PleaseReview

an ideagen product

Advanced Features Configuration Guide

Table of Contents

1 Overview

This document is designed as a reference source for PleaseReview System Administrators who are required to implement any of the advanced features of the PleaseReview application.

The aim of this guide is to document PleaseReview's advanced features and provide system administrators with an understanding of the approach to take and identify the skills required to implement these features of PleaseReview.

Within this document we will look at flexible configuration, feedback documents, archiving, additional review roles, metrics, and custom controls. A brief description of these features are as follows:

Flexible configuration - the different approaches to consider if you want to configure the system specifically for different groups of users, and identify the tools available to do this;

Feedback documents - a mechanism for allowing the review owner to collect general feedback on the review by means of bulleted lists and ratings;

Metrics – a function to allow organizations to create an extra data entry/display tab on each review with customized fields, which can record information specific to the organization's requirements;

Custom controls – a function to allow organizations to use an additional customized comment categorization that uses a two-category approach (primary and secondary);

Archive – a mechanism which saves a zip file on disk with the document and comment details whenever a review is deleted from PleaseReview;

Additional review roles – an introduction to some of the additional roles that can be created within PleaseReview.

2 Flexible configuration

2.1 Overview

If you are deploying PleaseReview to a large user base, you may find it necessary to segment and structure the system appropriately. You may find that different user groups are reviewing different types of documents and with different business rules.

PleaseReview has the following features to enable such segmentation:

- Flexible configuration (settings and system policies can be overridden by workgroup or review type);
- Nested workgroups;
- Assignment of users to different workgroup administrators;
- Cost-Center Licensing;
- Review types.

This section explains the key concepts regarding how these features work. It is intended to be read by anyone planning a large PleaseReview implementation.

Although it covers the concepts involved and the rules PleaseReview uses at a high level, it does not explain all the detail about the exact rules, what configuration parameters are available, or how to use the administration screens. All this information is available in the PleaseReview System Administrator User Manual.

We have tried hard to make PleaseReview as easy to use as possible, so if you do not enable these new features, end users generally do not need to be aware of the concepts described here. If you do enable them most will be invisible to end users who will just see their configured PleaseReview workgroup(s) and need not be aware of other configurations in other workgroups. However, where review types are used there will be an impact on the end users.

2.2 Configuration settings

PleaseReview has many configuration parameters which can be used to change the business rules and other behavior. These configurations are applied at the Root directory, and most of these can be applied to a specific workgroup and then are inherited by workgroups below, but can be overridden further down if required. More information on workgroups can be found in Workgroups.

By way of example, let us consider the “Re-Open review” policy which controls whether a review owner can re-open a review which was previously closed or completed. The system-wide default for new installations of PleaseReview is to enable this feature, and this is done by having it pre-set on the root workgroup. If you wanted to change the system-wide value you would change the value of the setting at root, but if you wanted to use a different setting in different parts of the system you could override it in a workgroup and then override it again, if necessary, further down the tree.

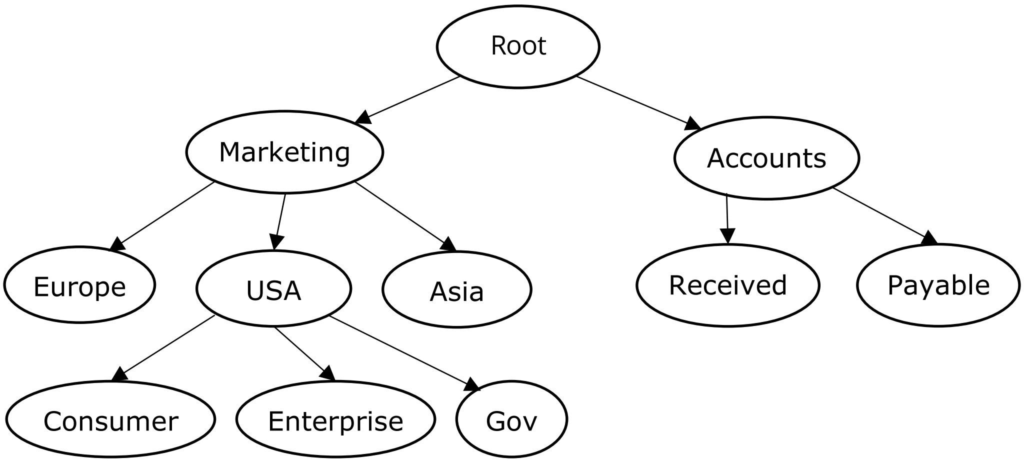
Each setting also carries an ‘allow override’ flag. If the ‘Re-Open review’ setting is set on the root workgroup but has AllowOverride set to ‘no’, then it is not possible for other workgroup administrators to override that policy further down.

As another example you can also change the list of comment categories (Major, Minor, Typo etc.) either globally or in any workgroup.

Some settings interact with other settings or have specific business rules associated with them. The complete list of all settings plus any special notes can be found in the System Administrator Users Guide.

2.3 Workgroups

2.3.1 Nested Workgroups



The picture above represents a hierarchy of nested workgroups. In this case they represent an organizational structure. In practice it is quite likely that higher levels of such a hierarchy might mirror the organizational structure, but lower levels may represent individual projects.

To participate in a review, a user must be placed within a workgroup as a member and given a role (e.g. if a user has a workgroup role of 'Reviewer' it means he may participate in other people's reviews in that workgroup but may not create his own). A user may be a member of any number of workgroups and can have different roles in each. Where workgroups are nested, his role in one workgroup is independent of his role in its parent and child workgroups – i.e. a user that is a member of Marketing»USA is not necessarily a member of Marketing. It is quite likely that higher levels of a workgroup will not actually contain reviews and therefore do not need to have any members (except possibly an administrator who can create sub-workgroups).

2.3.2 Home Workgroup

As well as being a member of each workgroup where he will participate in reviews, each user also has a 'home workgroup' which normally corresponds to his position within the organizational hierarchy. The home workgroup defines who can be an administrator for certain users, and cost-center licensing, both of which are covered later, but it is important to note that a user does not have to be a *member* of his home workgroup. For instance, it might be that all staff in the USA Marketing department have their home workgroup set to Marketing»USA, but because you only conduct reviews in lower level workgroups, they would not actually be a *member* of Marketing»USA (i.e. have a workgroup role in that workgroup) and therefore would not be able to create reviews in it.

2.3.3 Workgroup administrators

When you initially install PleaseReview, a special user account called sysadmin is created. This account cannot be deleted and has universal rights from the perspective of user and workgroup management.

When you create a workgroup, you can give one or more users the workgroup admin permission which allows them to add and remove members from that workgroup and set their roles. However, this does not necessarily give the workgroup administrator any privileges over the user accounts themselves (for instance if Bill from Marketing is added to an Accounts workgroup, the Accounts administrator cannot administer Bill's user account; only the Marketing administrator can do that).

User administration is based on a concept of 'user owner'. The user owner for a user is the person who has administrative rights over the user account, but not necessarily that user's workgroup roles. The user owner is defined as anyone who has workgroup administrator rights on a user's home workgroup or any ancestor of the home workgroup.

For example, if Bill's home workgroup is Marketing»USA then the workgroup administrator for Marketing»USA, Marketing, and root all have user owner privileges over Bill's account.

Note again that the user does not have to be a member (i.e. have a review role) in his home workgroup, and the user owner cannot add/remove/change roles for that user in any workgroups where he is not administrator.

There are several ways you use the workgroup structure to manage users and workgroups, for instance:

1. If each user normally only works in one workgroup, you could set that as their home workgroup and have the workgroup administrator act as their user owner.

2. Have the user administration take place at a departmental level, but create sub-workgroups for individual projects etc. In this case the user would not be a member of his home workgroup. The user can also be a member of other workgroups outside the 'umbrella' of his home department, to allow for cross-functional working.

3. Taking the above example to the extreme, if most working is cross-functional but user accounts should still be administered based on a user's department, you could set up two workgroup trees – an organizational one to use for home workgroups (where there are no reviews) and a different structure for the actual reviews (where no users have their home).

4. Use the sysadmin account for all user and workgroup administration and not bother with any of the new features at all.

5. Don't bother with workgroups at all, just use root (which may be re-named) for everything. In this case, the workgroup administrator for root is user owner for all users (as is the sysadmin of course).

2.3.4 Cost-Center licensing

A single implementation of PleaseReview may serve different user communities that are spread across different cost-centers in the organization, and the central facility that is responsible for providing PleaseReview (for instance IT) may need to cross-charge each user community for the number of PleaseReview license seats they require. The cost-center licensing feature builds on the nested workgroups by allowing the PleaseReview server license to be divided up into licenses for individual departments.

For example, consider the scenario where the Marketing department has budgeted and paid the IT department for 100 seats of PleaseReview but is not using all 100 seats now, whereas the Accounts department has paid for 50 seats and is using them all. If IT purchased a single 150 user license, there is nothing to stop the Accounts department adding more users and 'eating into' the Marketing department's share of the license seats, which means that later when Marketing want to ramp up their usage they discover that seats they have already paid for have been used up somewhere else.

With the cost-center licensing feature, the IT department can install a license key of 150 seats and allocate 100 to Marketing and 50 to Accounts. This is done by a mechanism called 'sub-licenses'. The Marketing and Accounts sub-licenses are created on the system by the system administrator. They do not have to be generated by PleaseReview, because they only allocate the 'master' license of 150 (which is, of course, generated by PleaseReview).

Allocation of users to licenses is done based on the user's home workgroup. In other words, everyone whose home workgroup is inside the Marketing 'tree' in the workgroup hierarchy will be allocated to the Marketing license. Where there is cross-functional working, the license occupied by a specific user depends only on their home workgroup and not the workgroup where the review is taking place.

When we talked about user administration and cross-functional working above, we noted that it was possible to have one set of organizational workgroups to use as home workgroups but to perform most or all reviews in a different set of workgroups not inside any specific department. When we take cost-center licensing into consideration you can see that the licenses must be applied to workgroups where users have their home, but it is possible to have reviews in workgroups which are not covered by a license at all.

It is important to understand that, whilst dividing the user base up into licenses is done using the mechanism of workgroups, you should not think of a license as covering a set of workgroups. Rather, the license covers a set of people.

Sub-licensing does not change the way users are counted against a license. Setting the home workgroup assigns a user to a license but what type of license they take up (or indeed whether they take up a license seat at all) depends on the workgroup role(s) they are allocated in workgroups where they are a member.

More information about how to create and manage sub-licenses can be found in the PleaseReview System Administrator User Manual.

2.3.5 Directory integration

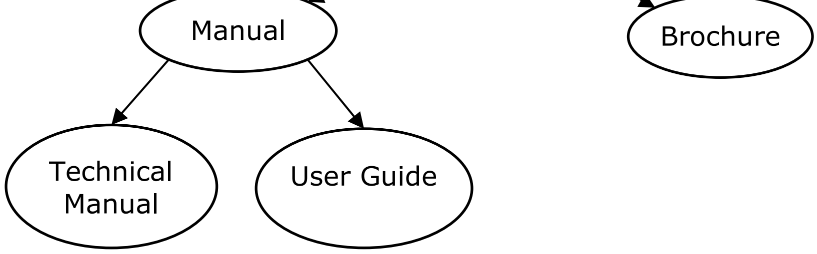
For Active Directory and other integrations which include workgroup mapping (such as Documentum), this works at the level of an individual workgroup as it did in previous versions. The integrations make no attempt to map the organizational structure to the workgroup tree in PleaseReview.

2.3.6 Summary of key points

- Workgroups allow flexibility and different behaviors to be configured for different user groups within an organization;
- Configured and set up by admin, each user is set a home workgroup which defines who can administer the user's privileges and where the user's license is taken from;
- You can force configuration settings, so they cannot be changed within a workgroup, by un-checking the override flag (Note: This does not apply to 'default' settings, such as default duration and default reminders);
- You only need to configure settings where you require a different behavior to that inherited from the Workgroup or its parent/grandparent etc.

2.4 Review Types

t : +44 1629 699 100 | e : info@ideagen.com | w : ideagen.com



Review Types are a mechanism for allowing flexibility within a Workgroup. A Review Type allows you to specify a standard set of review creation parameters (duration, comment categories, reminders etc.). These are created within the admin interface and can then be selected by Review Type name by the review owner when creating a review.

As shown in the picture above, it is also possible to nest Review Types in a hierarchical fashion, with settings inherited from the parent (in a manner like nested workgroups). Most of the review settings that can be specified for a workgroup (for example the comment categories) can also be specified as part of a Review Type.

Note that, in the case where Review Types are nested, the user can create a review of any type available to them, whether it is a 'leaf' or one which has sub-types.

In addition, Review Types are not global; they are attached to a specific workgroup. However, unlike workgroup membership, if a Review Type is attached to a specific workgroup, it is also available in all sub-workgroups.

Information on how to set up Review Types can be found in the System Administrator User Manual.

2.4.1 Advanced Review Type settings

The review template setting within Review Type config settings is a feature which covers a few of the more complex configurations. This requires someone with XML editing skills to edit the template according to the organization's requirements.

The review template setting covers the following features; template type (single phase or multi-phase, defining specific participants and their review role, defining shared distribution lists, and customizing which emails are sent.

In this context, the review template setting is treated as a single 'setting' of a Review Type.

Because Review Types are nested hierarchically, it is possible to create a situation where a parent Review Type has a template attached, but a child Review Type does not. In this case, the child Review Type will inherit the template from its parent, just like any other setting. This can be overridden by having the child specify a blank review template.

A Review Type can be used in any child workgroup of the one where it is defined. The Review Type will need to have shared distribution lists of the appropriate name set up in all workgroups.

Each template has a template type (whether it is single or multi-phase). Using a Review Type which has a single-phase template will automatically create a single-phase review and likewise for multi-phase.

2.4.1.1 Specifying a review template in XML

Adding or editing the 'template' property of a Review type will display a multi-line edit box, into which the XML for the review template should be pasted.

Any special characters such as angle brackets and quotation marks must be entered using the appropriate XML entity reference (e.g. ">" for ">").

All elements and attribute names must be in lower case.

Here is an example template for a single-phase review:

```
<template type="single">
  <subject id="104" role="Reviewer" />
  <subject id="103" role="Contributor" />
  <subject id="105" role="Viewer" />
  <defaultemail value="false" />
  <settings entryname="EmailOptions" stringparam1="Invitation"  />
  <settings entryname="EmailOptions" stringparam1="Finished"  />
  <settings entryname="EmailOptions" stringparam1="EditPublish"  />
  <settings entryname="EmailOptions" stringparam1="EditAllocate"  />
  <settings entryname="EmailOptions" stringparam1="EditRevise"  />
  <review independent="1" />
  <distlist title="*" />
</template>
```

Here is an example template for a multi-phase review:

```
<template type="multi">
  <settings entryname="AutoProgress" intparam1="1" intparam2="1" />
  <settings entryname="TransferComments" intparam1="2" />

  <phase number="1" title ="{0} Phase 1: Initiation" duration="3" description="multi-
phase 1">
       <subject id="104" role="Reviewer" />
       <subject id="103" role="Contributor" />
       <defaultemail value="false" />
       <settings entryname="EmailOptions" stringparam1="Invitation"  />
       <settings entryname="EmailOptions" stringparam1="Finished"  />
       <settings entryname="EmailOptions" stringparam1="EditPublish"  />
       <settings entryname="EmailOptions" stringparam1="EditAllocate"  />
       <settings entryname="EmailOptions" stringparam1="EditRevise"  />
       <review independent="1" />
       <distlist title="*" />
  </phase>

  <phase number="2" title ="{1} Phase 2: Initiation" duration="3" description="multi-
phase 2">
       <subject id="104" role="Reviewer" />
       <subject id="103" role="Contributor" />
       <defaultemail value="false" />
       <settings entryname="EmailOptions" stringparam1="Invitation"  />
       <settings entryname="EmailOptions" stringparam1="Finished"  />
       <settings entryname="EmailOptions" stringparam1="EditPublish"  />
       <settings entryname="EmailOptions" stringparam1="EditAllocate"  />
       <settings entryname="EmailOptions" stringparam1="EditRevise"  />
       <review independent="1" />
  </phase>
</template>
```

In the <template> element, the type attribute is mandatory. <template type="single"> or <template type="multi">

The <subject> element explicitly adds the specified user and role to the review or phase.

The <distlist> element adds all the entries for that distribution list into the review or phase with the roles as specified in the distribution list. Matching of the distribution list title is case insensitive. If there is no shared distribution list of that name in the workgroup, it will be ignored.

Several distribution lists may be specified. Specifying "*" for the title adds all entries from all the shared distribution lists of the selected workgroup. When a subject exists in more than one distribution list, the subject from the top of all distribution lists, sorted in ascending order by title, will take effect.

In the case when a subject is specified in the <subject> element and found in the distribution list specified in the <distlist> element, the role specified in the <subject> element will take effect.

Say the same template is available to multiple workgroups. These workgroups do not share distribution lists (because a distribution list exists inside a workgroup), but for the template to work properly they must each contain a distribution list of that name (they do not necessarily all have the same people in them); otherwise they will just be ignored.

Personal distribution lists are not supported for this feature.

The review owner will always be included in every phase as per normal phased review setup. If they are listed in the <distlist> or <subject> element, this will be silently ignored.

The <phase> element is only used for multi-phase reviews and the phases must be numbered sequentially starting from 1.

The title attribute is used to construct the phase title. If it contains {0}, the top-level review title will be substituted; if it contains {1}, the top-level review id will be substituted.

The title attribute must not be left blank or missing.

The description and duration attributes are used to construct the phase's description and duration respectively.

For a multi-phase template, <subject> and <distlist> may only appear underneath a <phase>; for a single-phase template, they must appear directly under the <template>.

By default, PleaseReview creates the following email options:

1. Invitation - to the review.
1. Alert - the owner is informed when all reviewers have completed the review.
1. Reminder(s) as specified by system configuration parameters.

When the <defaultemail> element is set to true or is missing, the default email options settings will be created as usual. If it is set to false, only emails listed in the template settings section will be created.

When one of the subjects has Contributor permissions, the email option "EditPublish" will be automatically created in the database. However, when <defaultemail> element is set to false, the insertion will not happen as only email options listed in the template settings section will be created.

In general, the parameters match the titles of the user interface elements. However, it is recommended that the individual setting up the template should simply set up dummy reviews and phases using the UI and then look in the database to see which settings have been created and use this to populate the xml file.

Templates created or modified by customers should be tested and will therefore be stable, otherwise if the template has an error this will probably cause a general exception error for the user in the UI.

2.4.2 Summary of Key points

- Review Types allow flexibility within a workgroup;
- Configured and set up by admin, they are then available within the workgroup they have been set up in (and all sub-workgroups below);
- Review Types always override workgroup configuration settings;
- You can force configuration settings, so they cannot be changed within review set up by un-checking the override flag (Note: This does not apply to 'default' settings, such as default duration and default reminders);
- You can default the Review Type, but you are unable to mandate it;

- You only need to configure settings where you require a different behavior to that inherited from the Workgroup or its parent/grandparent etc.

2.5 How configuration inheritance works

In the previous sections we explained how a configuration setting can be set at a higher level and then will either be inherited or overridden by lower levels. You can see that when PleaseReview needs to look for the value for a specific setting, it will start at the bottom of the hierarchy and work upwards, stopping when it finds a value.

2.5.1 Atomicity of settings

Each separate setting is located independently, so for any given review, some settings may be inherited from the review type, some from the workgroup, and some from parent workgroups or root.

Each setting is also considered atomic (i.e. a single unit). For instance, the list of comment categories is always taken from a single place. It is not possible to 'override' (to add or remove) a single category; instead whenever you override you must specify a complete replacement list of categories.

2.5.2 Review settings

Most configuration settings, including the two examples we used earlier, logically apply to a review. If review types are in use, this causes a possible conflict as different values for the same setting could be applied to the review type and the workgroup. The way this is resolved is that settings attached to a review type will always be considered higher priority than those based on the workgroup.

For example, if the Marketing»USA»Gov workgroup contains a review of type 'Brochure', then whenever PleaseReview looks for a review setting it will look in the following places in order:

- 'Brochure' review type
- 'Product Literature' review type
- Marketing»USA»Gov workgroup
- Marketing»USA workgroup
- Marketing workgroup
- Root workgroup

What happens if the search reaches the root workgroup and still doesn't find a value? The answer is that when the system is installed, every possible setting is explicitly created and attached to the root workgroup. Settings can never be deleted from the root workgroup, so this eventuality can never happen.

As you can see, it is possible to create extremely complex structures of both workgroups and review types, and some vigilance is needed on behalf of the administrators to make sure it does not get overly complicated.

2.5.3 User settings

Some configuration settings, for example the width of the comment panel, are a user preference rather than a review setting. In this case, the system will first look for a setting specific to that user, then in the user's home workgroup, and so move up the workgroup tree.

For instance, if user Bill has his home workgroup set to Marketing»USA, when PleaseReview wants to find a user setting, it will look in the following locations:

- Directly attached to user Bill
- Marketing»USA workgroup

- Marketing workgroup
- Root workgroup

As you can see, Bill's user settings do not depend on the workgroup he is working in; if he is part of a cross-functional team working on a review in the Accounts department, his user settings are still defined by his home workgroup but the review-related settings such as the list of comment categories are defined by the workgroup the review is in.

2.5.4 Conflicts within configuration settings

You can also turn off the 'allow override' flag on each setting which restricts the availability of that setting being configured in workgroups and review types further down the tree. It is important to note though, that dependent on when the 'allow override' was turned off will depend on the impact it has over configuration settings further down the tree as it is possible to create apparent contradictions.

For instance, in the earlier example, if the Marketing»USA workgroup defines a set of comment categories with AllowOverride = 'no', a review of a named review type will still use the categories from the review type in preference to those from the workgroup. Note though, that this apparent contradiction can only occur if the AllowOverride = 'no' has been set after the creation of the review type configuration, otherwise the setting would not have been available in the review type configuration.

This applies the same to workgroups as it does to review types, so, if you define a set of comment categories on Marketing»USA»Gov, those will still apply even if you later define categories on Marketing»USA with AllowOverride = 'no'.

2.5.5 Summary of Key points

- Each configuration is setting is independent, so in any review a setting may be inherited from a Review Type, workgroup, parent workgroup or root;
- Review types always override workgroup configuration settings;
- Apparent contradictions may occur, for example if an AllowOverride flag is set to 'no' after a setting has been set up in a lower workgroup or Review Type, so consideration to using this approach is required early in the set-up of workgroups and/or Review Types.

3 Feedback

3.1 Overview

This section explains the basic Feedback functionality and how to configure it.

The purpose of this functionality is to allow the Owner of the review to include a 'feedback document' which enables review participants to give overall high-level feedback on the document/document set.

Two types of feedback are supported: Bullet points and Ratings:

Bullet points - allow reviewers to collaborate to build a bulleted list using familiar PleaseReview constructs such as reply and accept/reject;

Ratings - allow each reviewer to select from a predefined list of options, which an author can accept and reject.

The feedback document itself is a Word document which must be created using Word styles recognized by PleaseReview to identify rating item types and questions. It can also include instructional text for the participants. Participants will only be able to click on specific text in the document.

The feedback functionality is controlled by two main elements:

1. The feedback document itself which contains paragraphs in certain Word styles (known as trigger styles) that are recognized by PleaseReview;
2. The feedback configuration file which defines which rules are associated with which style.

Note: If the feedback document is not loaded into the review as a 'feedback document' type at the document upload screen, it will be treated as a normal review document regardless of the presence of trigger styles.

PleaseReview comes with a predefined feedback configuration file and a sample feedback document. In order to use the feedback functions, you will need to construct a feedback document based on the sample which actually contains the items you want to collect feedback on.

If you are a review owner and your organization already has a standard PleaseReview feedback document which you are going to use then you probably don't need this manual. You can freely edit the boilerplate text within the feedback document if required.

If you wish to set up a feedback document, section 3.2 will explain which feedback types are available in the standard configuration and how these relate to styles in the word document. If you need to do more advanced configuration, such as changing the definitions or defining your own feedback documents, section 3.3 explains the format of the feedback configuration file.

3.2 Word styles

The following table shows a list of the Word styles which are enabled in the default configuration file and what they will produce. The exact wording of the options shown in the comment window are defined in the configuration file.

Note: PleaseReview is only concerned with the style name. You can change any formatting associated with the style in Word and the feedback functionality will not be affected.

Some of the rating styles are standard US Department of Defense source selection proposal ratings.

Word Style	Type	Description
DoD Ratings		
FB_Risk	Rating	Risk rating
FB_technicalRating	Rating	Technical rating
FB_performanceEval	Rating	Past performance relevancy rating
FB_performanceConfidence	Rating	Past performance confidence rating
FB_RiskTech	Rating	Combined technical/risk rating
Generic Feedback Types		
FB_SpecBullet	Bullet	A bullet list
FB_SpecBulletClsd	Bullet	Bullet list that does not allow replies or like/dislike
FB_DisagreeToAgree	Rating	Allows the user to select a value on a scale of 'Strongly Agree' to 'Strongly Disagree'
FB_SatisfiedToDissatisfied	Rating	Allows the user to select a value on a scale of 'Very Satisfied' to 'Very Dissatisfied'
FB_SurveyCount10	Rating	Allows the user to select a value on a scale of 1 to 10
Other Samples		
FB_RiskNoCount	Rating	Sample of a rating list without displaying how many participants select each rating
FB_TableNoCount	Rating	Sample of a rating table without displaying how many participants select each rating

3.3 Xml configuration file

Feedback documents are defined in an XML configuration file on the PleaseReview server at the location runtime/config/feedbackconfig.xml.

Note: after editing this file, you must restart the TaraService as well as the TaraWeb UI, and if you have existing feedback documents referencing any of the feedback types you have changed, they may not work correctly.

Editing the FeedbackConfig.xml is not for the faint-hearted! This document does not give a definitive specification of every element and certainly would not enable you to define your own config file from scratch, but it gives some pointers to allow you to find your way around the standard file. Hopefully you will be mainly copying sections and/or editing relatively small parts.

To edit this file, you will need to be very comfortable with editing complex XML files (probably using a folding editor), and if you want to change the output formats you will need to be familiar with XHTML and possibly WordProcessingML too. If in doubt, you might get a computer programmer to help you. If you have any specific questions, contact support.pleasereview@Ideagen.com

The following table lists the main elements defined in the FeedbackConfig.xml

Parameter	Value	
feedbackType	Represents a specific feedback type which may either be a 'base' (abstract) type or a final usable type	
	name	A name for the feedback type
	inherits	Specifies which type this one is based on. This means it is possible to create a feedback type by inheriting from a previous type and not having to repeat all the configuration for each type
triggerStyle	The name of the Word style in the feedback document that triggers this specific feedback item	
outputStyle	Defines the style used to output feedback items of this type. See below for more details on this	
options	Sets options for this feedback type	
	allowReply	Only relevant for bullets. 'true' to enable replies to bullets or 'false' otherwise
	maxLength	Only relevant for bullets. Maximum length of a bullet value in characters
	enableLike	Only relevant for bullets. 'true' to enable the like/dislike functionality or 'false' otherwise
bsAdapter	Changing this is outside the scope of this document	
webAdapter	Changing this is outside the scope of this document	
ajaxRefreshHandler	Changing this is outside the scope of this document	
wordRenderXSL	Changing this is outside the scope of this document	
reconReportRenderXSLa	Changing this is outside the scope of this document	
Rating	Defines an individual rating value	
	name	A name for this rating
	title	Title used on the screen and report as required
	description	Description used on the screen and report as required. Description is optional
	color	Color used on the screen and report as required. Note this is optional and if no color is specified, it is not shown

If there is no <outputStyle> element, a system default output style is used.

For <outputStyle>, the child element should be either <paragraphStyle> (to output as a bulleted list) or <tableStyle> (to output as a table). Table style is only applicable for ratings, not bullets.

Either type can optionally have HTML style attributes, and <openContent> and <acceptedContent> sub-elements. The appropriate content element is selected depending on whether the bullet or rating is accepted or not.

an ideagen product
t : +44 1629 699 100 | e : info@ideagen.com | w : ideagen.com

Inside the content node, variable values can be substituted. For bullets, these are:

Parameter	Value
value	The bullet point text, or blank if this is a reply
depth*10	Used to indent replies
reply	The text of a reply, or blank for the original bullet text

For ratings, the substitution values are:

Parameter	Value
color	Color associated with the rating
title	Title of the rating
subjects	Semicolon separated list of underlined names of people who selected that rating
description	Description of the rating
name	Name of the rating
subjectcount	Count of the people who selected that rating

Substitution variables must be enclosed within {braces}. Between the braces you can use [square brackets] to include literal text which will only be included if the variable value is non-blank. For instance, if value is 36 then "abc{value}def" and "{[abc]value[def]}" will both generate "abc36def". However, if value is blank then "abc{value}def" will generate "abcdef whereas "{[abc]value[def]}" will generate blank.

The <wordMLOutputStyle> element is used to define the styles used when the document is downloaded as Word. If this element is not present, the HTML styling defined above will be converted to Word but because of differences between how styling works between HTML and Word, it's likely that this will look wrong.

Like <outputStyle>, <wordMLOutputStyle> has sub-elements <openContent> and <acceptedContent>. The substitution variables that can be used are the same.

If you are unfamiliar with WordProcessingML, you can search the internet for ECMA-376 and download the full specification. However, if you stick to copying bits from elsewhere in the config file and/or from the XML of existing Word documents, you certainly don't need to be familiar with the whole specification.

Note that the {subjects} variable in <worldMLOutputStyle> will generate its own <w:r> and <w:t> element, hence the variable should not be enclosed within in any <w:r> and <w:t>. Because of the way whitespace is treated in WordProcessingML, there should be no whitespace before and after the {subjects} variable when used in the <wordMLOutputStyle> section.

Finally, the <wordMLNumberingOutputStyle> is used to generate the Word list definitions to go in the header of the document used in conjunction with the <wordMLOutputStyle> when the output is a bulleted list.

4 Custom control installation notes

4.1 Overview

The custom control function provides enhanced comment categorization when creating and editing comments and proposed changes in PleaseReview when using the **classic interface only**. The custom control can either be applied to the whole system, a Workgroup (and, by inheritance, on all Workgroups below it) or a review type.

If you wish to implement custom controls, please contact support.pleasereview@ideagen.com to request the zip file.

Note: If using the new interface this can be achieved in the Admin UI -> Workgroup Management, -> Settings -> Review -> Comment Categories.

4.2 Unzip the custom control installation zip file

The Custom Control Installation zip file consists of 2 parts:

- The custom control ascx and dll files (TaraWeb folder)
- The control set repository template file

4.3 Copy the Zip TaraWeb contents to the PleaseReview TaraWeb directory

Copy the contents of the TaraWeb directory from the zip file into your PleaseReview TaraWeb directory.

It is important that the files appear at the correct directory level. If you have copied them correctly, there should be a file called CustomControlDemo.dll inside TaraWeb\bin.

4.4 Update the control set repository file

Locate the ControlSetRepository.xml file in the PleaseReview directory 'Runtime/config'.

If the file exists, you are currently using the Metrics function in PleaseReview and will need to append to the existing file.

If it doesn't exist, locate the ControlSetRepository.xml.template file in the Taraweb\Support\Admin\docs directory, copy it to the PleaseReview directory Runtime\config and rename as ControlSetRepository.xml

Note: After putting this file in the directory for the first time or editing it you will need to restart the web application. See Restarting Taraweb and Taraservice.

4.4.1 Appending to an existing file

Before appending to an existing file, you should check whether the workgroup you wish to apply the custom controls to has metrics tab applied to reviews.

If you wish the custom control to appear in a workgroup that already has the metrics tab applied to reviews, then you need to copy both the <controlSet> elements 'CommentWindowEdit' and 'CommentWindowList' (as highlighted below) from the ControlSetRepository.xml.template file and embed this within your existing <controlSetGroup>.

If you wish the custom control to appear in a workgroup that currently doesn't have the metrics tab applied to reviews, then you need to copy the <controlSetGroup> element (as highlighted below) from the ControlSetRepository.xml.template and embed this within your

```xml
<?xml version="1.0" encoding="utf-8" ?>
<!--
    Copy the following ControlSetGroup into the ControlSetRepository.xml file in the runtime folder
-->
<controlSetRepository>
  <controlSetGroup name="CustomControlSetGroup">
        <controlSet name="CommentWindowEdit" location="commentWindowEdit" path="../CustomControlDemo/CommentWindowEdit.ascx">
                <customProperties>
                        <title name="Advance Defect Categorization"/>
                        <modes>
                                <mode name="new"/>
                                <mode name="edit"/>
                                <mode name="merge"/>
                                <mode name="revise"/>
                        </modes>
                        <control name="Dropdown1" persistence="AdvancedCategorization_CommentType">
                                <option value="Action Item" default="true"/>
                                <option value="Defect"/>
                        </control>
                        <control name="Dropdown2" persistence="AdvancedCategorization_DefectType">
                                <option value="Fatal" filter="Defect"/>
                                <option value="Major" filter="Defect"/>
                                <option value="Minor" filter="Defect" default="true"/>
                                <option value="Cosmetic" filter="Defect"/>
                        </control>
                </customProperties>
        </controlSet>
        <controlSet name="CommentWindowList" location="commentWindowList" path="../CustomControlDemo/CommentWindowList.ascx">
                        <customProperties>
                        <title name="Advance Defect Categorization"/>
                        <control name="dropdown1" persistence="AdvancedCategorization_CommentType">
                        </control>
                        <control name="dropdown2" persistence="AdvancedCategorization_DefectType">
                        </control>
                </customProperties>
        </controlSet>
  </controlSetGroup>
</controlSetRepository>
```

existing <controlSetRespository>.

For this custom control the <controlSetGroup> node is named 'CustomControlSetGroup'.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<!--
    Copy the following ControlSetGroup into the ControlSetRepository.xml file in the runtime folder
-->
<controlSetRepository>
  <controlSetGroup name="CustomControlSetGroup">
        <controlSet name="CommentWindowEdit" location="commentWindowEdit" path="../CustomControlDemo/CommentWindowEdit.ascx">
                <customProperties>
                        <title name="Advance Defect Categorization"/>
                        <modes>
                                <mode name="new"/>
                                <mode name="edit"/>
                                <mode name="merge"/>
                                <mode name="revise"/>
                        </modes>
                        <control name="Dropdown1" persistence="AdvancedCategorization_CommentType">
                                <option value="Action Item" default="true"/>
                                <option value="Defect"/>
                        </control>
                        <control name="Dropdown2" persistence="AdvancedCategorization_DefectType">
                                <option value="Fatal" filter="Defect"/>
                                <option value="Major" filter="Defect"/>
                                <option value="Minor" filter="Defect" default="true"/>
                                <option value="Cosmetic" filter="Defect"/>
                        </control>
                </customProperties>
        </controlSet>
        <controlSet name="CommentWindowList" location="commentWindowList" path="../CustomControlDemo/CommentWindowList.ascx">
                        <customProperties>
                        <title name="Advance Defect Categorization"/>
                        <control name="dropdown1" persistence="AdvancedCategorization_CommentType">
                        </control>
                        <control name="dropdown2" persistence="AdvancedCategorization_DefectType">
                        </control>
                </customProperties>
        </controlSet>
  </controlSetGroup>
</controlSetRepository>
```

Note: As the file already existed in the directory you do not need to restart the web application.

4.4.2 Editing the XML to meet requirements

If you want to edit the custom controls, you can do so, most of the XML should be self-explanatory, but a few key points to mention;

controlSetGroup

You can have multiple sets of custom controls and apply them at different workgroups (in the same way as any configuration setting, they automatically will apply to all workgroups below unless overridden).

The mechanism for this is the <controlSetGroup> element at the top of the file. You can create multiple control set groups within the XML file and give each one a unique name. For this custom control the <controlSetGroup> node is named 'CustomControlSetGroup'.

Further on in the process, when you enable the custom control function, enter the ControlSetGroup name that relates to the custom control you want to apply to the workgroup.

controlSet

There are two <controlSet> elements required for custom controls, 'CommentWindowEdit' and 'CommentWindowList'.

'CommentWindowEdit' relates to the part of the comment window where you create new comments, or edit, merge, or revise existing comments.

'CommentWindowList' relates to the part of the comment window which displays the list of all comments made on the selected paragraph.

Parameter	Purpose	Example
Title name	This is the title that appears on the screen above the custom controls.	E.g. "Advanced Defect Categorization"

control name

Parameter	Purpose	Example
Control name	This is a unique reference for the control name created in controlSet 'CommentWindowEdit' and then referred to in controlSet 'CommentWindowList' Note: There are always 2 controls here and the names should be left unchanged. ("Dropdown1" and "Dropdown2")	
persistence	This will help you identify the metrics within the T_Settings table. More information on this can be found below.	E.g. Project Identification Code
Option value	These are the values you wish to appear in the dropdown list	E.g. "Major"
filter	This is specific to the secondary dropdown, and the filter will restrict	E.g. "Defect"

	which values appear dependent on the value selected in the primary dropdown.	
default	You can set one of the optional values as a default value	

persistence

One part of the configuration to be aware of is the <persistence> element.

This ensures the metrics values are stored in a table called T_Settings in the database.

You can see from the sample XML file that it will create two T_Settings entries for each comment, one with entry name is "AdvancedCategorization_CommentType" and another with the entry name "AdvancedCategorization_DefectType" which will have the values selected from the dropdown lists stored in StringParam1.

In addition to this the RefObjectType="Comment" and the RefObjectID=the ObjectId from the T_Comment table for that comment. See the section 6 below on extracting the data.

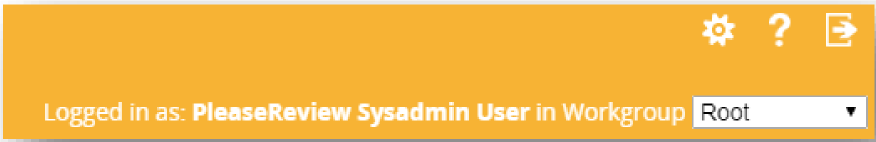
4.5 Add the Configuration setting to the workgroup

Finally, you need to enable the custom control function within PleaseReview. (Unless you have appended the XML to an existing controlSetGroup, in which case it will already be applied)

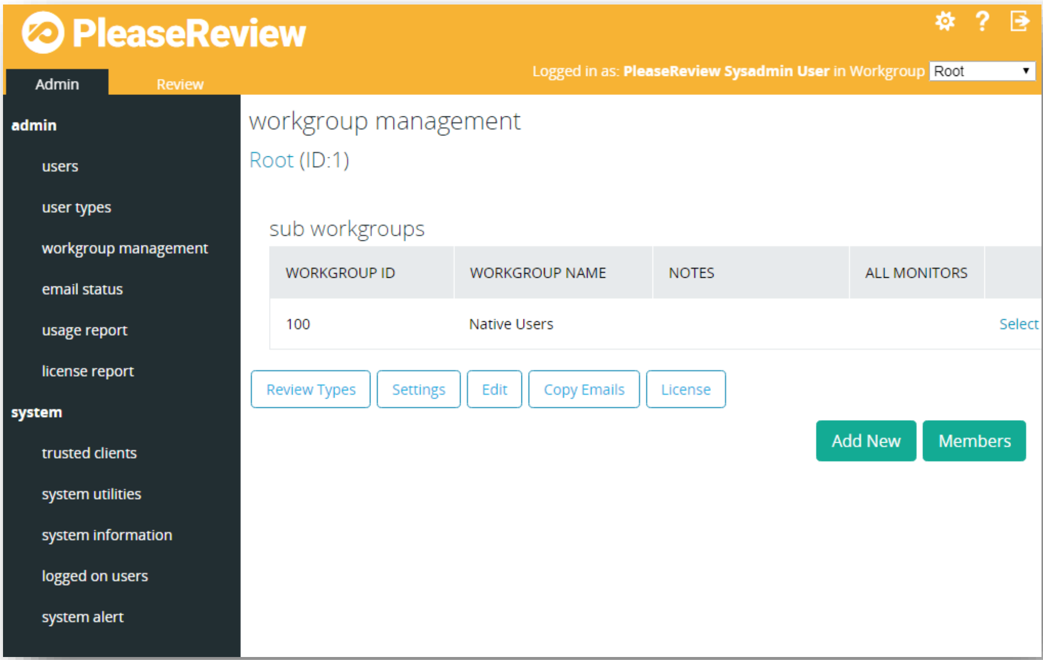
This can be done by following these steps;

Note: this example is for enabling the custom control for the root workgroup, which will cause them to be enabled on all workgroups unless overridden below.

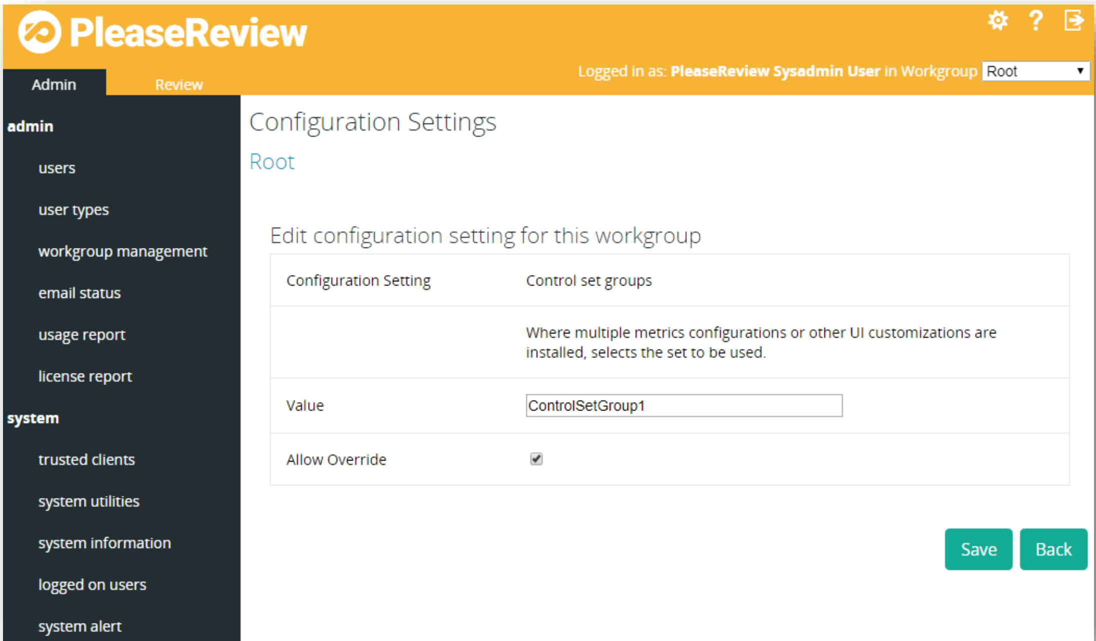
- Log in to PleaseReview as sysadmin
- Ensure the root workgroup in the drop-down on the top right is selected (or whichever workgroup you wish to apply to if at a lower level)



- Select the Admin tab
- You should see the 'workgroup management' screen. If not, select 'workgroup management' in the left-hand menu
- Click on 'settings'



- Locate the 'Review' section, and find the entry called "Control set groups"
- Click 'edit'
- Enter "CustomControlSetGroup" and click save



This has enabled the custom control on the root workgroup and, by inheritance, on all workgroups below it.

4.6 Extracting the data

To mine the data see the example SQL below:

```
select r.title review, c.objectId CommentId, c.ParagraphId, s.EntryName,
s.StringParam1
from T_Comment c,
T_Review r,
T_Settings s
where s.RefObjectType ='COMMENT'
and c.ObjectId = s.RefObjectId
and r.ObjectId = c.ReviewId
```

The actual values are stored in the T_Settings table, which holds miscellaneous data attached to various different things in the system.

The "`s.RefObjectType ='COMMENT'`" looks for settings attached to comments, and the "`c.ObjectId = s.RefObjectId`" is what joins the specific setting to the specific comment.

5 Redaction categorization configuration notes

5.1 Overview

The redaction categorization function is used to customize the appearance of redactions when downloading redacted PDF's from PleaseReview. The redaction categorization is based on the comment category applied to a redaction, so the appearance of redactions can differ based on the comment category associated with the redaction.

5.2 Default template

When your instance of PleaseReview is upgraded to PleaseReview 6.2 or later, the file RedactionCategory.xml will be in the PleaseReview directory 'Runtime/config'.

If the file doesn't exist, locate the RedactionCategory.xml.template file in in the PleaseReview directory 'Runtime/config', copy to the PleaseReview directory 'Runtime/config' and rename as RedactionCategory.xml.

The default template has two redaction categories set up, based on the redaction requirements of the European Medicines Agency policy 70 on the publication of clinical data for medicinal products for human use.

Those categories are 'EU70-PPD' and 'EU70-CCI'.

This default template applies the redaction appearance requirements as set out by the EMA in the requirements document published 19 December 2016, which can be found here: http://www.ema.europa.eu/docs/en_GB/document_library/Regulatory_and_procedural_guideline/2016/12/WC500218567.pdf

The RedactionCategory.xml file is as follows:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<!-- RedactedColor, OutlineColor, FillColor can have 'transparent' color -->
<Redactions version="1.0">
  <!-- <Blank> category can fill up as &lt;Blank&gt; -->
<Redaction category="EU70-PPD">
    <RedactedColor>#73CBEB</RedactedColor>
    <OverlayText>
      <!-- Supported fonts (Case-sensitive) :
          Times-Roman, Times-Italic, Times-Bold, Times-BoldItalic
          Helvetica, Helvetica-Oblique, Helvetica-Bold, Helvetica-BoldOblique
          Courier, Courier-Oblique, Courier-Bold, Courier-BoldOblique
          -->
      <Font>Helvetica</Font>
      <FontSize>12</FontSize>
      <FontColor>#000000</FontColor>
      <!-- Alignment: left, center, right -->
      <Alignment>left</Alignment>
      <Repeat>false</Repeat>
      <AutoSize>false</AutoSize>
      <CustomText>PPD</CustomText>
    </OverlayText>
    <Appearance>
      <OutlineColor>#000000</OutlineColor>
      <FillColor>transparent</FillColor>
      <!-- Opacity from 0 to 255 -->
      <Opacity>255</Opacity>
    </Appearance>
  </Redaction>
<Redaction category="EU70-CCI">
    <RedactedColor>#000000</RedactedColor>
    <OverlayText>
      <!-- Supported fonts (Case-sensitive) :
          Times-Roman, Times-Italic, Times-Bold, Times-BoldItalic
          Helvetica, Helvetica-Oblique, Helvetica-Bold, Helvetica-BoldOblique
          Courier, Courier-Oblique, Courier-Bold, Courier-BoldOblique
          -->
      <Font>Helvetica</Font>
      <FontSize>12</FontSize>
      <FontColor>#FF0000</FontColor>
      <!-- Alignment: left, center, right -->
      <Alignment>left</Alignment>
      <Repeat>false</Repeat>
      <AutoSize>false</AutoSize>
      <CustomText>CCI</CustomText>
    </OverlayText>
    <Appearance>
      <OutlineColor>#000000</OutlineColor>
      <FillColor>transparent</FillColor>
      <!-- Opacity from 0 to 255 -->
      <Opacity>255</Opacity>
    </Appearance>
  </Redaction>
</Redactions>
```


For the appearance of redactions to reflect these requirements, the comment categories applied to the redaction(s) in a review must match the redaction category in the above file, so in this instance 'EU70-PPD' or 'EU70-CCI'.

Amending the 'comment categories' for a review can be carried out at a system level, workgroup level or using review types. For information on how to do this, please refer to the PleaseReview System Administrator User Manual.

Any redactions which, do not have a matching redaction category in the RedactionCategory.xml file, will have the default redaction appearance.

5.3 Customizing the Redaction Categorization template

The redaction categorization file can be amended to meet organizations specific requirements.

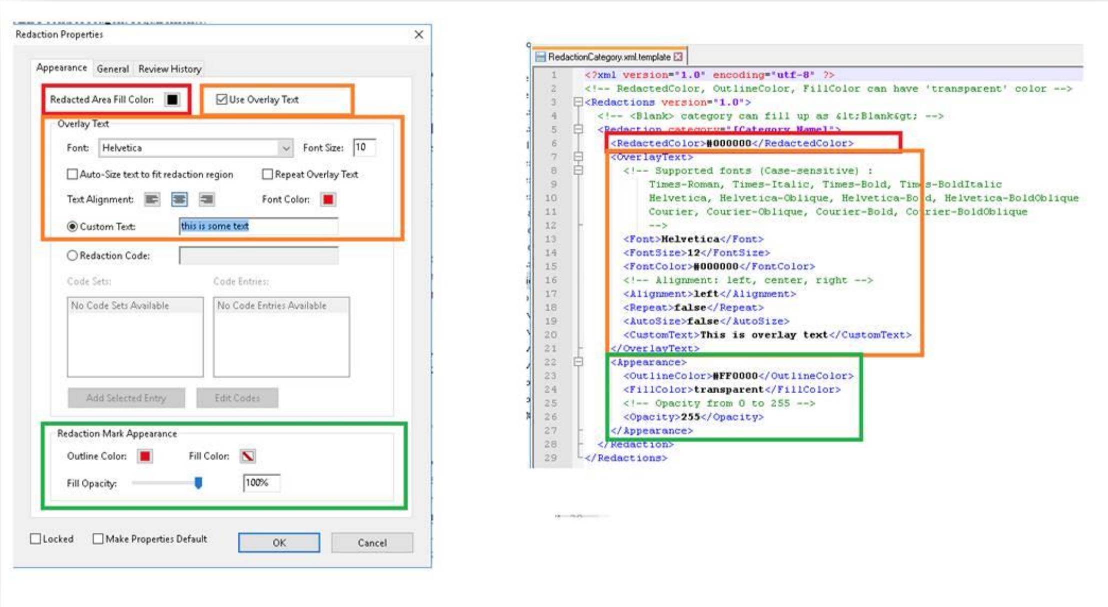
5.3.1 Editing the XML to meet your requirements

To amend the file, go to the PleaseReview directory 'Runtime/config' and locate the file RedactionCategory.xml

There are no limitations on how many redaction categories are set up. You can make amendments to the existing redaction categories or add additional redaction categories by copying the entire <Redaction category></Redaction> node and pasting it at the next sibling node.

Note redaction categories should be unique. If 2 matching redaction categories appeared in the configuration, the first match will be selected.

The configurations are divided into 3 element groups: Redacted Color, Appearance and OverlayText.



Parameter	Purpose	Example
Redaction category	The redaction category name should match the comment category that you want to associate this appearance with	E.g. "EU70-PPD" or "major"

Redacted color

Parameter	Purpose	Example
RedactedColor	The redaction color that should be applied to the redaction marks. Note all colors use the Hexadecimal RGB color format (the same as HTML)	E.g. #000000

Appearance

Parameter	Purpose	Example
Font	Supported fonts are: Times-Roman, Times-Italic, Times-Bold, Times-BoldItalic, Helvetica, Helvetica-Oblique, Helvetica-Bold, Helvetica-BoldOblique, Courier, Courier-Oblique, Courier-Bold, Courier-BoldOblique Note: if an unsupported font is entered, the default of Helvetica will be applied	E.g. Times-Roman
FontSize	The font size (in points) to be applied to the custom text on the redaction marks	E.g. 12
FontColor	The font color to be applied to the custom text on the redaction marks. Note all colors use the Hexadecimal RGB color format (the same as HTML)	E.g. #000000
Alignment	The text can be aligned left, right or center	E.g. left
Repeat	The determines whether any custom text is repeated	E.g. true or false
AutoSize	This determines whether the text is auto sized	E.g. true or false
CustomText	The overlaying text to be applied to the redaction.	E.g. PPD

Overlay Text

Parameter	Purpose	Example
OutlineColor	The outline color to be applied to the redaction marks. Note all colors use the Hexadecimal RGB color format (the same as HTML)	E.g. #000000
FillColor	The fill color to be applied to the redaction marks. Note all colors use the Hexadecimal RGB color format (the same as HTML)	E.g. #000000
Opacity	Opacity supported from 0 to 255	

Any changes in the configuration will take effect immediately.

If you wish to make changes to the template and you are a hosted customer, contact support.pleasereview@Ideagen.com for assistance.

Note: This feature does not apply to legacy PDF review using the PDF plugin or pre-6.0 review-as-image.

6 Archive

6.1 Overview

The archive feature allows PleaseReview to create and retain a complete 'data dump' of each review as it is deleted from the system. The end user has no visibility of, or control over, this process; it is enabled and controlled purely by settings on the server.

Archiving is an extra cost option and must be enabled in the license key.

Archives are written into a named file path on the server – one file per review. Substitution variables may be included into the directory and filenames, e.g. to use a different directory for each week or month.

PleaseReview does not provide any functionality to process the archive files after they are produced, so you must ensure you have a retention mechanism in place. Specifically note that PleaseReview does not have any feature to re-import an archive into a live PleaseReview system.

Archive only happens when the review is physically removed from the system. If the recycle bin feature is enabled, this means it occurs only when the review is purged from the recycle bin.

The archiving is processing-intensive and can significantly slow down the process of deleting reviews. We recommend archiving is implemented in conjunction with auto-delete which means it will happen in the background.

The remainder of this section concentrates on:

- How to enable and control the archiving functionality
- What goes into the archive file

6.2 Enabling and configuring archiving

6.2.1 License key

Firstly, as noted above, you must have archiving enabled in your license key. If archiving is not enabled in the license key, you can still set up the configuration. However, no archiving will happen.

6.2.2 Configuration parameter

The "AllowArchive" system configuration parameter can be used to turn archiving on and off for a specific workgroup or review type.

Note that in a system with multiple license keys, there is an interaction between the license key and the AllowArchive configuration parameter which is not necessarily obvious: for a review to be archived, the review owner's license key must include archiving as well as having archive enabled for the review itself.

For instance, consider a scenario where the Legal group has one license for PleaseReview which has archiving enabled, and the Sales group has a separate license without archiving enabled. User John has his home workgroup set to Legal and user Andy has his home workgroup set to Sales.

Now let's say both John and Andy have Author permissions in the workgroup 'Customer X' and that workgroup has archiving enabled. Any reviews owned by John in that workgroup will be eligible for archiving but, because Andy's license does not include it, reviews owned by him will not be.

6.2.3 Archive directory and filename parameterization

The archive directory is specified by the value Tara.Review.Archive in the Config.xml file. Variables may be substituted into the archive directory configuration parameter using braces. The default value is:

{workgroupid}/{date:yyyyMM}/{ReviewId}/{ReviewTitle}.zip

A relative path will be assumed to be relative to the PleaseReview "runtime/data/archive" directory. For clustered systems you would normally set the archive path to be a shared location (e.g. SAN), otherwise each review will appear in the archive location only for the server that initiated the archiving.

Substitution variables allowed are:

Parameter	Purpose
ReviewId	
ReviewTitle	
Date:xxx	Date and time on which the file was archived: where xxx is a standard .NET date format. See http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx for information on custom date/time format strings
WorkgroupId	
WorkgroupName	
OwnerId	User id (subject id) of the review owner
OwnerName	Full name of the review owner
OwnerEmail	Email address of the review owner

The names of the substitution variables are not case sensitive.

You can use the same substitution variable more than once – for instance use {date:yyyy}/{date:MM} to create a subfolder for each month under a folder for each year.

If the filename generated already exists, the name part will be suffixed with (1) (2) etc. It is strongly recommended that the filename contains the review id to avoid this kind of complication.

6.2.4 Archive sets and legal hold orders

6.2.4.1 Overview

As well as the standard archive directory configuration above, it is also possible to select specific reviews to be archived in a specific location based on keywords present in the review title or other metadata.

The inspiration for this feature is for a 'litigation hold order' which requires a company to refrain from deleting any electronic information pertaining to a specific topic, but it may be used for other purposes as well, hence the feature is known as 'archive sets'.

For each archive set, you may specify one or more phrases (typically a drug name, code word etc.) and a list of which places to search (typically review title, description, document title, metrics values). When a review is deleted, PleaseReview will identify whether that phrase is contained within any of those places and if it is, that review will be identified as being associated with that archive set and will be archived in the specific directory for that archive set.

Note that in the context of archive sets, there is no way of searching the entire text of the document for a keyword.

If the keyword matching does not meet your requirements, it is possible to implement a DLL in .Net which matches reviews to archive sets using whatever rules you like (for instance, if you are using PleaseReview in conjunction with a Document Management System, you may want to relate the review to certain keywords or folders in the DMS).

There may be more than one archive set in place at one time. To allow for easy deletion of documents associated with one specific archive set, there will be a sub-directory for each archive set and, in the event a document is associated with more than one archive set, a copy of the document will appear in all archive sets.

6.2.4.2 Configuration

An XML file does the configuration – archiveconfig.xml. If the configuration is changed, the system must be restarted to pick up the new configuration.

Note that archiving defined in the archiveconfig.xml is in addition to archiving configured by the configuration mechanism described above. Normally you would either use normal archiving or ‘archive set’ archiving but if they are both turned on, both would happen.

```
<archive>
   <archiveSet name="Case-1293">
       <archiveDir value="Data/archive/Case-1293” />
       <text match=”phenoldibutrazine” />
       <text match=”Dh3-c16-n1-2o3” />
       <text match=”spudilike” />
       <field select=”review.title” />
       <field select=”document.title” />
       <field select=”review.description” />
       <field select=“metrics” />
   </archiveSet>
```

In this example, it looks for any one of 3 text values in any one of 3 fields or any metrics value.

6.2.4.3 Implementing a custom archive set filter DLL

A custom filter is configured as a separate archive set like this:

```
   <archiveSet name="Case-483">
       <archiveDir value=”Data/archive/Case-1293” />
        <filter class=”MyCorp.PleaseReview.HoldOrderFilter,MyCorp.dll”>
   </archiveSet>
</archive>
```

The filter class must implement the interface Pleasetech.Tara.BusinessServices.Interface.IArchiveSetFilter defined thus:

```
Interface IARchiveSetFilter
{
   void Init(XmlElement configElement, BSManager bsManager);
   bool IsReviewInArchiveSet(Review review);
}
```

The Init method is called once for each archive set and passes in the config element for that specific archive set. If multiple archive sets reference the same filter class, multiple of the filter class will be instantiated and the init method of each will only be called once.

Each time a review is considered a candidate for archiving, the IsReviewInArchiveSet method of each filter is called for that specific review, and each filter instance must return true if the review is included in that archive set.

6.3 Contents of the archive

The archive file for each review is a zip file and contains the following:

- Reconciliation report as PDF
- Original documents
- Meta information in an xml file including:

- Review attributes
- Participant details
- Metrics
- All comments

If the review contains Zones, the archive feature generates all the reports and includes all versions of the documents.

For phased reviews, each phase generates a separate archive. For sub-reviews these are treated the same as any other review.

All reports are in PDF (specifically PDF/A) in landscape format.

6.3.1 Naming convention within the archive

Below is the folder structure and files inside the archive zip file:

```
{archive.zip}
       |— /doc
              |— {all the original documents}
       |— /comment attachment
              |— /{document id}
                     |— /{para id}
                            |— {document}
       |— /editzone history
              |— {all versions of the document}
              |— {reconciliation reports for each version}
       |— metainfo.xml
       |— {review title}.pdf
```

Note that any characters that are not valid windows filenames (< > : " / \ | ? *) will be removed from filenames in the archive.

6.3.2 Contents of 'doc' folder

All the original documents are stored in this folder. Because it is possible to have multiple documents of the same name, the document file name is prefixed with the document id.

6.3.3 Contents of 'comment attachment' folder

Comment attachments will be stored inside its associated paragraph id folder, and then stored in the document id folder. Subfolders are used to denote document id and paragraph id, and the filename is the original filename prefixed with the document ID of the attachment.

Example:

/472/00005372/9234 – The quick brown fox.docx

If the generated attachment filename together with the archive path exceeds the windows built-in limit of 260 characters, the name of the file will be the document ID only instead of id with original filename. It is possible to refer to the metainfo.xml file to locate the original file name if necessary.

6.3.4 Contents of 'editzone history' folder

All version of Zone history and recon report (including the original version) will be stored in this folder.

Example:

- Document Id - Sample Document – recon ver1.pdf

- Document Id - Sample Document – recon ver2.pdf
- Document Id - Sample Document – ver1.docx
- Document Id - Sample Document – ver2.docx
- Document Id - New Text Document – recon ver1.pdf
- Document Id - New Text Document – ver1.txt

6.3.5 Meta xml file contents

The Meta xml filename is “metainfo.xml”. It contains attributes of all objects in a ‘raw dump’ format. The exact schema for the file is not described here as it should be self-explanatory.

All Dates are in an ISO 8601 compliant date format with no time zone information - e.g. "2013-08-12T15:28:09".

Note: if custom review metrics are present, they are stored in the XML file only when the ‘simple’ metrics mechanism is used (i.e. where PleaseReview is responsible for persisting the metrics values into the T_Settings table). If the ‘custom’ metrics mechanism is used (where the user-supplied code is responsible for the persistence), the metrics values are not present in the archive file.

Metrics are stored in an element like this:

<metric name="tower height" value="1034">

The ‘name’ attribute corresponds to the control’s ‘caption’ attribute in the ControlSetRepository.xml.

an ideagen product t : +44 1629 699 100 | e : info@ideagen.com | w : ideagen.com

7 Additional Review Roles

Each review role in PleaseReview is configurable and made up of several distinct permissions. Not every possible combination is included in the standard installation. Some possible additional roles are listed below:

- **Reply Only** (Reviewer can reply but cannot make new comments or changes).
- **Comment Only** (Reviewer can make comments but not proposed changes).
- **Propose Change Only** (Reviewer can make proposed changes but not comments).
- **Close Only** (Author can accept or reject proposed changes but cannot make comments or propose changes).

To enable one or more of these roles, it is necessary to manually update the database using SQL. This should be done by someone who is confident with using the relevant tool to execute SQL statements.

Open the database creation script (runtime/config/createdb/createdb.xxx.txt) and search for "optional roles". You will see 8 insert statements all commented out. There are two insert statements for each of the above roles. To effect one or more of the roles, execute both the SQL statements pertaining to that role by pasting them into an SQL tool (for Oracle, make sure you commit after performing the inserts).

After you have done the updates, you will need to restart both the TaraWeb and TaraService processes for this to take effect. See Restarting Taraweb and Taraservice.

If you wish to create other roles, or to remove or change the privileges associated with existing roles, you should contact Ideagen support for assistance before proceeding.

an ideagen product
t : +44 1629 699 100 | e : info@ideagen.com | w : ideagen.com

8 Restarting Taraweb and Taraservice

8.1 Restart Taraweb

You can do this in several ways, either via the AdminClient (select Action -> Restart -> Taraweb) or via your IIS admin/manager (You should find PleaseReview under Application Pools, then right click PleaseReview and select recycle).

8.2 Restart Taraservice

You can do this in several ways, either via the AdminClient (select Action -> Restart -> Taraservice) or via the services control panel (You should find Taraservice, then stop and restart the service).

For more information on restarting the Taraweb or Taraservice go to the Installation and Administration Guide.

9 Notices

All trade names, trademarks, and service marks are the rightful property of their respective Owners.