input.get("AttachmentExternalUUID#" + index)
AttachmentFileName#=input.get("AttachmentFileName#" + index)
AttachmentMimeType#=input.get("AttachmentMimeType#" + index)
AttachmentOnGeneration#=input.get("AttachmentOnGeneration#" + index)
```

Scan Agent will loop though the document attachment settings starting with an index of 1. The number 1 is appended to the setting name and then the expression is evaluated. First, the expression for **AttachmentFile#1* is evaluated. If it is not empty, it means there is an attachment. Scan Agent then processes the rest of the document attachment settings for attachment number 1 to look for any optional settings.

If attachment 1 is found, then Scan Agent bumps the index to 2 and repeats the process. The first **AttachmentFile#* setting that evaluates to empty ends the processing and no further attachments will be processed for the document even if they are specified.

4.3.1. Edit a Document Attachment Setting

A document attachment setting is manually edited. The value entered in this line must be a valid JavaScript expression that evaluates to a string. To enter a simple literal value, it must be surrounded by quotation marks. The other common expression to use in a document attachment line is *input.get("NAME")* where *NAME* is the name of the input

attribute to use for the value. Remember for document attachment settings, a JavaScript variable named *index* is available, which contains the current attachment number that is being processed. See the standard settings shown in the previous section for examples of how attachment setting expressions are most commonly used.

4.4. Set Document Attributes

Document attributes are used to fill in the metadata of a document when it is being created or updated. The following table describes the available document attributes and the valid usage for each one.

Attribute Name	Required	Notes
CabinetOrFolder	Yes	The cabinet or folder where the document will be stored in. The cabinet or folder must already exist in the TechDoc system.
CreateMissingFolders	No	Determines if any folders specified in *CabinetOrFolder* are missing, if Scan Agent should attempt to create them so that the path will be valid. If you include this field, the value must be **Yes** or **No**. Otherwise, it defaults to **No**.
DocCategory	Yes	The abbreviation of the document category to associate to the document. The document category must already exist in the TechDoc system.
DocNumber	Yes	The unique document number for this document. If the document number is already in use, this is considered to be an update request instead of a create request.
DocType	Yes	The abbreviation of the document type to associate to the document. The document type must already exist in the TechDoc system.
FetchAccess	Yes	Must be the word **Normal** or **Restricted** to designate that normal or restricted access should be assigned to this generation. **Normal** is most commonly used.
FileName	No	Normally not needed. This is the file name (with no device or path) for TechDoc to save as the original file name. If omitted, Scan Agent will extract the file name from the full path name of local file specified at the end of the line in the Ascent Information file.

Attribute Name	Required	Notes
MimeType	No	Normally not needed. This would be a standard mime type such as **text/rtf**. If omitted, TechDoc will automatically determine the appropriate mime type.
NonresidentURL	No	When specified, this URL identifies where the document or information about the document can be found on the Internet. If the document is released and sent to a Search Manager, this URL will be used as the fetch URL for the document. This should not be specified for resident documents.
Organization	Yes	The abbreviation of the organization to associate to the document. The organization must already exist in the TechDoc system.
Overwrite	Yes	Must be the word **Yes** or **No** to determine whether or not the previous generation (if there is one) will be overwritten by this file.
Owner	Yes	The username of the owner to associate to the document. The user must already exist in the TechDoc system and they must have the privilege and access to create or update the document in the specified cabinet or folder.
PointOfContact	No	Optional plain text field displayed "as is" to document fetchers.
ReadAccess	Yes	Must be the word **None**, **Local**, **Campus**, **Community**, or **Global**. Remember that the available choices on this can be restricted by the document category chosen above.
ReleaseDate	No	The release date is not needed unless you are releasing the document. If you include this field, you must also include the revision field below.
Render	No	If you are releasing the document, the render flag is used to determine if TechDoc should attempt to render the file to a PDF. If you include this field, the value must be **Yes** or **No**. Otherwise, it defaults to **No**.
Resident	No	If you are creating a non-resident document (a document without a file) this attribute should be set as No. If you include this field, the value must be **Yes** or **No**. Otherwise, it defaults to **Yes**.

Attribute Name	Required	Notes
Revision	No	The revision is not needed unless you want the document automatically marked as released. The revision must be unique on the document. In other words, multiple generations (versions) of the document that are released must have different revisions. If the release date is not specified, the current date and time will be used.
SmUpdatePriority	No	The priority of updates sent to any SMs resulting from the transaction. If present, the value must be between **-255** and **255** where **0** is normal. For non-Admins, the range is limited to **-255** to **0**, only Admins can choose a priority higher than normal (zero). If not present, it defaults to **0**.
Title	No	The title of the document, which should normally be included but is not required.
WebSearch	Yes	Must be the word **No**, **Campus**, **Community**, or **Global**. Remember that the available choices on this can be restricted by the document category chosen above.

4.4.1. Edit a Document Attribute

An attribute can be manually edited or looked up using the "Lookup" button. The value entered in this line must be a valid JavaScript expression that evaluates to a string. To enter a simple literal value, it must be surrounded by quotation marks. The other common expression to use in a document association line is *input.get("NAME")* where *NAME* is the name of the input attribute to use for the value. See section 5.1.5 for more information on using the JavaScript input object.

4.4.2. Lookup a Document Attribute

The "Lookup a Document Attribute" dialog is used to find a particular document category, document type, organization, etc., that exists in TechDoc. All of the possibilities will be listed for the selected input attribute. Note that free form input attributes such as, point of contact, reason, revision, etc., cannot be looked up.

4.5. Set Document Keywords

The "Set Document Keywords" dialog allows you to specify the keywords that will be placed on documents when they are created or updated. Keywords that have been

added will be displayed in the list as *KeywordName=Value*. If a keyword is listed that should not be used, it should be deleted rather than given a blank value.

4.5.1. Add a Document Keyword

Enter the name of the keyword to add and the value that it should have. As a quick reference, keywords and their values can be retrieved from the server using the Lookup button.

4.5.2. Edit an Existing Document Keyword

Edit the name of the keyword and/or the value that it should have. As a quick reference, keywords and their values can be retrieved from the server using the lookup button.

4.5.3. Lookup a Document Keyword

The "Lookup a Document Keyword" dialog retrieves from TechDoc all of the keywords (and their values if the keyword is a drop down). Select a keyword from the list. If the keyword is a drop down type keyword, select a value from the value list.

4.6. Set Failure Email

Each worker can be configured to send email in the event of a failure when creating or updating a document. To enable failure email for a worker, email must first be setup under "Global Settings". The failure email message can be custom tailored for each worker such as, the sender, recipient, subject, body, etc.

4.6.1. Add a New Failure Email Line

The "Add a New Failure Email Line" dialog is used to add a new JavaScript line, which is used to change the settings on the email (from, to, bcc, cc, subject), as well as, add content to the body of the email. See section 5.1.3 for more information on using the JavaScript *email* object.

4.6.2. Edit an Existing Failure Email Line

The "Edit an Existing Failure Email Line" dialog is used to edit an existing JavaScript line, which is used to change the settings on the email (from, to, bcc, cc, subject), as well as,

add content to the body of the email. See section 5.1.3 for more information on using the JavaScript *email* object.

4.7. Set Folders

The "Set Folders" dialog allows you to set or change the folders that are used by the worker processes to process files. It is very important that the folders are set correctly. The *pickup* folder is the folder where files are placed that will become documents in TechDoc. The *inwork* folder is where a file is moved to when Scan Agent starts the process of creating or updating a document for the file. After a document has been created or update with a file, it is moved to the *processed* folder. If any error should occur while the document is being created or updated, the file will be moved to the *errors* folder instead of the *processed* folder.

4.8. Set Miscellaneous Options

The "Set Miscellaneous Options" dialog contains various worker settings. The settings and their descriptions are listed in the table below.

Setting	Description
Days Before Purging Error File	The number of days a file that fails to be processed will remain in the *errors* folder.
Days Before Purging Processed File	The number of days a file that is successfully processed will remain in the *processed* folder.
Priority	The priority of this worker from 1 to 10 (with 1 being the lowest priority). Workers with higher priority receive more processing time than workers with lower priority. Unless specifically need, the priority should always be set to 5.
Seconds Between Polls	The number of seconds the worker waits each time before checking the *pickup* folder for new files.
Show Failure Dialog	When the worker fails to process a file, if this box is checked, a failure dialog will be displayed.
Show Success Dialog	When the worker successfully processes a file, if this box is checked, a success dialog will be displayed.

4.9. Set Password

This "Set Password" dialog is used to configure the connection settings used by the worker when creating or updating a document in TechDoc. Refer to the table below for a description of each setting.

Setting	Description
Username	The username of the account to connect with.
Password	The password of the account to connect with.
Verify Password	The password should be entered a second time to ensure it was typed correctly and matches the previous.
Interchange Format	The protocol Scan Agent should use when uploading a document to TechDoc. BDL is the original protocol used by Scan Agent since the beginning. REST is a new protocol that has been added that provides greater upload speeds but only works with TechDoc version 7 or newer.
Servlet Path	The servlet path of the TechDoc server. For example, if you were connecting to *www.example.com*, you would enter the address https://www.example.com/ followed by the servlet prefix which is usually *servlet/* to end up with the value https://www.example.com/servlet/
Service Path	The service path of the TechDoc server. For example, if you were connecting to *www.example.com*, you would enter the address https://www.example.com/ followed by the service prefix which is usually *service/* to end up with the value https://www.example.com/service/
Optional Proxy	Scan Agent supports connecting to TechDoc through a proxy server if needed. If a proxy server should be used, enter the address of the server followed by a colon and the port number for example, 1.1.1.1:1234 or proxy.example.com:5678

4.10. Set Raw Input Attributes

The *Ascent* worker receives input data from the file placed in the pickup area. However, the other workers (*Capture* and *Autoupload*) receive the "raw" file to be uploaded in the pickup area. In order to provide the workers with input data, this feature is used to prompt the user for raw inputs that cannot be hardcoded in the "Set Document Attributes", "Set Document Associations", or "Set Document Keywords" dialogs.

Once the raw input data has been entered by the user, the raw input name and value pairs are stored in the JavaScript *input* object so that entered data can be used when processing the document associations, attributes, and keywords. See section 5.1.5 for more information on using the JavaScript *input* object.

4.10.1. Add a New Drop Down

The "Add a New Drop Down" dialog can be used to add a drop down input that the user would be queried for when Scan Agent goes to process a file. The "Input Name" is the variable name that is placed in the JavaScript *input* object. The "Display Label" is the label that will be presented next to the drop down on the raw input attribute window when it is displayed, when Scan Agent goes to process a file. A drop down can be made sticky meaning that once a value has been selected for the drop down, each time it is shown after that, the previously selected value will be the default selected value. Values are added to and removed from the drop down using the "Add" and "Delete" buttons. Values can be editing by using the "Edit" button and reordered in the list using the "Move Up" and "Move Down" buttons.

4.10.2. Add a New Free Form

The "Add a New Free Form" dialog can be used to add a free form input that the user would be queried for when Scan Agent goes to process a file. The "Input Name" is the variable name that is placed in the JavaScript *input* object. The "Display Label" is the label that will be presented next to the free form input on the raw input attribute window when it is displayed, when Scan Agent goes to process a file. A free form can be made sticky meaning that once a value has been entered, each time it is shown after that, the previously entered value will be automatically entered. The "Type" is the data type of the free form input, i.e. a data, number, string, or URL. The "Display Length" is the number of display character places the input window should show. The "Max Length" is the maximum number of characters the input will accept.

4.10.3. Add a New Drop Down Choice

The "Add a New Drop Down Choice" dialog is used to add a choice to the list of values for a drop down.

4.10.4. Edit an Existing Drop Down Choice

The "Edit an Existing Drop Down Choice" dialog is used to edit a choice in the list of values for a drop down.

4.10.5. Edit an Existing Drop Down

The "Edit an Existing Drop Down" dialog can be used to edit an existing drop down input that the user is queried for when Scan Agent goes to process a file. The "Input Name" is the variable name that is placed in the JavaScript *input* object. The "Display Label" is the

label that will be presented next to the drop down on the raw input attribute window when it is displayed, when Scan Agent goes to process a file. A drop down can be made sticky meaning that once a value has been selected for the drop down, each time it is shown after that, the previously selected value will be the default selected value. Values are added to and removed from the drop down using the "Add" and "Delete" buttons. Values can be editing by using the "Edit" button and reordered in the list using the "Move Up" and "Move Down" buttons.

4.10.6.   Edit an Existing Free Form

The "Edit an Existing Free Form" dialog can be used to edit an existing free form input that the user is queried for when Scan Agent goes to process a file. The "Input Name" is the variable name that is placed in the JavaScript *input* object. The "Display Label" is the label that will be presented next to the free form input on the raw input attribute window when it is displayed, when Scan Agent goes to process a file. A free form can be made sticky meaning that once a value has been entered, each time it is shown after that, the previously entered value will be automatically entered. The "Type" is the data type of the free form input, i.e. a data, number, string, or URL. The "Display Length" is the number of display character places the input window should show. The "Max Length" is the maximum number of characters the input will accept.

4.11. Set Success Email

Each worker can be configured to send email in the event of a success when creating or updating a document. To enable success email for a worker, email must first be setup under "Global Settings". The success email message can be custom tailored for each worker such as, the sender, receiver, subject, body, etc.

4.11.1.   Add a New Success Email Line

The "Add a New Success Email Line" dialog is used to add a new JavaScript line, which is used to change the settings on the email (from, to, bcc, cc, subject), as well as, add content to the body of the email. See section 5.1.3 for more information on using the JavaScript *email* object.

4.11.2.   Edit an Existing Success Email Line

The "Edit an Existing Failure Email Line" dialog is used to edit an existing JavaScript line, which is used to change the settings on the email (from, to, bcc, cc, subject), as well as,

add content to the body of the email. See section 5.1.3 for more information on using the JavaScript *email* object.

4.12. Test Worker Configuration

"Test" validates the connection settings, email settings, document and keyword settings, etc., for all configured workers.

5. JavaScript Expressions

Throughout the configuration of Scan Agent, there are many areas where JavaScript expressions can be helpful. These expressions can be used to generate document numbers automatically, set document metadata fields programmatically, and much more.

The simplest JavaScript expression is to use a literal value enclosed in quotation marks. For example, if all documents being processed were going to have a Doc Type of *SCANNED*, you could use the simple expression:

DocType=" SCANNED*"*

Scan Agent uses the Java Nashorn JavaScript engine, which implements ECMAScript 5.1 specification. As such, a wide range of processing can be performed in each JavaScript expression. For example, if all documents being processed were going to have a Comment of literal text and today's date, you could use an expression like:

Comments="This document was scanned on " + now.format("%m-%d-%y") + "."

5.1. JavaScript Objects

In the subsections below, we will talk about each of the JavaScript objects provided by Scan Agent and how they can be used in JavaScript expressions.

5.1.1. *autogroup* Object

The *autogroup* object provides a grouping system that assists with generating of folder names on the server. You specify how many objects should be contained in the group. When you call for the next entry, an item count is incremented. If the limit is reached, the group number is incremented and the item count is reset to 1. Then the current group number is returned. Normally, the returned group number is appended to the

PTI-TD-00015
Revision 10

current folder name. This allows large numbers of documents to be split up into sub folders preventing the original folder name from containing all of the documents. The *autogroup* object has only one method:

- *next* – Returns the next group number in the sequence.

  Example: *autogroup.next("GROUP_NAME_HERE", numberOfLevels, itemsPerLevel, "SEPARATOR_HERE")*

Normally, *autogroup* is used to assist with the creation of subfolders to prevent too many documents from being placed in a single folder. To use it this way, you would specify the full folder or cabinet path as the *groupName*. This will cause the return value to be tied to that unique folder or cabinet. The *numberOfLevels* refers to how deep you want the subfolders to be created. When working with a smaller set of documents, you may choose to only have *numberOfLevels* set as 1 or 2 whereas gigantic set of documents would need many levels of depth. The *itemsPerLevel* specifies how many documents or subfolders should be placed in a single folder. If you specify the number levels to be greater than 1, the separator will be placed between the levels.

For example, say you specified *autogroup.next("/MyCabinet", 1, 250, "/")*. The first 250 times you call next, it will return "000". The next 250 times you call next, it will return "001". The return value from next should be appended to the path you specified, like so: *"/MyCabinet" + "/" + autogroup.next("/MyCabinet", 1, 250, "/")*.

5.1.2. *autonumber* Object

The *autonumber* object is used to generate numbering sequences automatically. Auto numbering is used primarily for sequential document numbering, but can be used for any metadata field on a document. The *autonumber* object has only one method:

- *next* – Returns the next number in the sequence.

  Example: *autonumber.next("YOUR_PATTERN_HERE")*

To use auto numbering, you must construct a pattern using one or more of the codes from the following list:

- *%H* – 2-digit hour, 00-23

- *%M* – 2-digit minute, 00-59

- *%S* – 2-digit seconds, 00-59

- *%L* – 3-digit milliseconds, 000-999

- *%m* – 2-digit month, 01-12

- *%d* – 2-digit day, 01-31

- *%y* – 4-digit year, 0000-9999

- *%Y* – 2-digit year, 00-99

- *%#* - A unique number associated with this mask starting with 1. %# is special from the other codes in that you can use additional pound signs to force the minimum number of digits that will be inserted. For instance, if the next number is 1 and %#### is specified, then 0001 will be inserted.

The date and time codes are filled with the current date and time that the next method is called. For example, if you created the pattern *autonumber.next("MyTestDocument-%H-%M-%S-%####")* to use for a document number and then uploaded some documents starting at 3:05:23PM, you would end up with documents numbered as: MyTestDocument-15-05-23-0001, MyTestDocument-15-05-23-0002, MyTestDocument-15-05-24-0001, etc. The first number 15 representing the hour of current time of in 24-hour time, the second number 05 representing the fifth minute of the hour, the third number 23 (or 24 as the time continues to pass) representing the 23 or 24 second in the hour, and the last number 0001, 0002, etc. as each document is created.

When creating an *autonumber* sequence for document numbers, it is suggested that you develop a unique scheme for each type of document. For example, if you had a worker that worked specifically with invoices, you may want to create an *autonumber* sequence like *autonumber.next("INVOICE-%Y-%m-%######")*. Each document created with this sequence would end up with a document number of INVOICE-12-03-000001, INVOICE-12-03-000002, etc. The first number 12 represents a 2-digit year of 2012, the second number represents the 2-digit month for March, and the third number represents the incrementing 6-digit number starting at 1. As months and years pass, the month and year of the number will increment respectively. In this example, if we created 150 documents in the month of March, our last number would be INVOICE-12-03-000150. When March then turns to April, the third portion of the number will start over i.e. INVOICE-12-04-000001, INVOICE-12-04-000002, etc.

It is important to note, that the Scan Agent keeps track of every *autonumber* sequence you create in each worker. If you shut down Scan Agent and then restart it later, it will resume each *autonumber* sequence where it left off. Because Scan Agent maintains the next number to use locally, different Scan Agent instances should not use the same *autonumber* mask, as collisions in numbers will occur. Workers on the same Scan Agent instance can safely use the same *autonumber* masks.

5.1.3. *email* Object

The email object is available for use when configuring the success and failure email options. The methods on the email object are:

- *from* – Sets the sender address on the email.

    Example: *email.from("yourAddress@example.com")*

- *to* – Sets the recipient address on the email.

    Example: *email.to("theirAddress@example.com")*

- *cc* – Sets the carbon copy recipients on the email.

    Example: *email.cc("bob@example.com")*

- *bcc* – Sets the blind carbon copy recipients on the email

    Example: *email.bcc("jim@example.com")*

- *subject* – Sets the subject of the email.

    Example: *email.subject("This is an email sent by the TechDoc Scan Agent")*

- *addParagraph* – Used to add a paragraph to email. Multiple paragraphs can be added to an email by specifying multiple *addParagraph* calls.

    Example: *email.addParagraph("This is a paragraph in the email")*

5.1.4. *hostname* Object

The hostname object is used to return the hostname of the computer where Scan Agent is running, and can be used in most fields throughout Scan Agent. This can be helpful for instance in the success or failure email configuration to allow you to specify the hostname of the computer sending the email. There may be multiple Scan Agent instances running on different computers and this could easily help you determine which instance has sent you an email. The hostname object has just two methods:

- *getFullName* – Returns the fully qualified host name of the computer where Scan Agent is running. In the event that the fully qualified host name is not available, the shorter version will be returned instead.

Example: *hostname.getFullName()*

- *getShortName* – Returns the short host name or "localhost" if the host name is not available.

    Example: *hostname.getShortName()*

5.1.5. *input* Object

The input object can be used in the configuration of document associations, document attributes, document keywords, email, and raw input attributes. The input object is populated with name value pairs acquired from the input source used by the worker. The Ascent Worker acquires its name value pairs by parsing the Ascent input file provided for each document. The Capture and Auto Upload Workers can acquire their name value pairs by processing a raw input attribute dialog if configured to do so. The input object has only one method:

- *get* – Returns the value of the specified input attribute. If the input attribute specified does not exist, or is not set, get will return an empty string.

    Example: *input.get("InputAttributeNameHere")*

5.1.6. *LocalFileName* Object

The *LocalFileName* object can be used to return the full name, name and extension, name only, or extension only of the current file being processed. The *LocalFileName* object has four methods:

- *FullPath* – Returns the fully qualified path of the file currently being processed.

    Example: *LocalFileName.FullPath*

- *NameExtension* – Returns just the name and extension of the file currently being processed.

    Example: *LocalFileName.NameExtension*

- *NameOnly* – Returns just the name of the file currently being processed.

    Example: *LocalFileName.NameOnly*

- *ExtensionOnly* – Returns the just extension of the file currently being processed.

  Example: *LocalFileName.ExtensionOnly*

Let us say for example, the current file being processed is located at C:\ScanAgent\AutoUploadWorker\pickup\testFile.txt and you wanted to specify the Title of the Document to be the file being processed. If you entered in the Title field one of the following:

- *LocalFileName.FullPath* – The Title would be *C:\ScanAgent\AutoUploadWorker\pickup\testFile.txt*

- *LocalFileName.NameExtension* – The Title would be *testFile.txt*

- *LocalFileName.NameOnly* – The Title would be *testFile*

- *LocalFileName.ExtensionOnly* – The Title would be *txt*

5.1.7. *now* Object

The *now* object is used to return the current date and time using the format you specify. Now uses the same format as the *autonumber* except there is no %# code. The *now* object has only one method:

- *format* – Returns the current date and/or time based on the pattern you specify.

  Example: *now.format("YOUR_DATE_TIME_FORMAT_HERE")*

To use the *now* object, you must construct a pattern using one or more of the codes from the following list:

- *%H* – 2-digit hour, 00-23

- *%M* – 2-digit minute, 00-59

- *%S* – 2-digit seconds, 00-59

- *%L* – 3-digit milliseconds, 000-999

- *%m* – 2-digit month, 01-12

- *%d* – 2-digit day, 01-31

- *%y* – 4-digit year, 0000-9999

- *%Y* – 2-digit year, 00-99

For example, if you specified *now.format("%H-%M-%S")* for a document revision and started to upload documents at 8:05AM, your documents would have a revision of 08-05-00, 08-05-01, etc.

5.1.8. *utils* Object

The *utils* object provides many different ways to convert and format data. The *utils* object has many methods:

- *coalesce* – Returns the first non-null/non-empty value in the array. You may dictate that the returned value should be of a minimum/maximum length.

    Example: *utils.coalesce(values)*

    Example: *utils.coalesce(values, minLength)*

    Example: *utils.coalesce(values, minLength, maxLength)*

- *convertToDocNumber* – Converts the specified input document number to a valid document number for use in TechDoc by converting invalid characters to _.

    Example: *utils.convertToDocNumber(inDocNumber)*

- *convertToSystemPath* – Converts the specified TechDoc folder path into a system path by splitting the TechDoc folder path into parts using backslashes and/or forward slashes as separators. You may also trim a specified number of leading or trailing characters.

    Example: *utils.convertToSystemPath(inFolder)*

    Example: *utils.convertToSystemPath(inFolder, leadingTrim)*

    Example: *utils.convertToSystemPath(inFolder, leadingTrim, trailingTrim)*

- *convertToTechDocFolder* – Converts the specified system path into a TechDoc path by splitting the system path into parts using forward slashes as separators. You may also trim a specified number of leading or trailing characters.

    Example: *utils.convertToTechDocFolder(inPath)*

Example: *utils.convertToTechDocFolder(inPath, leadingTrim)*

Example: *utils.convertToTechDocFolder(inPath, leadingTrim, trailingTrim)*

- *findYear* – Attempts to find and return the first four-digit number located in the specified string.

    Example: *utils.findYear(value)*

- *limitLength* – Limits the length of a string to the maximum length specified.

    Example: *utils.limitLength(value, maxLen)*

- *lowercase* – Lowercases the specified text.

    Example: *utils.lowercase(inText)*

- *replace* – Replaces each substring of the text that matches the literal target sequence with the specified literal replacement sequence.

    Example: *utils.replace(text, target, replacement)*

- *replaceAll* – Replaces each substring of the text that matches the given regular expression with the given replacement.

    Example: *utils.replaceAll(text, regex, replacement)*

- *uppercase* – Uppercases the specified text.

    Example: *utils.uppercase(inText)*

6. Suggestions and Feedback

We hope you enjoy Scan Agent, but if you use it frequently, we also expect you to say "I like the Scan Agent but I wish it could …" If you find yourself in this situation, please let us know by sending us email at suggestions@docubrain.com

If you do happen to find a problem, please feel free to use the same email above. While we do prefer suggestions and compliments, we also accept criticism (preferably constructive :-)

At Prevo Technologies, Inc., we take the quality of our products very seriously. If you do find an issue or something that you feel can be improved upon in Scan Agent, please let us know.