Digital content is short-lived, yet will often be valued in the future. How can we keep it alive? Finding the right action to enable future access to our cultural heritage in a transparent way is the task of PLATO.

Planets Preservation Planning Tool: Plato 2.1

User Manual

V1.0

November 10, 2009

Table of Content

A. Introduction 6

Digital preservation 6

Preservation planning 7

About this document 8

What is new in Plato 2.1? 8

B. The workflow 10

C. About Plato (Planning tool) 12

Welcome page 12

The HOME screen: Entrance to Plato 13

Create a new Plan 14

Load Plan 16

Navigation 16

Plan setting 17

Export Plan 17

D. Workflow steps in Plato 18

1. Define Requirements 18

1.1 Define Basis 18

1.2 Define Sample Records 25

1.3 Identify Requirements 29

How to specify requirements in the objective tree editor 32

2. Evaluate Alternatives 39

2.1 Define Alternatives 40

2.2 Go/No-Go 42

2.3 Develop Experiments 43

2.4 Run Experiments 44

2.5 Evaluate Experiments 46

3. Analyse Results 49

3.1 Transform Measured Values 49

3.2 Set Importance Factors 53

3.3 Analyse evaluation results 54

4. Build Preservation Plan 55

4.1 Create Executable Plan 56

4.2 Define Preservation Plan 57
4.3 Validate Preservation Plan 58

List of Figures

Figure 1 Overview of PLANETS Preservation Planning workflow 10
Figure 2 Welcome page 12
Figure 3 The home screen 13
Figure 4 Possibilities to create a Plan 15
Figure 5 Create Preservation Plan 15
Figure 6 Plato navigation 16
Figure 7 First entry of the toolbar 16
Figure 8 Progress indicator 17
Figure 9 Buttons: Save, Discard changes, save and proceed 18
Figure 10 Define Basis 18
Figure 11 Identification 19
Figure 12 Status 20
Figure 13 Description 22
Figure 14 Upload Policy XML 24
Figure 15 Policies 25
Figure 16 Define Sample Records 25
Figure 17 Collection Profile 26
Figure 18 Sample Records 27
Figure 19 Button Show the Fragments & Templates 36
Figure 20 Descriptive Information 39
Figure 21 Define the alternatives to consider for the plan 40
Figure 22 CRIB Service Registry: list of potential alternatives 41
Figure 23 Add new Alternative 41

Figure 24 Take the Go decision 42
Figure 25 Develop Experiments 43
Figure 26 Run Experiments 44
Figure 27 Result Files 45
Figure 28 Evaluate Experiment 46
Figure 29 Requirements to evaluate 47
Figure 30 Evaluation of requirements 47
Figure 31 Transform Measured Values 49
Figure 32 Transformation tables for different requirements 50
Figure 33 Transformation tables for ordinal scale 50
Figure 34 Transformation table for numeric scale 51
Figure 35 Interpolation of measured values 51
Figure 36 Aggregation mode for transformation 51
Figure 37 Transformation of numeric values 53
Figure 38 Set Importance Factors 53
Figure 39 Set Importance Factors 54
Figure 40 Analyse evaluation results for Preservation Plan for Papers 55
Figure 41 Details for Changelogs 55
Figure 42 Create Executable Plan 56
Figure 43 Define preservation plan 57
Figure 44 Validate plan 58

List of Tables

Table 1 Information which is on the welcome page 13

Table 2 Example for the boxes in Create Preservation Plan 16

Table 3 Examples for boxes for Identifikation 20

Table 4 Examples for the boxes in Status 21

Table 5 Examples for the boxes for Description 23

Table 6 Examples for the boxes in Collection Profile 27

Table 7 Examples for the boxes in Sample Records 29

Table 8 Examples for the boxes in Descriptive Information 39

Table 9 Examples for the boxes in Take the Go decision 42

Table 10 Examples for Develop Experiments 44

Table 11 Examples for the boxes in Run Experiments 45

Table 12 Examples for the boxes in Create Executable Plan / When 57

Table 13 Examples for the boxes in Create Executable Plan / What 57

A. Introduction

Digital preservation

The fast changes of technologies in today's information landscape have considerably shortened the lifespan of digital objects. While analogue objects such as photographs or books directly represent the content, digital objects are useless without the technical environment they have been designed for. In contrast to a book, Word documents cannot be read, a simulation cannot be re-run and re-evaluated, sensor data cannot be interpreted without a suitable hardware, software and documentation environment. Digital objects are under threat at several levels: media failure, file format and tool obsolescence, or the loss of necessary metadata. Especially for born-digital material this often means that the contained information is lost completely. Digital preservation has become a pressing challenge for any kind of IT-related operation.

Given that a digital object needs an environment to function, we can either recreate the original environment (emulation) or transform the object to work in different environments (migration). A growing number of tools performing migration and emulation are available today; each tool has particular strengths and weaknesses, and most often, there is no optimal solution. On the other hand, requirements vary across institutions and domains, and for each setting, very specific constraints apply that need to be considered. The process of evaluating potential solutions against specific requirements and building a plan for preserving a given set of objects is called preservation planning. Preservation planning is the centerpiece of the Reference Model for an Open Archival Information System (OAIS, ISO Standard 14721:2003)[1]. So far, it is a mainly manual, sometimes ad-hoc process with little or no tool support.

The planning tool **Plato**[2], developed as part of the Planets project[3] by the Digital Preservation lab at the Vienna University of Technology[4], is a publicly available web-based decision support tool accessing a distributed architecture of preservation services. It implements a solid planning process and integrates a controlled environment for experimentation and automated measurements of outcomes. This enables trustworthy, evidence-based decision making as required by the Trustworthy Repositories Audit & Certification Criteria[5] (TRAC, currently under evaluation for ISO standardization).

[1] ISO-14721:2003: OAIS, Blue-Book: http://public.ccsds.org/publications/archive/650x0b1.pdf

[2] Plato Project Homepage: http://www.ifs.tuwien.ac.at/dp/plato

[3] Planets Project Homepage: http://www.planets-project.eu

[4] Digital Preservation Lab, Department of Software Technology and Interactive Systems, Vienna University of Technology: www.ifs.tuwien.ac.at/dp

[5] Trusted Repositories Audit and Certification Checklist: http://www.crl.edu/sites/default/files/attachments/pages/trac_0.pdf

Preservation planning

To ensure digital content stays accessible and authentic for future users a plan has to be created taking into account legal and technical constraints such as storage space, infrastructure and delivery, copyright issues, and costs, user needs, and object characteristics.

*A preservation plan defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection). The Preservation Plan takes into account the preservation policies, legal obligations, organisational and technical constraints, user requirements and preservation goals and describes the preservation context, the evaluated preservation strategies and the resulting decision for one strategy, including the reasoning for the decision. It also specifies a series of steps or actions (called **preservation action plan**) along with responsibilities and rules and conditions for execution on the collection. Provided that the actions and their deployment as well as the technical environment allow it, this action plan is an executable workflow definition.*[6]

The four-phase high-level workflow shown below can further be divided into 14 steps. Evaluation of candidate actions uses controlled experiments and increasingly automated measurements.

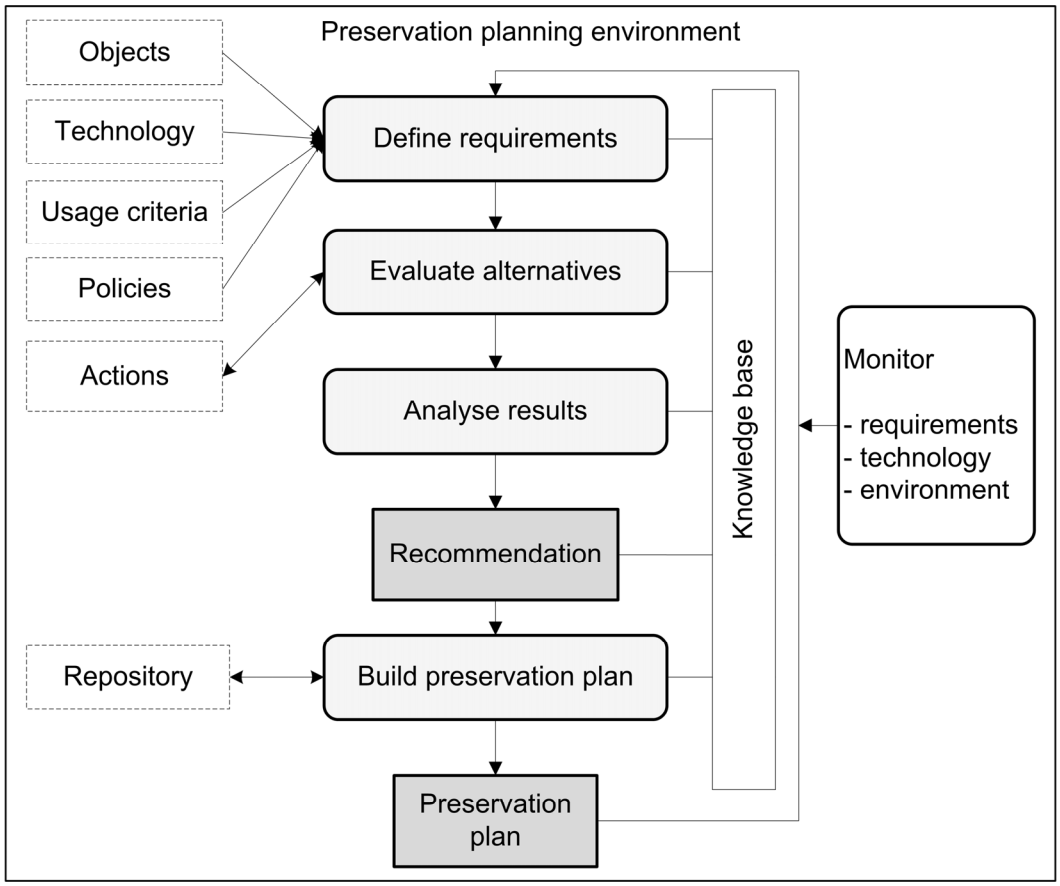


Figure 1 Preservation planning environment

[6] Full definition available at http://www.ifs.tuwien.ac.at/dp/plato/docs/plan-template.pdf

Potential migration and emulation tools are applied to sample content and evaluated according to a hierarchy of requirements, based on Utility Analysis. A service-oriented framework greatly automates experiments and allows users to leverage various publicly available web service registries that provide access to potential preservation action tools. Quality-aware services measure execution parameters and quality of the action tools and take this burden off the experimenter.

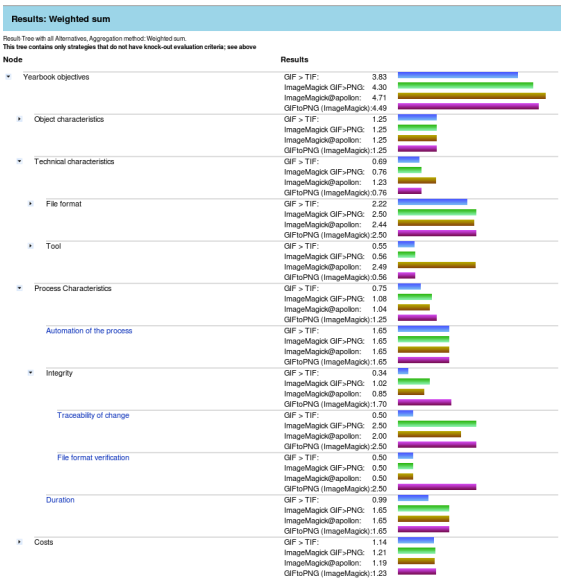


Figure 2 Visualisation of results

The result of using Plato is a complete preservation plan that can be deployed and executed.

About this document

This document describes version 2.1 of the Planets Preservation Planning Tool Plato and how to use it. Plato 2.1 implements the Planets Preservation Planning approach which is described in more detail in the next chapter.

A deployed version of Plato can be found at http://www.ifs.tuwien.ac.at/dp/plato .

***Please note**: The help pages of Plato are integrated in the tool and are kept up to date with every release. While we try to keep this manual updated as best as possible, the primary source of reference is constituted by the help pages.*

What is new in Plato 2.1?

The main new features of Plato 2.1:

- **Sensitivity analysis.** We have implemented sensitivity analysis to visualise criteria that are sensitive to variations in the weightings.
- **Objective tree editors and knowledge base.** We have redesigned the user interface for the objective trees and come up with a much easier to use and much faster editor for the knowledge base.

- **Jhove.** We have integrated JHove, including a neat visual side-by-side comparison feature for migrated sample objects to support visual evaluation.
- **Quality-aware migration services.** We have made a prototype registry containing quality-aware migratino services available through Plato, featuring automated evaluation of some of the requirements. These migration service measure quality and performance (time and memory) and provide this information together with the result. Corresponding publications about this technology can be found on the documentation page.Navigation structure. We have introduced a home screen providing a central point of entry.
- **Executable preservation plan.** Plato 2.1 creates an executable preservation plan in XML, which can be run in the Planets workflow execution engine.
- **Service Integration.** Updated access to Planets migration services.
- **Scalability.** Previously, it was not feasible to upload large sample objects to create a preservation plan, due to memory limitations. We have worked on this issue and are now supporting sample objects sets up to (roughly) 200MB per plan.
- **Policy definition.** You can now define your policies once and each preservation plan you create will be using these policies.

Please note: Plato is still ongoing development and will continuously be improved and enhanced. In most cases an update of Plato also comes with a database update which might reset the database's content. However, Plato also supports import and export of all preservation plans developed in Plato. A Preservation plan can be exported as an XML file and downloaded as such, and uploaded to a new version of the planning tool. Preservation plans are automatically upgraded to the newest version with every release.

Working on a preservation plan can seamlessly be continued after importing a plan.

B. The workflow

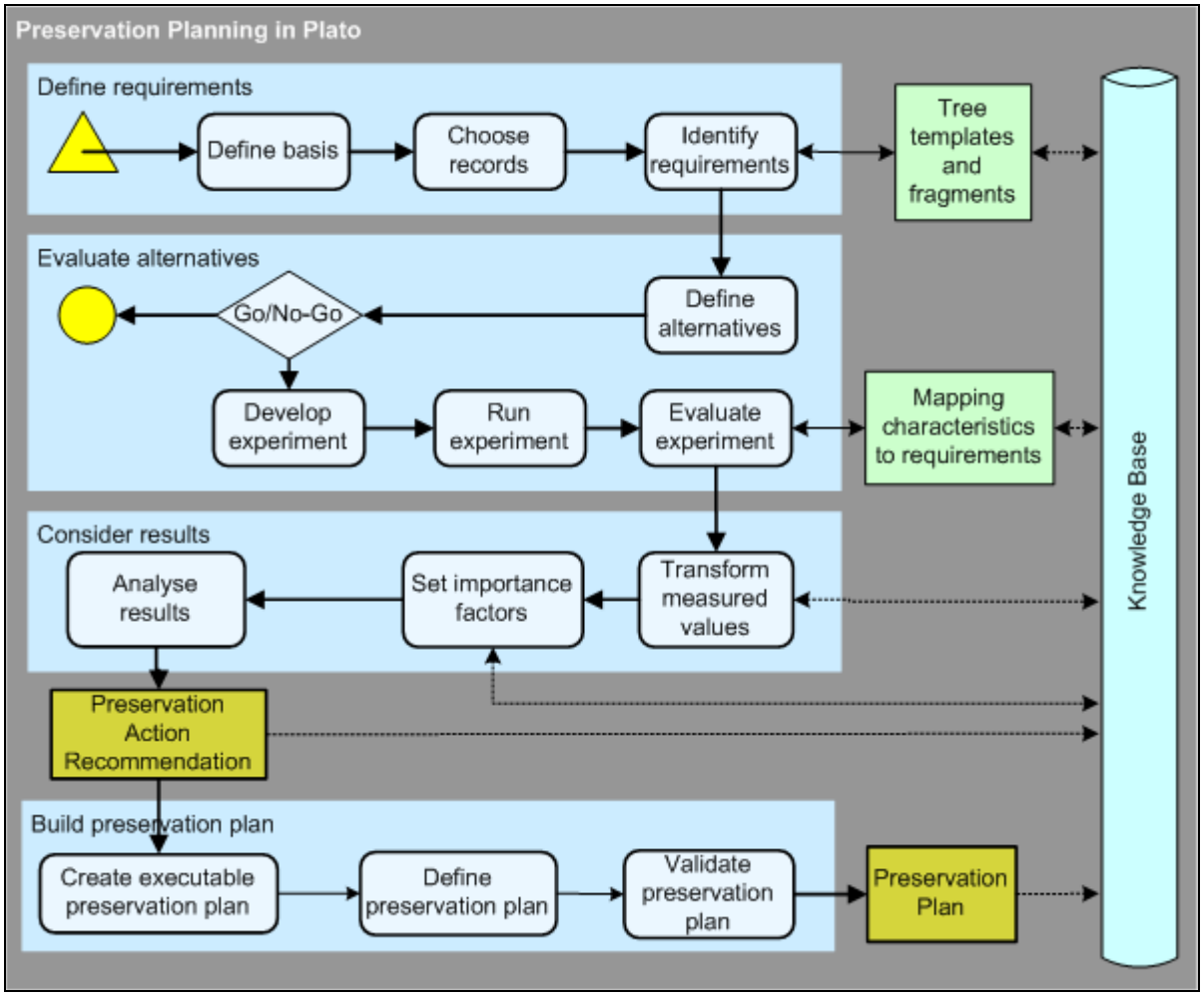
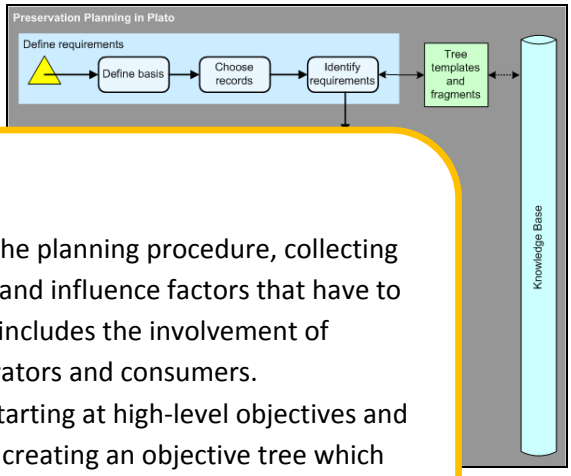


Figure 1 Overview of PLANETS Preservation Planning workflow

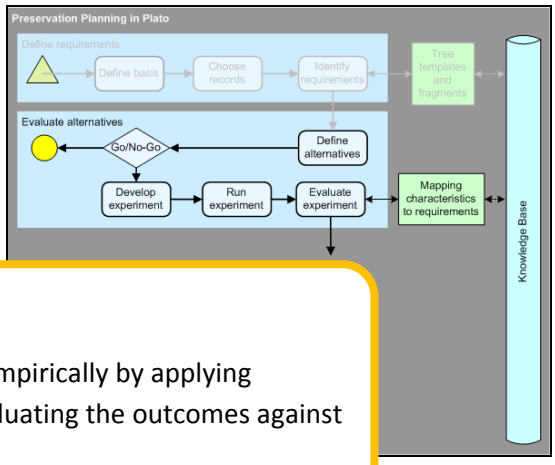
The Planets preservation planning workflow as described in Figure 1 consists of four main stages:

1. Define requirements
2. Evaluate alternatives
3. Consider results
4. Build preservation plan



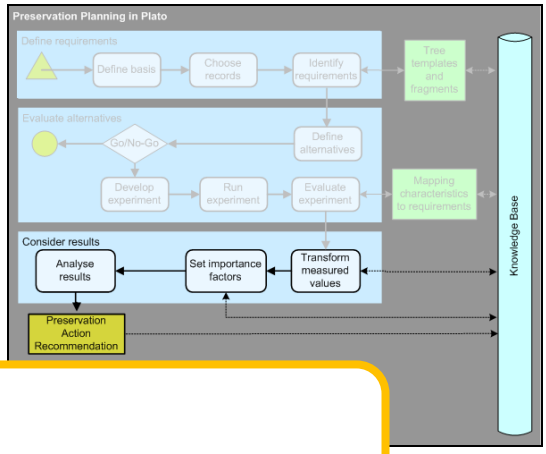
1. Define requirements

Requirements definition is the natural first step in the planning procedure, collecting requirements from the wide range of stakeholders and influence factors that have to be considered for a given institutional setting. This includes the involvement of curators and domain experts as well as IT administrators and consumers. Requirements are specified in a quantifiable way, starting at high-level objectives and breaking them down into measurable criteria, thus creating an objective tree which forms the basis of the evaluation of alternative strategies. Furthermore, as this evaluation would be infeasible on the potentially very large collection of objects, the planner selects representative sample objects that should cover the range of essential characteristics present in the collection at hand.



2. Evaluate alternatives

The evaluation of potential strategies is carried out empirically by applying selected tools to the defined sample content and evaluating the outcomes against the specified requirements.

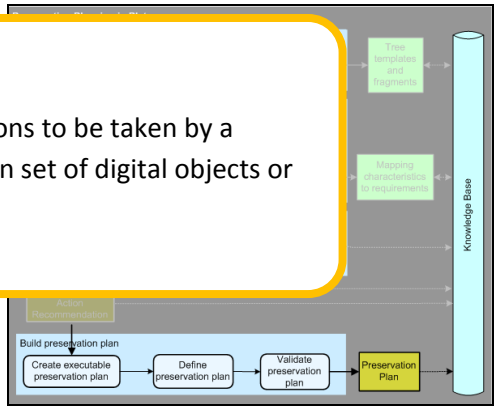


3. Consider results

Analysis of the results takes into account the different weighting of requirements and allows the planner to arrive at a well-informed recommendation for a solution to adopt.

4. Build preservation plan

A **preservation plan** defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection).



C. About Plato (Planning tool)

Welcome page

Five sections for introducing Plato

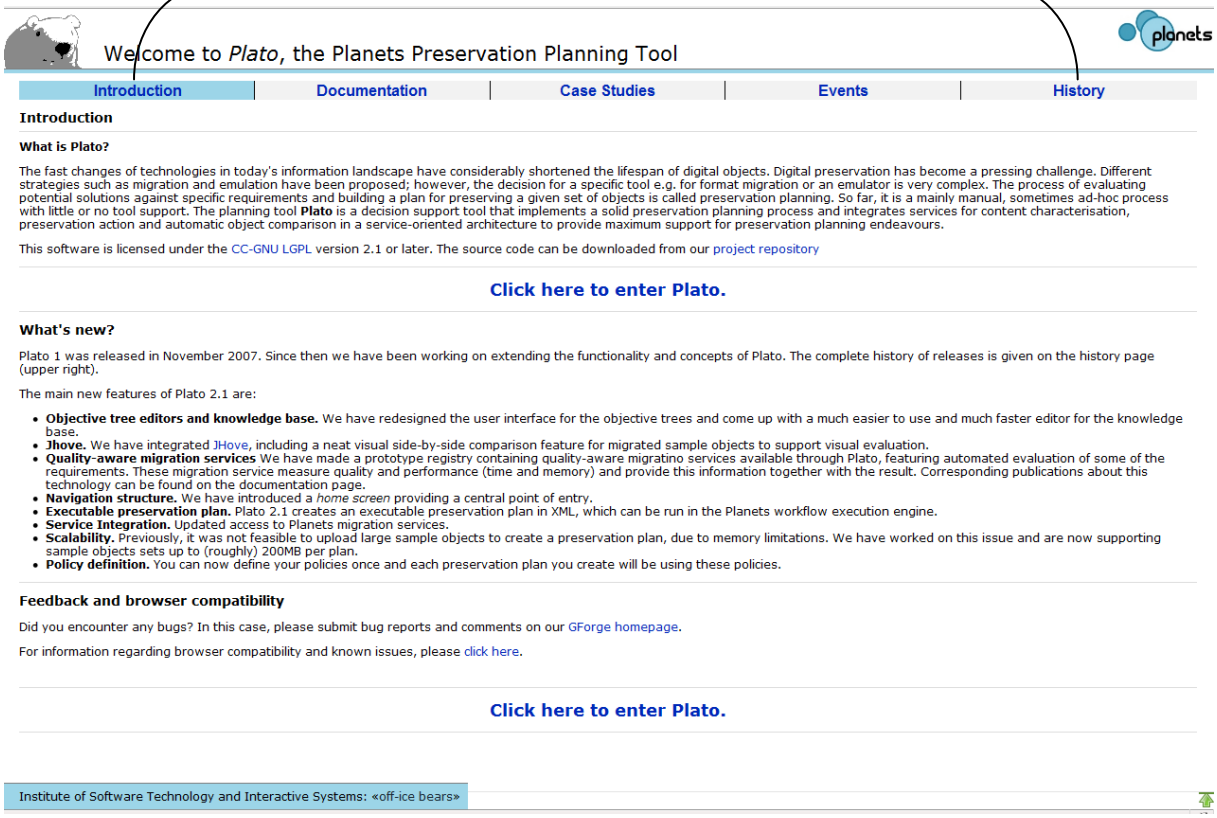


Figure 2 Welcome page

The first page (Figure 2) of Plato is a welcome page presenting detailed information about Plato. The information is subdivided into 5 sections:

Introduction	Documentation	Case Studies	Events	History
• What is Plato • News • What is new? • Feedback and browser compatibility	• Introductory material • Scientific Papers • Plan Deliverables	• Video Games • Interactive Multimedia Art • Electronic Theses and Dissertations • Scanned books • Scanned newspapers • Scanned yearbooks • Bitstream Preservation of Digital Photographs	• Upcoming Events • Past Events	• Release history and changes • Awards

Table 1 Information which is on the welcome page

To enter Plato you have to click the link: 'Click here to enter Plato'. In the login screen, you have to enter your credentials. If you don't have an account yet, you can create it right there.

The HOME screen: Entrance to Plato

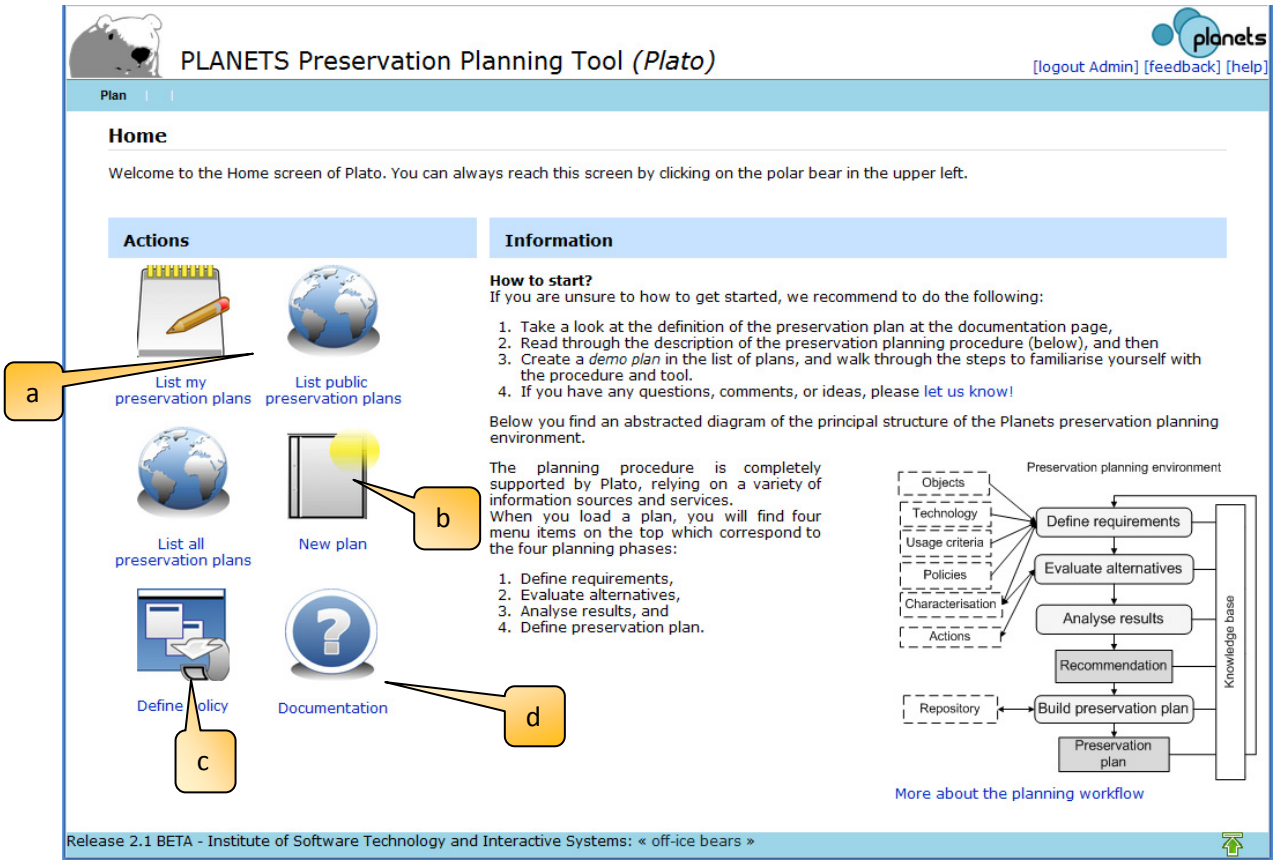


Figure 3 The home screen

The home screen provides a very short introduction text with further links and, depending on your role, allows you to access several pages:

a) Lists of your personal plans, published plans, or (in case you are administrator) all plans
b) Create a new plan
c) Define your organisation's policy model
d) Access the documentation pages

If you choose one of the plan lists, you arrive at a screen like the following:

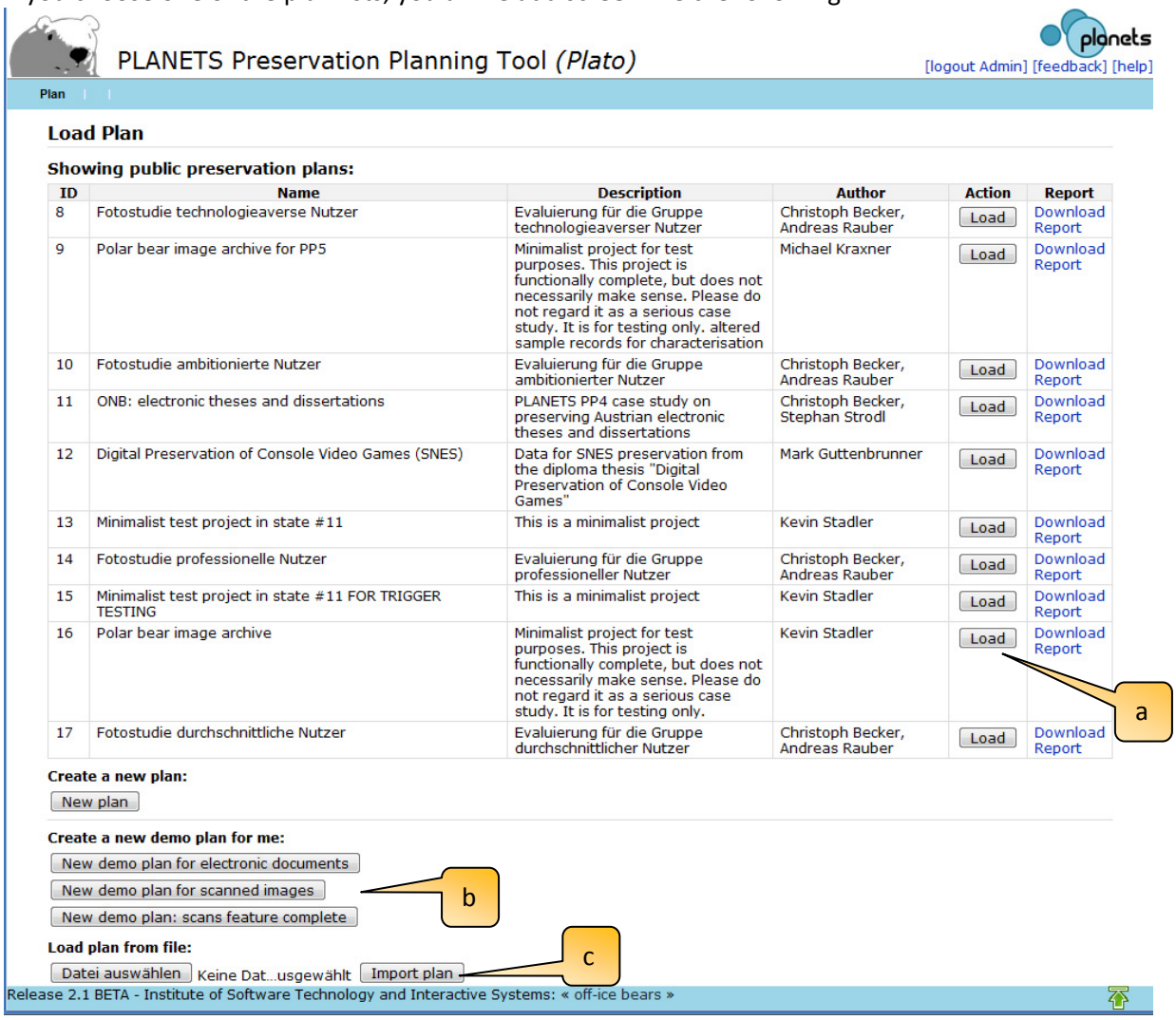


Figure 1 Load a plan

Here you can

- load an existing plan, [see a]
- create a new DEMO plan, [see b]
- or load plans from previously exported plans in XML [see c]

Create a new Plan

A new Plan can be started from scratch by either choosing ‘New Plan’ from the menu or clicking the ‘New Plan’ button on the ‘Load plan’ screen.

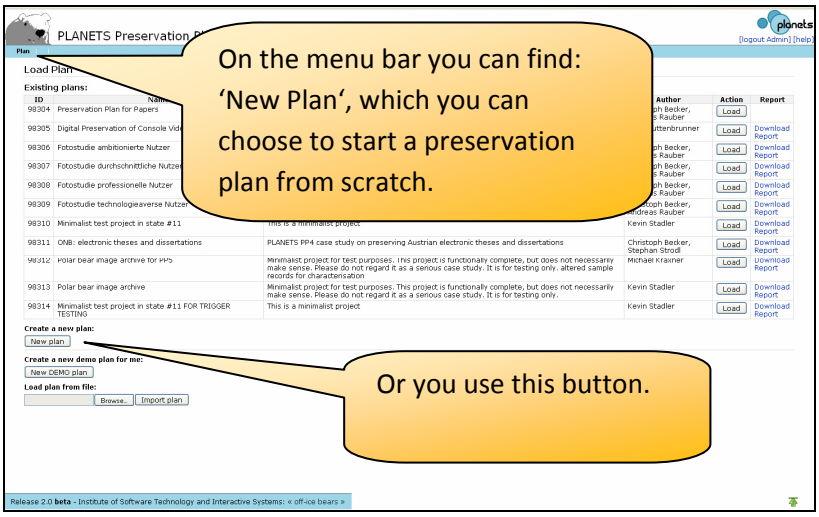


Figure 4 Possibilities to create a Plan

After you have clicked the button for a new plan a new site will open.

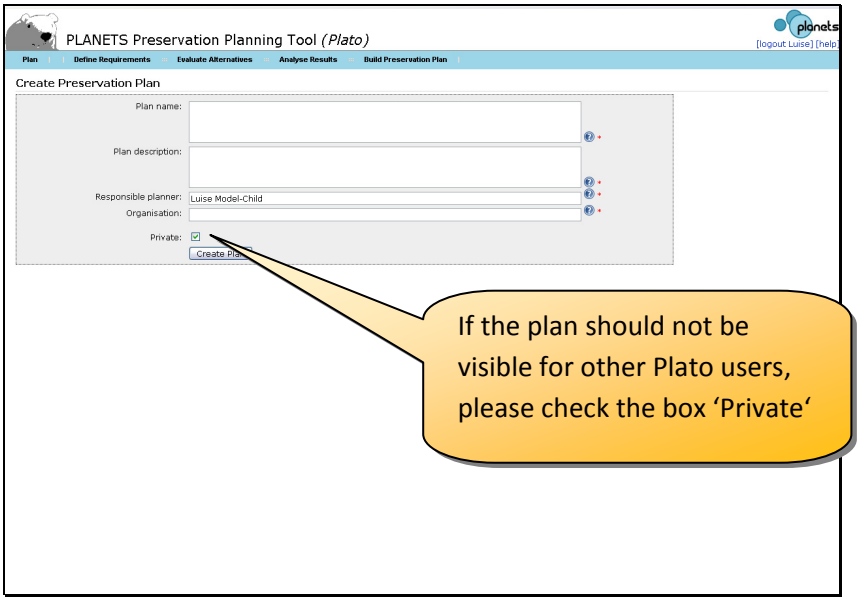


Figure 5 Create Preservation Plan

Name of the text fields	Examples which could be filled in the text fields
Plan name	• Preservation Plan for Papers • Digital Preservation of Console Video Games (SNES)
Plan description	• This is an example plan. The plan was created for the DELOS Summer School 2008 and revised afterwards. • Data for SNES preservation from the diploma thesis "Digital Preservation of Console Video

	Games"
Responsible planner	• Hans Hofman
Organisation	• National Library • Vienna University of Technology

Table 2 Example for the boxes in Create Preservation Plan

After filling out the boxes continue by clicking the button: Create Plan

Load Plan

After logging into Plato, a list of stored plans is provided including the ID of Plan's name, a description and the author(s) name(s) (see **Error! Reference source not found.**). The plan can be loaded with the 'Load' button on the right side. When loading an existing plan, the user will be forwarded to the workflow step corresponding to the plan's state (see Figure 1).

Navigation

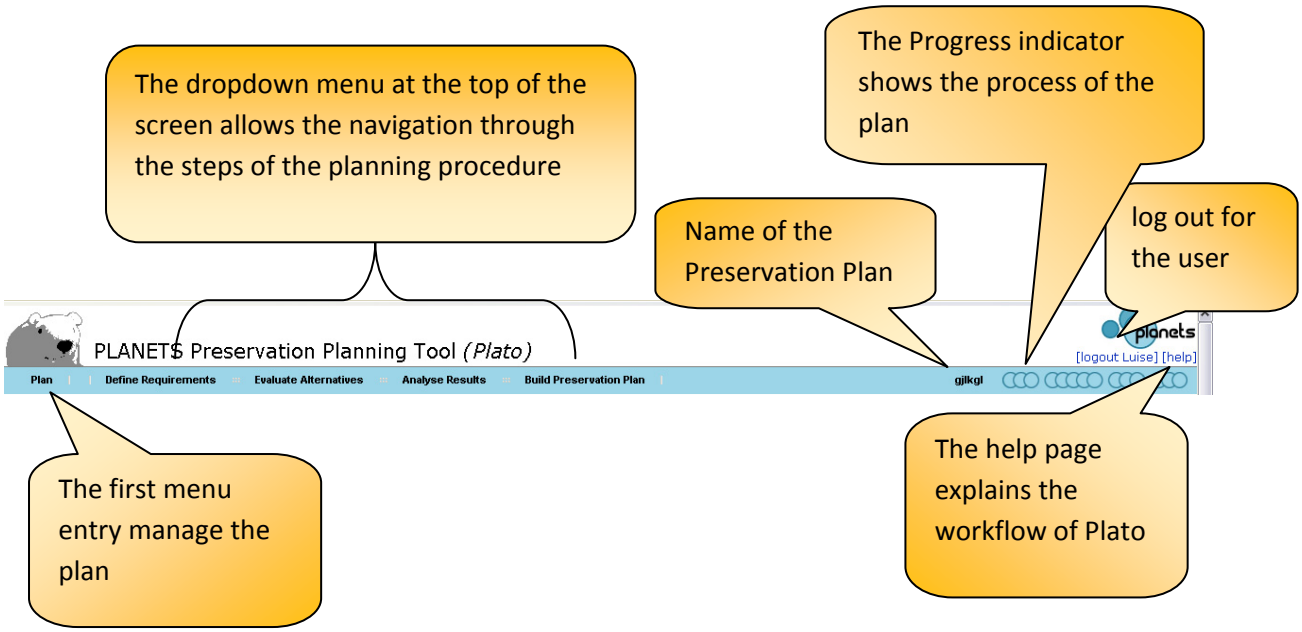


Figure 6 Plato navigation

The dropdown menu at the top of the screen allows in the first entry to manage the plan and in the following entries the navigation through the steps of the planning procedure (see Figure 7).

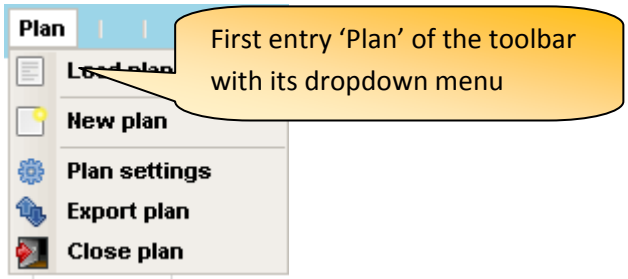


Figure 7 First entry of the toolbar

The first entry of the toolbar manages the plans, it includes:

Plan setting
It provides basic security options that you may use to protect, unprotect and delete plans that are created. There is also the option to upload the final report so that other users are able to see it at the Site: “Load page” (see Figure 3)
These options are only available to the user who created this plan.

Export Plan (Only available if a plan is opened.)
Exports the plan as an XML file.

Close Plan (Only available if a plan is opened.)
Closes the current plan and redirects to the Create New Plan/Load Plan page

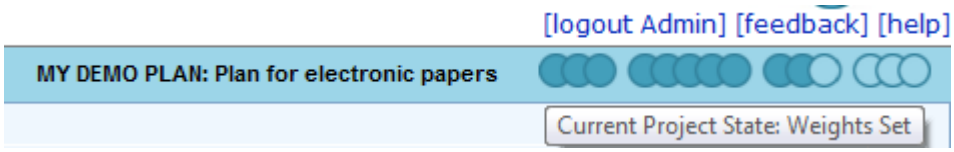


Figure 8 Progress indicator

On the right side of the Plato navigation bar the name of the plan is shown (see Figure 6). A Progress indicator shows the process of the plan (see Figure 8Figure 8). The progress indicator shows the plan’s state, how far the plan has progressed. It shows which steps in the workflow have already been completed (dark circles). Moving the mouse over the indicator shows the name of the currently reached state as shown in the figure above.
Above the progress bar are the log out as well as help functions, and a link to a feedback form.

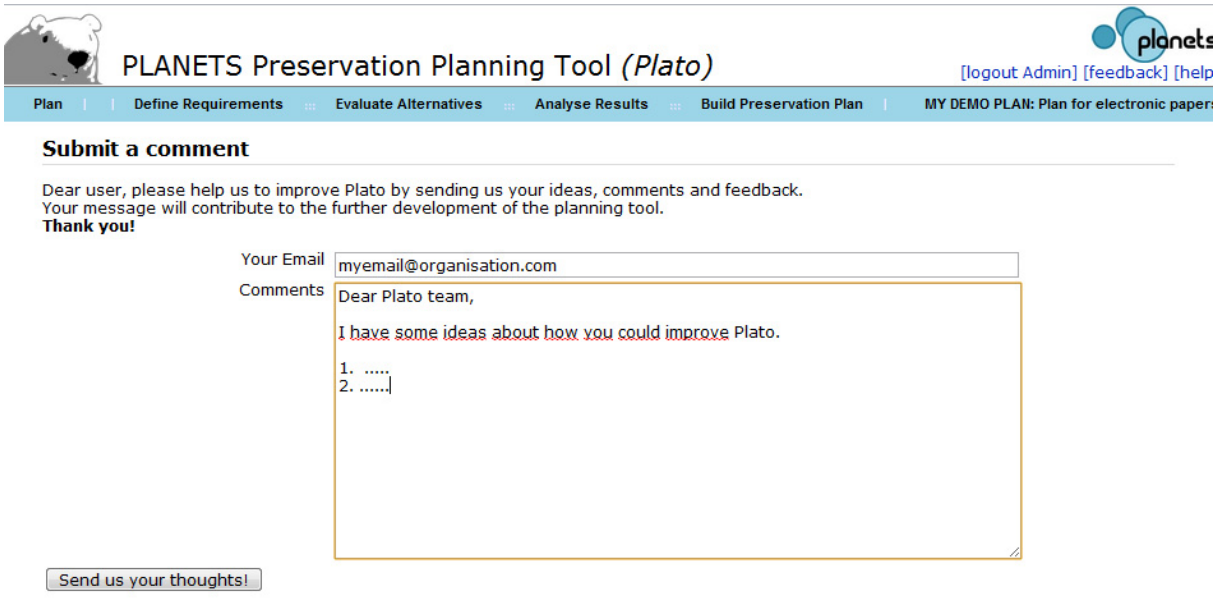


Figure 2 Feedback form

We very much value your feedback, so please take the opportunity to let us know your thoughts!

Figure 9 Buttons: Save, Discard changes, save and proceed

Each page of the workflow contains the buttons ‘Save’, ‘Discard changes’ and ‘Save and proceed’ (see Figure 9). The save button stores the current plan in the database. The ‘discard changes’ button restores the plan state of the last saving.

It is possible to easily navigate within workflow steps that are completed, i.e. have been passed by clicking ‘Save and proceed’. The plan’s state will be reset if you click either ‘Save’ or ‘Save and proceed’.

The next step of the workflow can be reached through ‘Save and proceed’. Plato validates the input and proceeds to the next workflow step if valid. Any validation errors that occurred will be indicated and can be corrected.

D. Workflow steps in Plato

1. Define Requirements

The first phase of the workflow lays out the cornerstones of the planning endeavour. It starts with collecting and documenting the influence factors and constraints on possible actions and procedures, then describes the set of objects under consideration and finally defines the complete set of requirements to be taken into account.

1.1 Define Basis

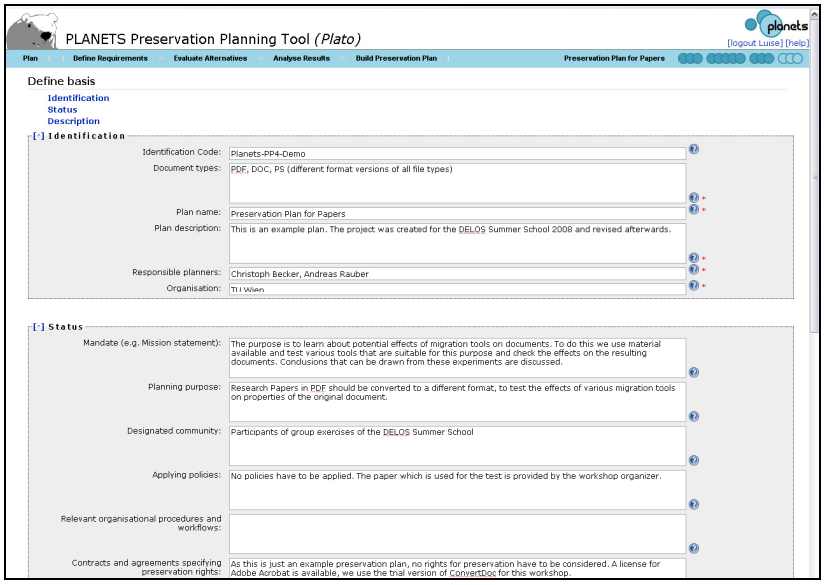


Figure 10 Define Basis

In this step, the preservation planner documents institutional policies, legal regulations, and usage criteria that might affect planning decisions for preservation. This may happen in an unstructured form, but preferably these factors are captured in a more formal way making it easier to derive decisions in the respective workflow steps. Examples include policies defining permitted file formats

for ingest, and policies related to intellectual property right and legal access regulations. Further important policy elements pertain to characteristics of the preservation action, whether preservation actions that are open source shall be preferred or if just a specific preservation strategy may be applied, such as emulation. This might be possible in cases where the institution doesn't have the copyright and thus any modifications of the digital object are prohibited. Furthermore, the event that led to the planning procedure is documented.

1.1.1 Identification

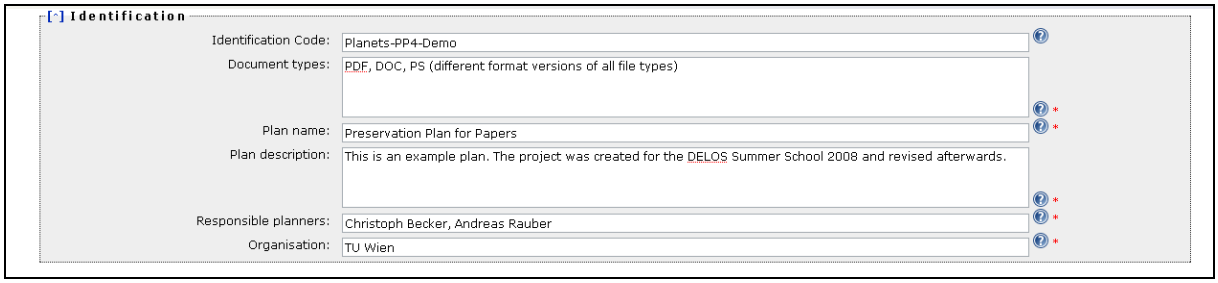


Figure 11 Identification

A preservation plan should be uniquely identified so that it can easily be referred to and retrieved.

Type	Examples
Identification code:	• Planets-PP4-Demo • ARCH-COLL-1299
Document types	• PDF, DOC, PS, (different format versions of all file types) • Digital Data from Cartridges of Super Nintendo Entertainment System (SNES) video games (Binary Streams) • The material exclusively consists of GIF masters, each a scanned page or part of a yearbook.
Plan name	• Preservation Plan for Papers • Digital Preservation of Console Video Games (SNES)
Plan description	• The preservation plan looks at the yearbook material that has recently (end 2008) been transferred from LTO tape to HDD storage on the Digital Preservation Teams SAN storage unit for content stabilization. We want to evaluate the file format for the master images and ensure that it is suitable for long term preservation. • Data for SNES preservation from the diploma thesis "Digital Preservation of Console Video Games"

Responsible planners	• Christoph Becker • Mark Guttenbrunner
Organisation	• Vienna University of Technology • The State and University Library

Table 3 Examples for boxes for Identifikation

1.1.2 Status

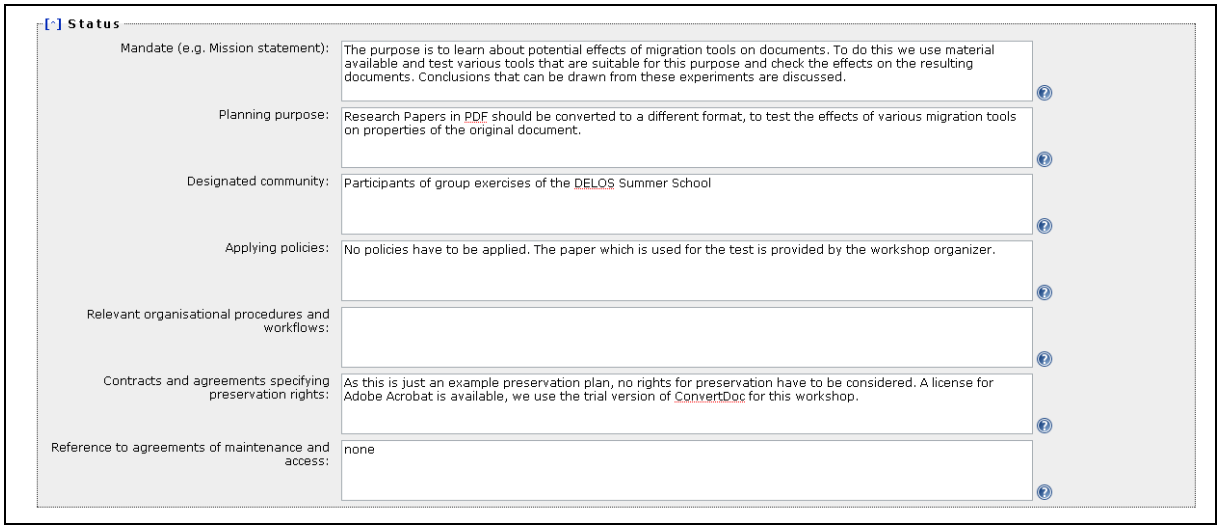


Figure 12 Status

The status of a plan includes both the planning progress – whether a plan is currently being defined, awaiting approval, or already has been deployed and is active – and the triggers which have led to the definition or refinement of the plan.

Type	Examples
Mandate (e.g. Mission statement)	• The mission of The State and University Library is to preserve our states cultural heritage in the form of any publication, whether it is printed or recorded, for the future.
Planning purpose	• The purpose of this plan is to find a strategy on how to preserve this collection for the future, i.e. choose a tool to handle our collection with. The tool must be compatible with our existing hardware and software infrastructure, to install it within our server and network environment. The files haven't been touched for several years now and no detailed description exists. However, The State and University Library has to ensure its accessibility for the next years.

	• The library has the legal obligation to preserve every published console video game like national libraries are obliged to preserve publications on paper and offer possibilities to display these games to the public.
Designated community	• General public • The target audience is visitors of the library. It is not necessary to publish the collection online. Access to games from the library collection to experience the games original look & feel should be possible for the public. Access to original media shall not be necessary to avoid damage to rare specimen.
Applying policies	• No policies have to be applied. The paper which is used for the test is provided by the workshop organizer. • For legal reasons only games physically in the possession of the library are preserved. This is because of copyright restrictions. The archive doesn't hold the copyright for the rest of the collection. • See policy model
Relevant organisational procedures and workflows	• Library account is needed for access. • Plan has to be presented and approved by the vice board
Contracts and agreements specifying preservation rights	• The institution has the necessary right to change the object, which might be necessary for migration. • Copyright held for the physical material. Legal mandate implies that transforming logical representation of the content is allowed.
Designated community	• Participants of group exercises of the DELOS Summer School • The target audience is visitors of the library. It is not necessary to publish the collection online. Access to games from the library collection to experience the games original look & feel should be possible for the public. Access to original media shall not be necessary to avoid damage to rare specimen.

Table 4 Examples for the boxes in Status

1.1.3 Description

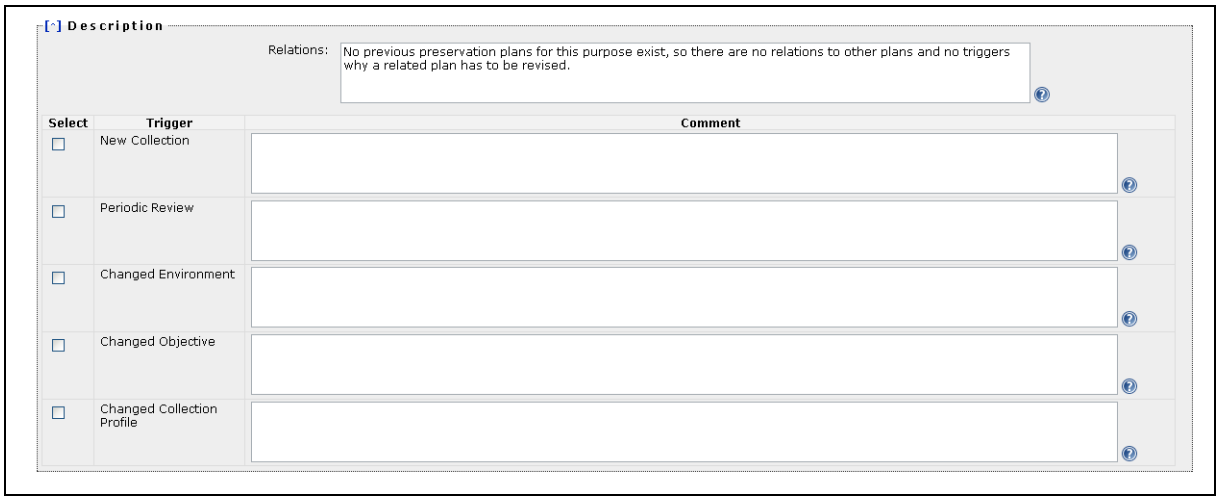


Figure 13 Description

New Collection

This is the most common event, where a preservation plan is created from scratch for a new collection, for which no plan was previously defined.

Changed Collection Profile

Changes in the collection profile of an existing collection may require a revision of existing preservation plan. Examples for changes in the collection profile are newly accepted object formats or significant changes in collection size. It is the responsibility of technology watch functions to ensure that these triggers are actually fired; the corresponding events should then be recorded in the planning documentation.

Changed Environment

The environment of a preservation plan consists of the technical environment, the designated communities and the host institution. Changes in the environment can lead to a change in preferences, for example with respect to the system context in which a preservation action needs to operate. They might also imply a change in factors which influence existing preservation plans, for example changed prices for hardware or software. Other changes are the availability of new preservation strategies or obsolescence of object formats which are used in an existing preservation plan. Changes in the environment require a revision of existing preservation plans, while the objectives for the evaluation usually will remain unchanged.

Changed Objective

Changes and developments in the environment can change the objectives for preservation evaluation over time. In this case it is necessary to evaluate existing preservation plans against changed objectives. Examples for these changes are technology developments or changes in high-level policies or legal obligations that have an impact on preferences and objectives. Changes in the designated community may also effect the goals and objectives.

Periodic Review

Periodical reviews of existing preservation plans are needed to continually verify the appropriateness of plans, and to improve and further develop existing plans. A periodic review should re-iterate the planning activity taking into account new developed preservation strategies, and seek to verify and potentially improve existing plans.

Type	Examples
Relations	• No previous preservation plans for this purpose exist, so there are no relations to other plans and no triggers why a related plan has to be revised. • No previous attempts to preserve these kinds of documents have been made.
New Collection	
Periodic Review	
Changed Objective	
Charged Collection Profile	
Changed Environment	Changed hardware environment: carrier refresh from LTO tape to HDD (on the Digital Preservation Team SAN). We see this as a very good opportunity to look at the file format for the master images and ensure that it is suitable for long term preservation.

Table 5 Examples for the boxes for Description

1.1.4 Policies

Why policies?

Organizations differ in many ways, type, size, direction, just to to name a few. That's why apart from technical and intellectual properties of digital objects also the strategy, policy, goals and constraints of the institution are an integral part of the preservation plan.

Usually organizations have created documents describing their policies, strategies, workflows, plans, and goals to provide guidance.

The policies template

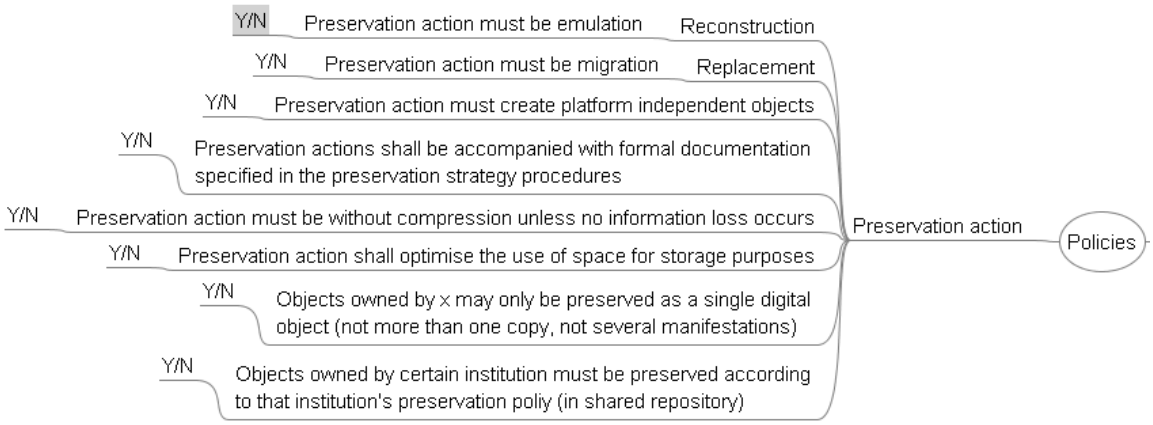
In the course of the Planets project a conceptual model of organizational digital preservation policies and strategies has been created. It incorporates relevant organisational characteristics and strategic directions to support the planning process in digital preservation projects.
Relevant policies have been structured in tree form which can be used as a template for defining your organizational policies.

How to create a policy tree

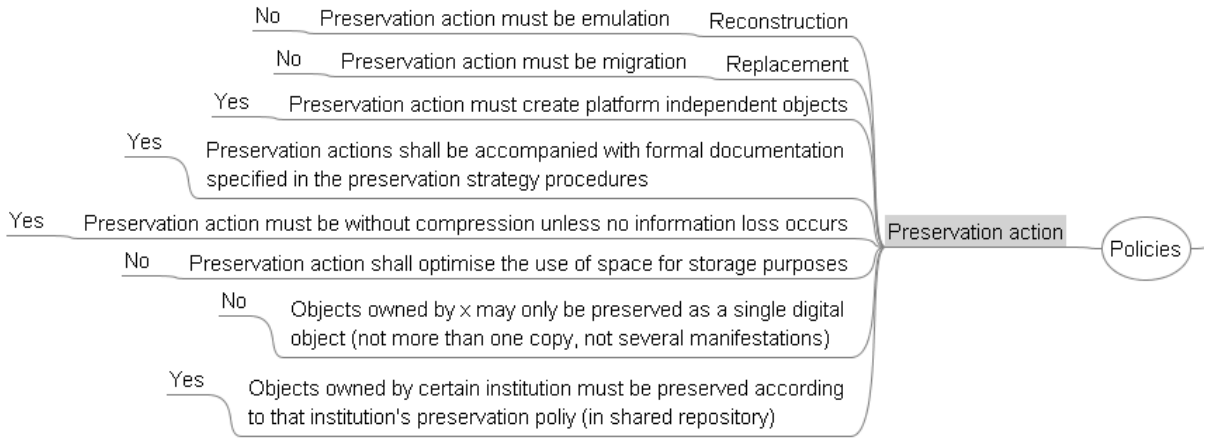
Based on the policy tree you can model you policies by replacing the leaves, which denote the scale

of the policy, by their actual values. You should remove policies you don't have in place by simply removing the node forming the policy.

In the tree, policies have been grouped into several levels, as for instance shown in the screenshot below. The screenshot shows policies pertaining to *Preservation Action*. Leaves in the tree denote scales the particular policy can be measured in. Policy *Preservation Action must be emulation* can be answered either *Yes*, or *No*.



After all scales have been replaced by their actual values, the policy tree might look like follows:



Upload the policy XML file

The workflow step 'Define Basis' allows to upload the policy tree which has been saved as Freemind file (.mm).

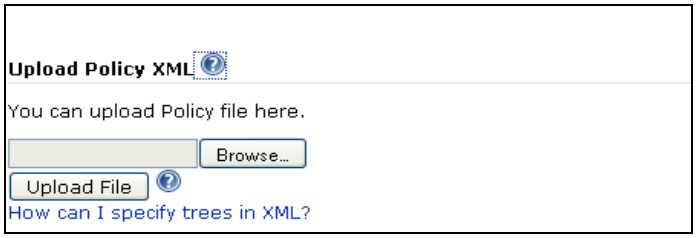


Figure 14 Upload Policy XML

After the tree has been uploaded (by pressing the button 'Upload File') Plato displays the tree and allows you to store the policies with the preservation project.

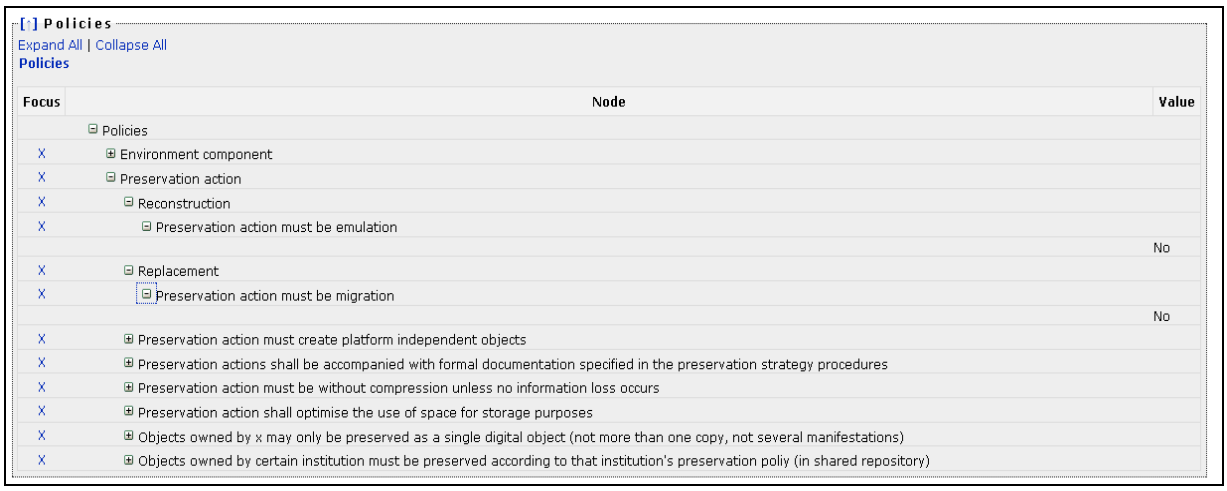


Figure 15 Policies

1.2 Define Sample Records

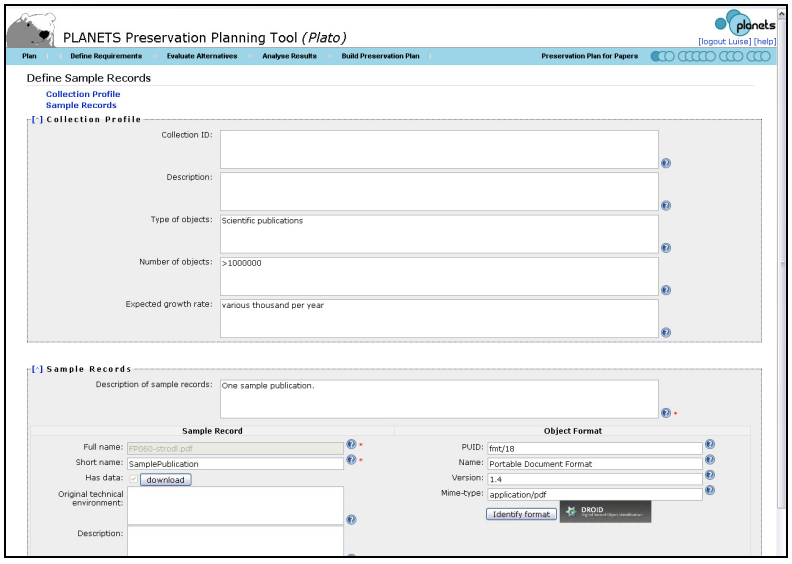


Figure 16 Define Sample Records

The second step describes the set of objects that forms the scope of the current plan, and selects a subset of representative objects for experimentation. A general description of the characteristics of the set of objects, called *collection*, includes basic properties such as the size of the collection, the class of objects and the object formats they are currently represented in. While this can be done in a manual descriptive way, a formal representation is desirable.

1.2.1 Collection Profile

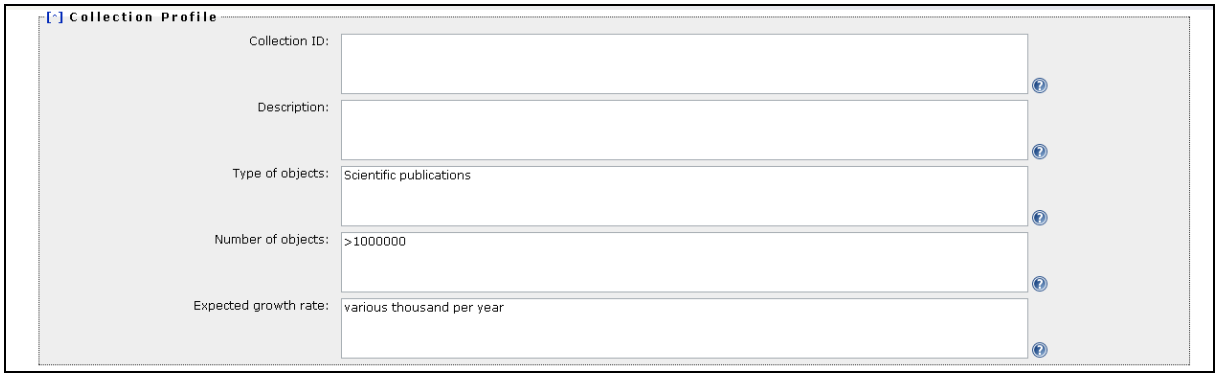


Figure 17 Collection Profile

Collection profiling tools can provide automated descriptions of the technical characteristics of objects. Characteristics of interest include object formats, file sizes and their variation within the collection, but also aspects such as an assessment of the risks of each object type and each object, thus leading to a risk profile of the collection. As a complete evaluation of the quality of preservation action tools is infeasible on the potentially very large collection of objects, the planner selects representative sample objects that should cover the range of essential characteristics present in the collection at hand. To reduce effort to a minimum, this subset should be as small as possible.

Type	Examples
Collection ID	• COLLECTION-12-2008 • Yearbook-collection-TSL-1200
Description	• Physical collection of Cartridges for the Nintendo SNES which have been transferred to binary streams. • The first part of the yearbook collection of the car periodical. This part contains the yearbooks published in the years 1965-1989.
Type of objects	• Scientific publications in PDF • This part of the collection consists of GIF files. • Digital Data from Cartridges of Super Nintendo Entertainment System (SNES) video games (Binary Streams).
Number of objects	• >1000000 • 1000
Expected growth rate	• Slow. Occasionally new games can be added

	to the library's collection. • No magazines have been scanned since 2006 and when the scanning is resumed then they will be scanned directly to the preservation format TIFF. Thus, the future growth of one magazine per year will not be related to this preservation plan.
Retention period ("time horizon")	• Some of the "analog" pages don't exist any longer, so we don't have the option to re-scan them. Furthermore this part of the collection is in high demand and shall thus be preserved for at least the next 10 years.

Table 6 Examples for the boxes in Collection Profile

1.2.2 Sample Records

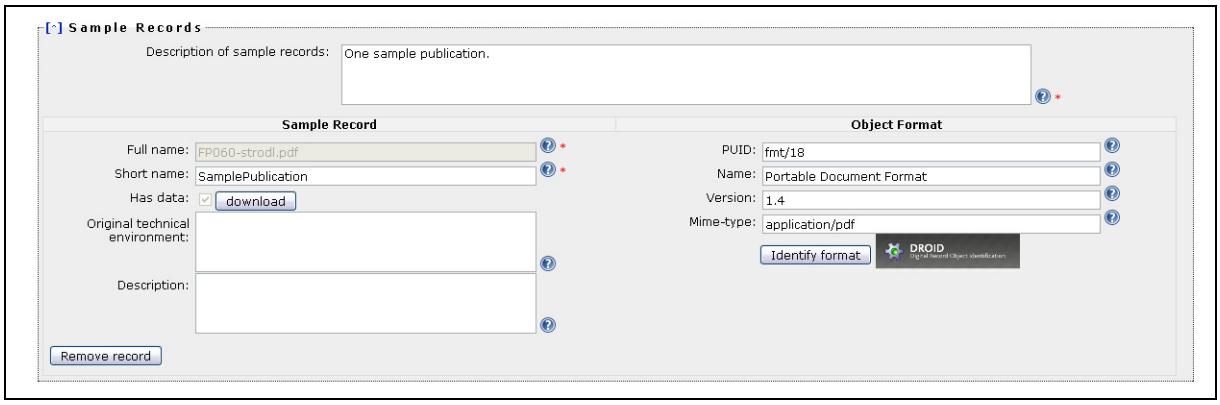


Figure 18 Sample Records

For the samples, an in-depth characterisation is performed, describing the significant properties and their technical characteristics such as their name and provenance, the file format, and specific risk factors. The uploaded sample record can be identified by clicking 'Identify format' on the page. For identification Plato uses the software tool DROID which it calls and displays the output: PUID (Pronom Unique Identifier), Name, Version and Mine.

Plato also provides the function to characterise uploaded files in-depth (for some formats). When you press the button 'Describe sample records in XCDL`, Plato will use characterisation services that extract properties of the objects using the XCL engine and thus producing an XCDL for each objects. This can be used to compare original and transformed objects during experiment evaluation. All information about sample objects, including the characterisation in XCDL, is stored in the preservation plan. Further information on this subject can be found in the publications on the documentation page.

Another integrated well-known characterisation tool is JHove. The screenshot below shows the tree view you get when you click the "View characteristics" button in Plato.

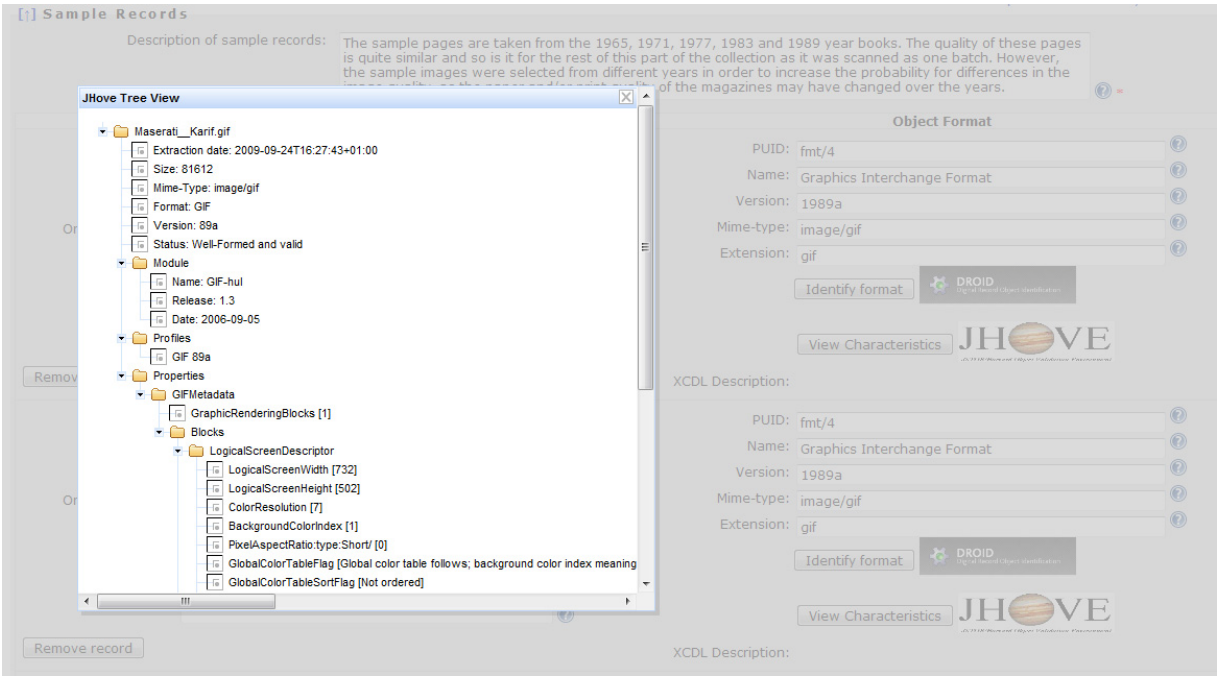


Figure 3 Characterisation of sample objects

Type	Examples
Description of sample records	• The sample pages are taken from the 1965, 1971, 1977, 1983 and 1989 year books. The quality of these pages is quite similar and so is it for the rest of this part of the collection as it was scanned as one batch. However, the sample images were selected from different years in order to increase the probability for differences in the image quality, as the paper and/or print quality of the magazines may have changed over the years.
Sample Record: Full name	• Very large publication containing > 20 figures.
Sample Record: Short name	• Very large publication
Sample Record: Original technical environment	
Sample Record: Description	• This is our larges publication file we have in our collection. It contains a lot of figures, tables and equations. • Randomly selected image to see if there would be

	any difference in the print quality since 1983.
Object Format: PUID	• Fmt/18 • fmt/4
Object Format: Name	• Portable Document Format • Graphics Interchange Format
Object Format: Version	• 1.4 • 1989a
Object Format: Mime-type	• image/gif

Table 7 Examples for the boxes in Sample Records

1.3 Identify Requirements

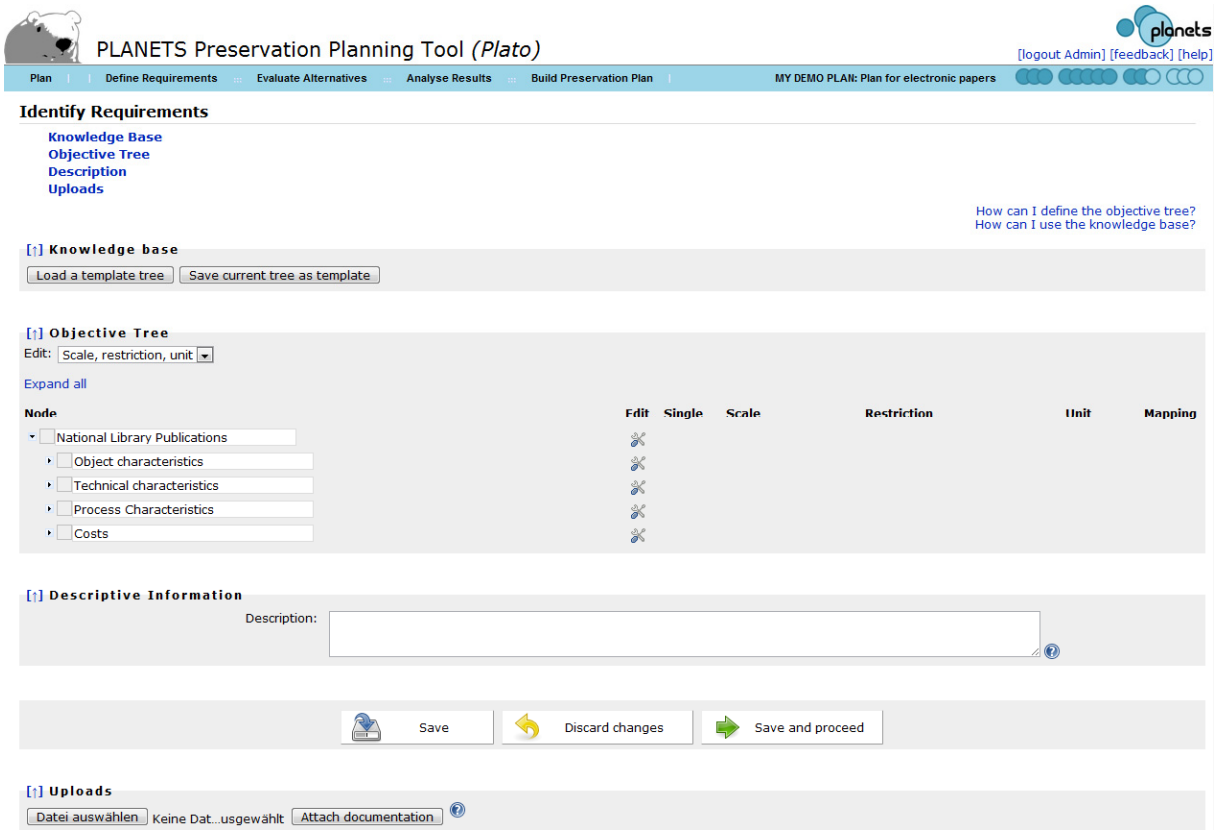


Figure 4 Identify requirements

Requirements definition is the heart of preservation planning. It is the basis for the decisions to be taken and documents the priorities and preferences of the institution. Requirements are collected from the wide range of stakeholders and influence factors that have to be considered for a given

institutional setting. This may include the involvement of curators and domain experts as well as IT administrators and consumers.

1.3.1 Objective Tree

The requirements are specified in a quantifiable way, starting at high-level objectives and breaking them down into measurable criteria, thus creating an *objective tree* which forms the basis of the evaluation of alternative strategies. This step has proven to be the most critical and complicated stage in the planning procedure. An incomplete requirement specification leads to a skewed evaluation and potentially wrong decisions. On the other hand, curators tend to exhibit a reluctance to quantify their preferences, and especially try to avoid questions such as *What is the loss I am willing to accept?* which are of central importance. The complexity involved in specifying goals and breaking them down to concrete, quantifiable criteria is a considerable challenge. However, through iterative refinement of abstract goals such as I want to preserve these objects exactly as they are towards more concrete requirements (The size needs to be unchanged) we ultimately arrive at measurable criteria such as The image width, measured in pixel, needs to be unchanged. As requirements need to be negotiated between the stakeholders, a common approach is to define the requirements in a workshop setting where as many stakeholders as feasible are involved. On a practical level, two tools have been very useful: post-it notes and mind-mapping software. While post-it® notes and flip charts as classical tool to support brainstorming activities have the benefits of allowing everyone to act at the same time, mind maps provide the better overview of the current state of requirements for all participants and allow a moderator to channel the discussion process. Often, a combination of both tools is the most productive approach. While the resulting objective trees usually differ through changing preservation settings, some general principles can be observed. At the top level, the objectives can usually be organised into four main categories – characteristics of the objects, the records, and the process, and requirements on costs.

Object characteristics describe the visual and contextual experience a user has by dealing with a digital object. These characteristics are often referred to as significant properties. A common way of describing them is to consider the five aspects “Content”, “Context”, “Structure”, “Appearance”, and “Behaviour”.

The tree contains the requirements for preserving a collection of static web pages containing documents and images. The branch Behaviour is divided into three different groups of criteria: deactivate, preserve, and freeze. This reflects the preferences of the archive that some functionality, such as menu navigation, is needed for properly accessing the web pages, while most active content shall be disabled or frozen. For example, visitor counters shall be preserved in the state they had at the moment of ingest.

Recently, several plans such as INSPECT have presented detailed analysis of the significant properties of different categories of objects, including vector images, moving images, e-Learning objects, and software. These can provide a very valuable input to this aspect of requirements specification. On the other hand, the automated characterisation of the sample objects defined in the previous step further supports the analysis of their significant technical properties.

- Record characteristics describe the foundations of a digital record, the context, interrelationships and metadata.

- Process characteristics describe the preservation process itself, for example the procedure of migrating objects. These characteristics include the complexity of applying preservation action tools or their performance and usability, but equally should cover aspects such as documentation or the degree of validation. The definition of process characteristics is particularly dependant on the specific context the preservation process is taking place. The technical environment may effectuate specific requirements on the interoperability of tools, while institutional policies or legal regulations may enforce specific licensing requirements or require a particular degree of automated documentation. Thus the institutional and technical context and constraints posed by it have to be considered carefully.

- Costs have a significant influence on the choice of a preservation solution, but are inherently hard to quantify. Ultimately the Total Cost of Ownership (TCO) is the guiding figure for deciding whether or not a preservation strategy meets the needs of an institution within the constraints of its budget. Instead of providing a single numeric criterion which is extremely complex to quantify, costs might also be defined as infrastructure characteristics, putting an emphasis on cost factors instead of the resulting figures for cost estimates. These cost factors can then be further broken down to cover hardware, software, and staff costs.

The objective tree thus documents the individual preservation requirements of an institution for a given partially homogeneous collection of objects. The tree as such is entirely independent of the strategy employed, be it migration or emulation. It is of vital importance that it is concerned solely with the *problem space* and does not specify solutions such as *We want to migrate to PDF/A*, unless these decisions have been made already on a higher level. While such specifications are sometimes brought forward in the requirements workshops, they commonly can be traced back to the reasons underlying them, such as preferences for transforming objects to standardised, widely supported file formats and deactivation of active content. The decision to migrate to PDF/A using a specific tool right be the right one; however, without proper documentation of the reasons and the evaluation leading to it the recommendation cannot be considered trustworthy. The tree shown in Figure 2 contains a branch named *technical characteristics*. In this specific case, the institutional policy constrained the class of preservation action to be considered to migration; emulation was not an option. Thus the requirements describe in a very specific form the desired characteristics of the target format the objects should be kept in. These characteristics together form a *risk assessment* of the format and become a central part of evaluating applicable tools and strategies.

An essential step of requirements definition is the assignment of measurable effects to the criteria at the leaf level of the objective tree. Wherever possible, these effects should be objectively measurable (e.g. e per year, frames per second, bits per sample). However, in some cases, (semi-) subjective scales need to be employed. For example, the quality of documentation that is available for a file format or a tool should not be judged by the number of pages alone; instead, a subjective scale such as *excellent, good, average, poor, very poor* could be used. Similarly, the *openness* of documentation of a file format could be one of *fully standardised*; *openly published*, but not standardised by a recognized body; and *proprietary*.

The assignment of measurable effects to criteria can also align them with characteristics that can be automatically extracted from objects to automate the evaluation procedure. Existing software tools

such as JHove allow automated extraction of some of the basic properties of common object formats; the eXtensible Characterisation Languages provide an in-depth description of the complete informational content of an object in an abstract representation. These descriptions can be used to derive properties to be measured, and support the automated comparison of these properties when migrating the objects to different formats.

Ongoing case studies revise and extend the previously conducted evaluation studies, build concrete preservation plans for specific collections of objects, and cover new types of objects that have not been evaluated yet in a variety of institutional settings.

The experience which is accumulated through carrying out planning activities and requirements definition can be shared between institutions easily through the supporting software, which contains a knowledge base of recurring fragments of objective trees and templates that can be used as a starting point.

How to specify requirements in the objective tree editor

Using the tree table editor, you can add and remove inner nodes and leaf nodes. Every leaf criterion denotes a requirement that can and must be measured. It therefore has to be assigned a measurement scale.

You can save an incomplete tree and continue working on it later. However, it is not possible to proceed in the workflow with an invalid objective tree.

The outcome of the first phase is a complete documentation of the planning context, the collection of objects at question, and the specific requirements that form the basis for the evaluation of alternative action paths.

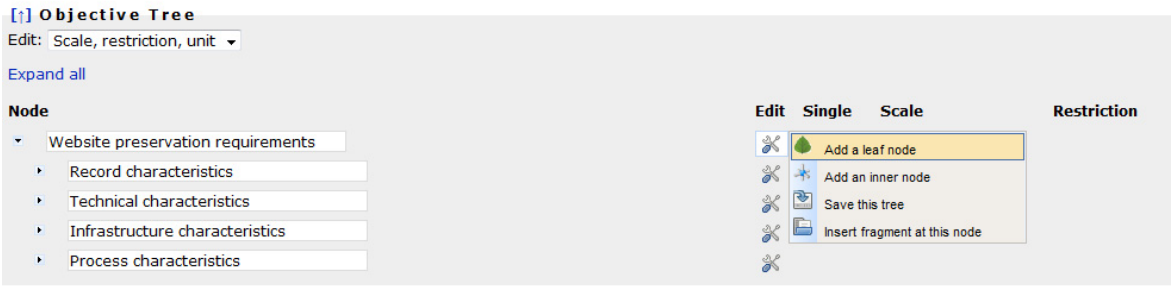
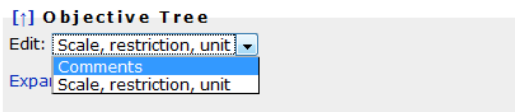


Figure 5 Tree editor

There are two views of the tree: The classic view with names and scales, and a documentary view that allows you to describe each node. You can switch between these views with the dropdown menu on the top left corner of the tree:



Switching to the "comments" view yields the following editor:

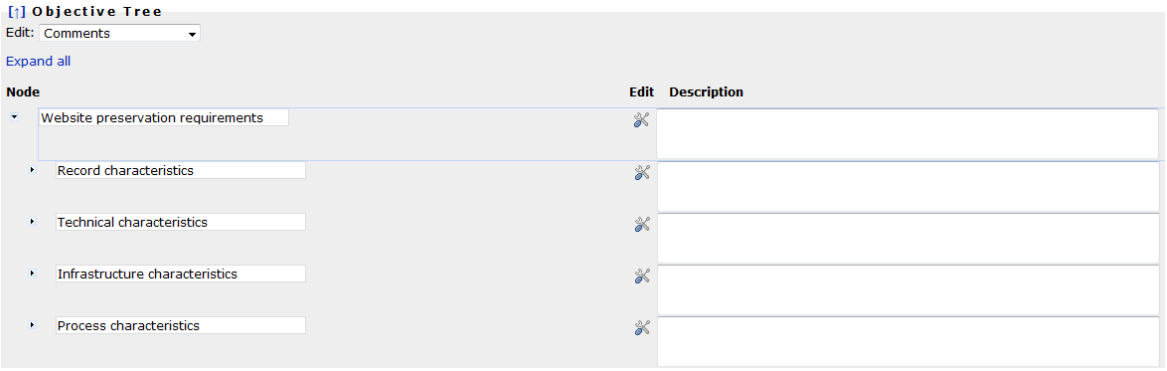


Figure 6 Node descriptions

Every node needs a name. To ensure you are entering a name for a new node, the name field is shown red until you provide a name. You cannot edit the tree while a field is red.

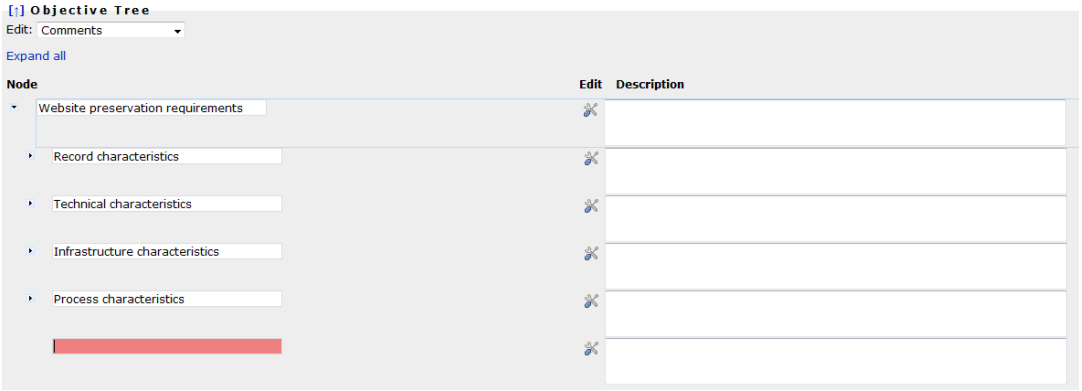


Figure 7 Node names are mandatory

Specifying scales

The following types of scales are supported:

- Boolean scales are binary yes/no decisions.
- "Integer Range" scales are integer numbers with lower and upper bounds that are set to 0 and 5, respectively, by default. You can change these boundaries in the restrictions field.
- Ordinal scales can be used for criteria that cannot be measured automatically or that you want to judge by yourself. For example, Availability of documentation in the provided example can have one of the values
    - public,
    - limited, or
    - none.

- Positive numbers and positive integers can be any number above or equal to zero. You can specify an upper bound by typing a number in the restriction field.
  You have to specify a measurement unit for these scales.

- Yes-Acceptable-No scales are an often-used type of ordinal scales and therefore directly accessible.

 **You have to specify a scale for every leaf node before you can proceed with the workflow.**

Single values: By default, every criterion is applicable to every single sample record. For example, a criterion specifying that the image width has to remain unchanged has to be checked for fulfillment for every sample record. However, there are criteria that do not need to be evaluated for every sample record again. For example, a criterion saying that the target format needs to be an official standard applies to an alternative action as a whole, not to each single object. Thus it can be marked as a single criterion that only has to be evaluated once per alternative.

Note that for example the criterion conforms to file format should be checked for every sample object, because a tool might fail to produce correct output files for specific input files. Single criteria thus denote criteria that are to be evaluated once per alternative.

Setting restrictions: Restrictions specify a scale more concretely and constrain the possible values. For ordinal scales, the restriction specifies the list of possible values that the scale contains. In this case, the specification of a restriction is obligatory.

For example, in the scale for 'availability of documentation' mentioned above, the three possible values have to specified. You can do so by setting the scaletype to 'ordinal' and then specifying the restriction 'public/limited/none' in the restriction field.

Boolean (Yes/No) as well as Yes/Acceptable/No are special cases of ordinal scales where you cannot change the restriction settings.

For numeric values, the following rules apply.

a) For IntRange scales, i.e. integer with a specified range, the restriction specifies the lower and upper boundary of the values in the form lower/upper. For example, a restriction of 1/10 specifies a value between 1 and 10.

b) For positive numbers and integers, the restriction specifies only the upper limit. Setting a restriction of '100' specifies that each evaluation value may not exceed 100.

Setting measurement units: A unit is a text specifying the unit of measurement, such as MB, seconds per object, or bits. Units are obligatory for any numeric scale. For other scales, you may specify a unit, but you don't need to.

Mapping criteria to object properties: Some of the criteria in the tree can be evaluated automatically - specifically, by comparing properties of files e.g. before and after migration actions have taken place. The right column of the objective tree provides the possibility to create a mapping

to object properties that can be extracted automatically from the set of sample records. This characterisation relies on the eXtensible Characterisation Languages (XCL).

The screenshot below shows a tree fragment where all but one leaves are mapped to object properties that can be compared automatically. The color of the paperclip on the icon shows if a property has been mapped or not.

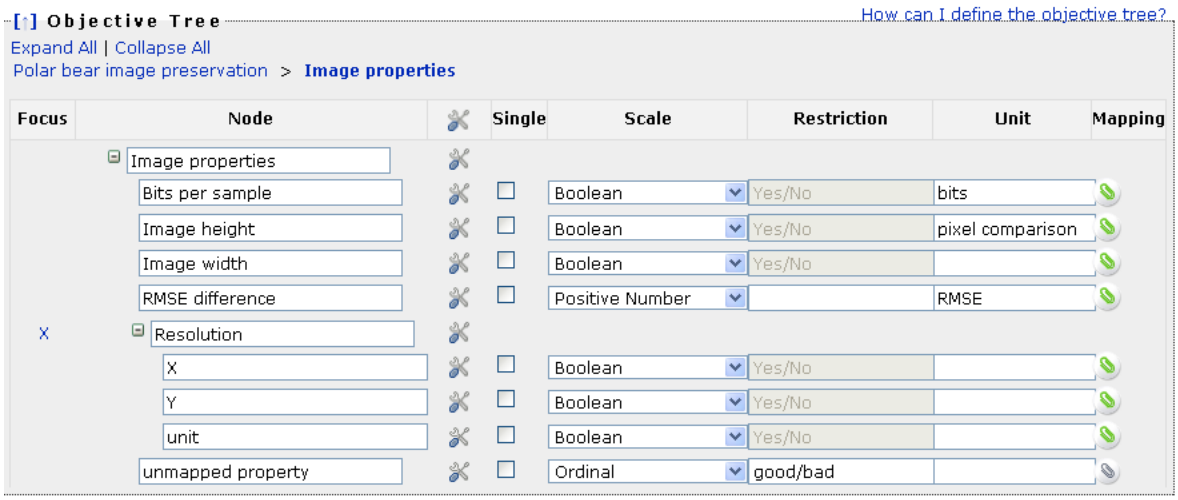


Figure 8 Mapping criteria to object properties

By clicking on the icon on the right column, you can edit the mapping as shown in the following screenshots.

1. First you choose the property with which the corresponding criterion shall be associated.

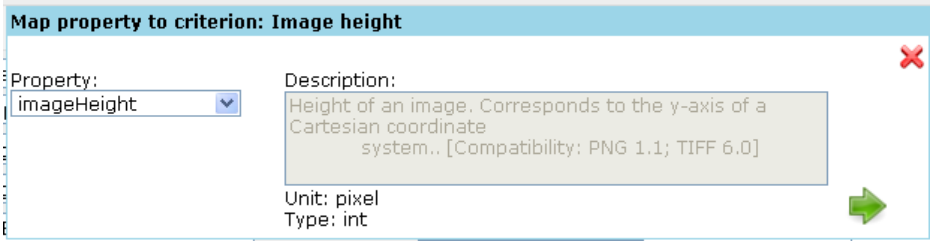


Figure 9 Map property to criterion

   You can confirm using the green arrow on the right.

2. You define which comparison metric shall be used to compare original and transformed objects. Please note that this metric will determine the scale to be used for measuring the leaf criterion.

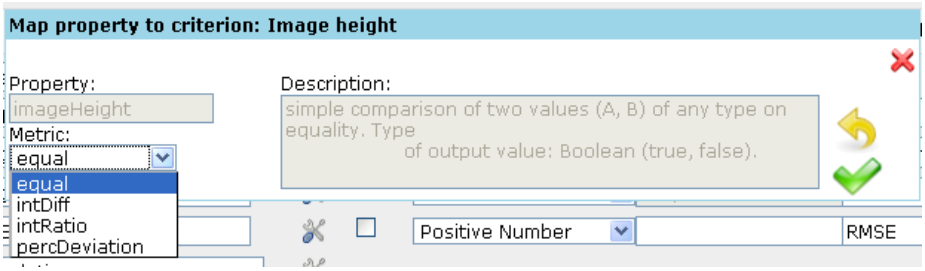


Figure 10 Choosing metrics

3. Finally, confirm again by clicking the green arrow, and the popup will close.

1.3.2 Templates and fragments: The knowledge base

To allow you the full or partial reuse of an objective tree which you have already defined within a former project, or simply to assist you in creating a nicely structured objective tree from scratch, Plato includes a knowledge base that holds useful templates and fragments of requirements.

You can save your trees (or parts of your trees, so-called fragments) in the library, and load templates or fragments from it.

- **Templates** should be used to store complete objective trees, be it from former case studies or just rough tree-layouts which you might want to re-use for other projects. When selecting a template, your project's current objective tree will be replaced by the template, which you can then customize to your needs.
- **Fragments** on the other hand can be used to store and organize snapshots of parts of an objective tree for partial reuse. They can be to added to any inner node in your project's objective tree.

If you want to start building your objective tree by using one of the templates provided in the template library, simply click the corresponding button:

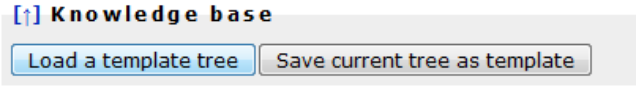


Figure 19 Button Show the Fragments & Templates

The knowledge base interface will open (it might take a second).

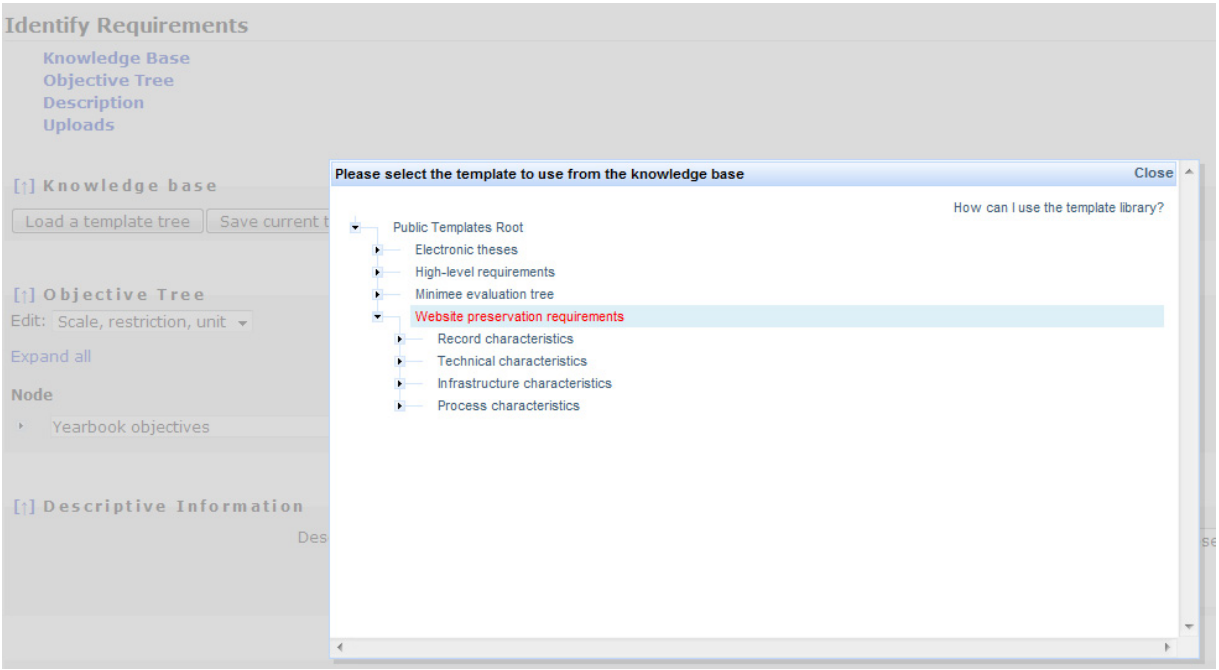


Figure 11 Selecting a template

You select the template you want to use by clicking on its name. Please allow a few moments for this operation. The popup will disappear and your current objective tree will be replaced by a copy of the selected template. Now you can start customizing it to your needs.

Saving a template: To save an objective tree into the knowledge base, open the edit menu next to the root node and select "Save this tree". A copy of the tree will then be stored into the library.

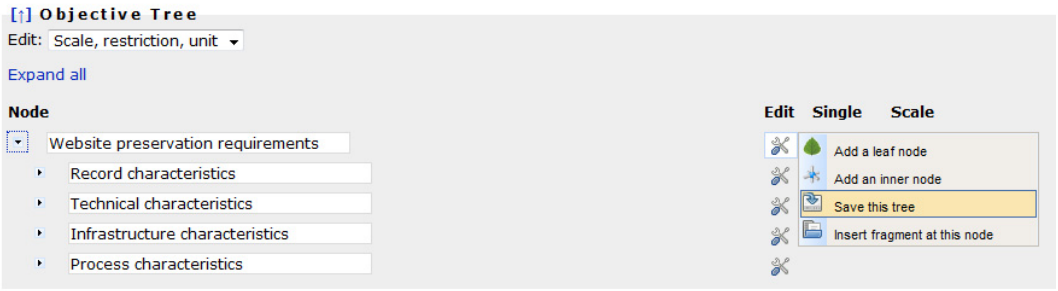


Figure 12 Saving a template

You can change its name and description of the tree before saving, as shown below:

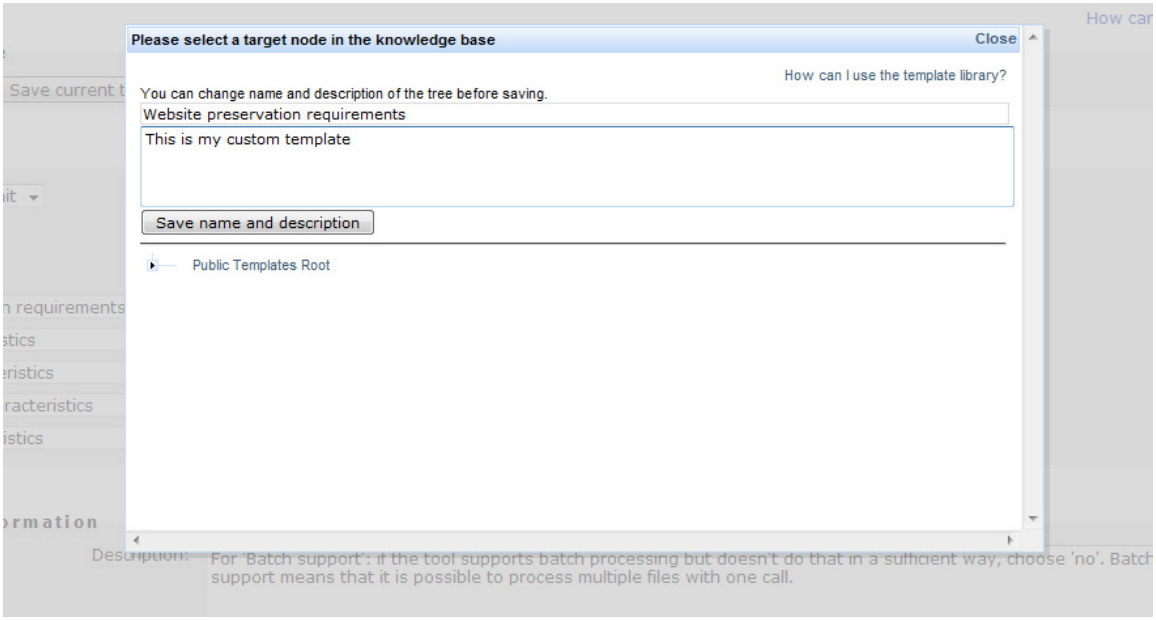


Figure 13 Changing name and description of a template

To complete the operation and store the tree, select the target node in the library to which you want to add the tree. The popup window will close, and a copy of your tree is stored in the public library and accessible to all users.

Saving a fragment: To save a node and its children to the template library, open the edit menu next to it and select "Save this branch":

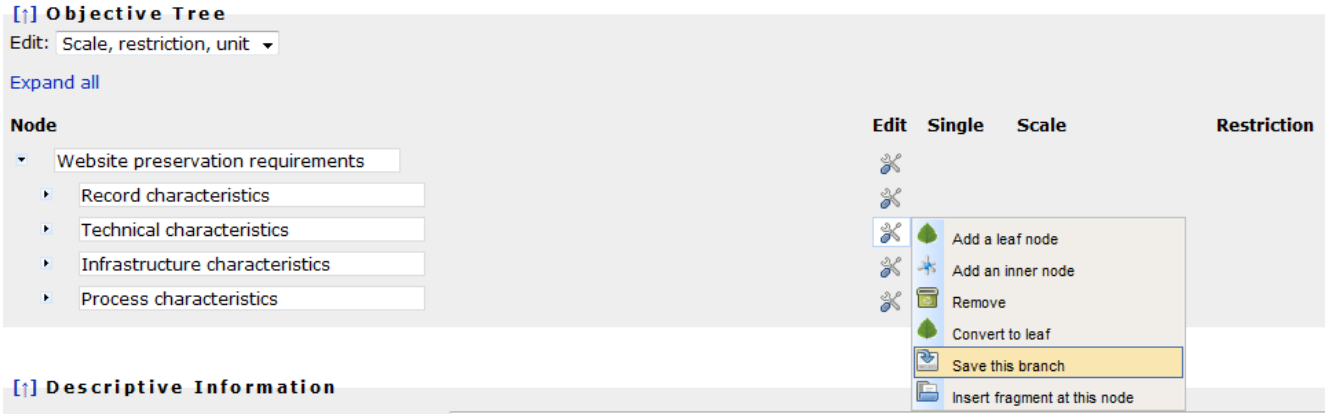


Figure 14 Saving a fragment

As with the templates, you can edit its name and description. To complete the operation and store the branch, select the target node in the library to which you want to add the fragment. A copy of the node, including its child-nodes and leaves as well as their scales and restrictions, is now stored in the library.

Inserting fragments from the fragments library into your objective tree: If you want to insert a fragment from the template library into your objective tree, open the edit menu next to the node where you want to insert the fragment, and select "Insert fragment at this node". (Insertion operations are only possible at inner nodes, not at leaves.)

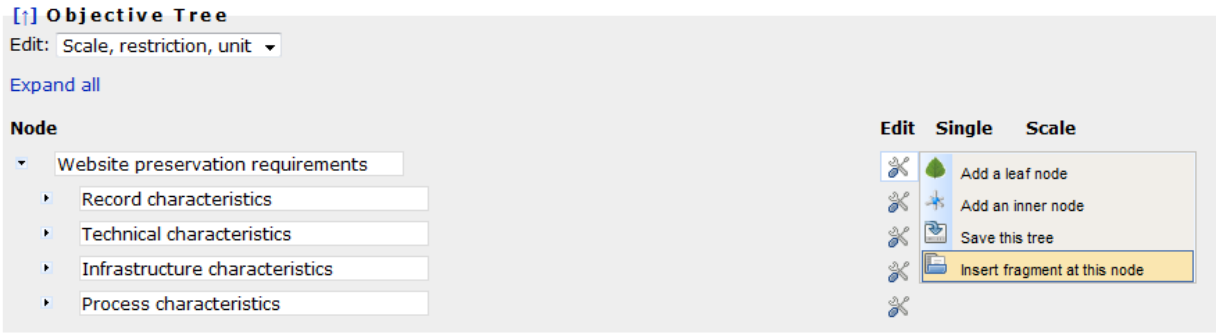


Figure 15 Inserting a fragment

The library popup will appear, asking you to select a fragment for insertion:

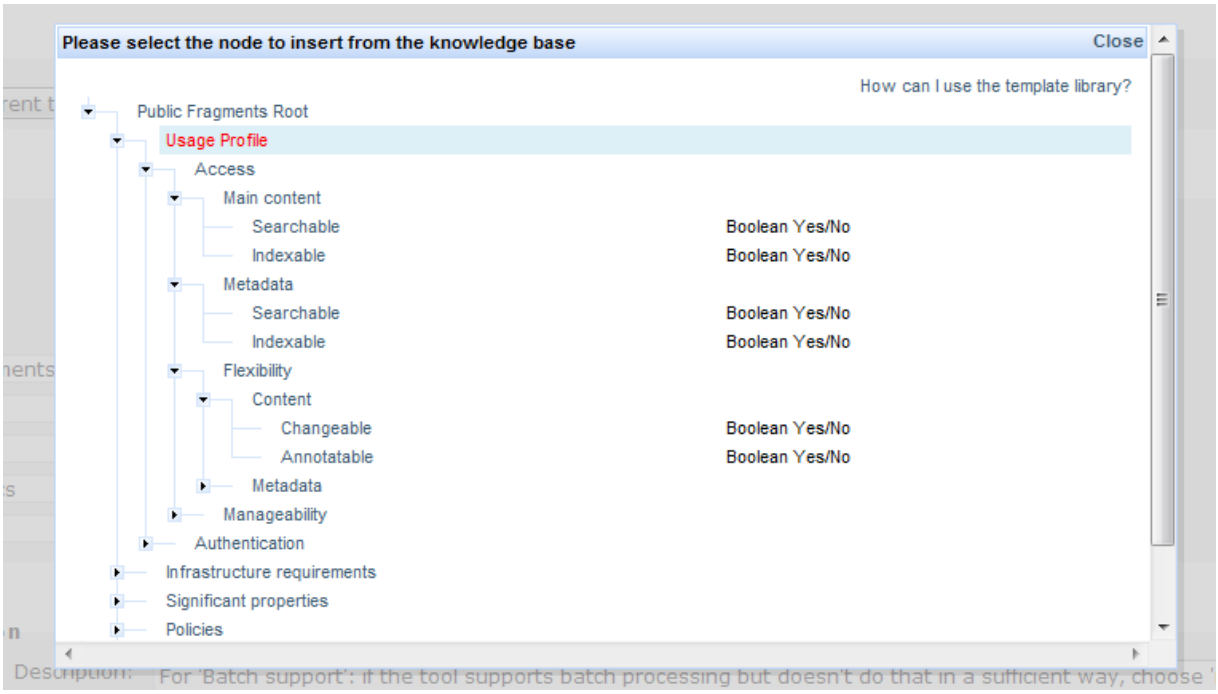


Figure 16 Selecting a node to insert

Again, to complete the operation and store the branch, select the fragment to insert and allow a few moments for the operation. The popup will close and the fragment will be inserted at the chosen location.

1.3.3 Descriptive Information

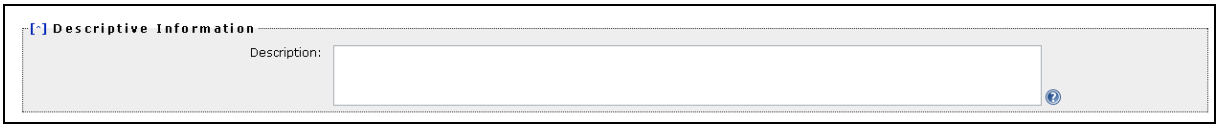


Figure 20 Descriptive Information

Type	Examples
Description	• For 'Batch support': if the tool supports batch processing but doesn't do that in a sufficient way, choose 'no'. Batch support means that it is possible to process multiple files with one call.

Table 8 Examples for the boxes in Descriptive Information

2. Evaluate Alternatives

The second phase of the planning workflow evaluates potential actions in a quantitative way by applying them to the previously defined sample content and analysing the outcomes with respect to the requirements specified in the objective tree. This empirical evaluation procedure results in an evidence base that underlies the decisions to be taken in the successive phases.

2.1 Define Alternatives

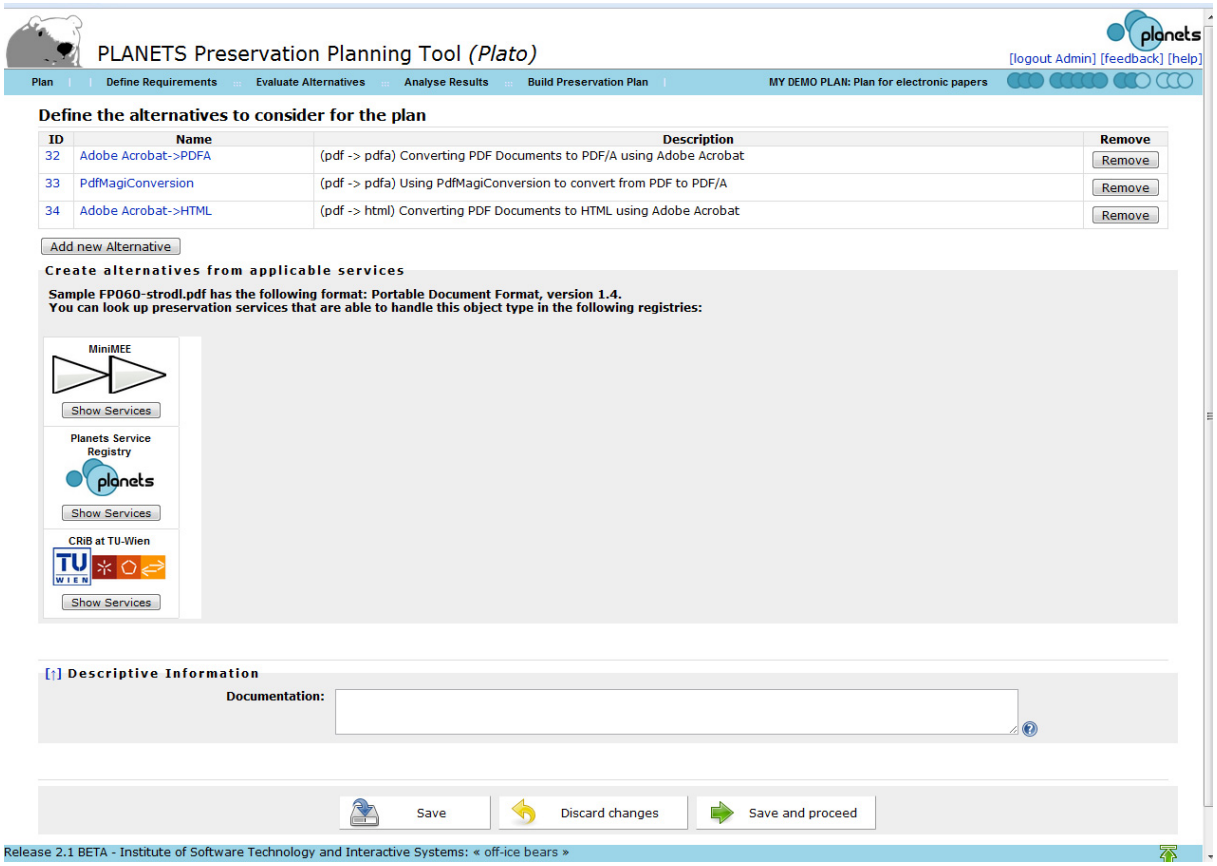


Figure 21 Define the alternatives to consider for the plan

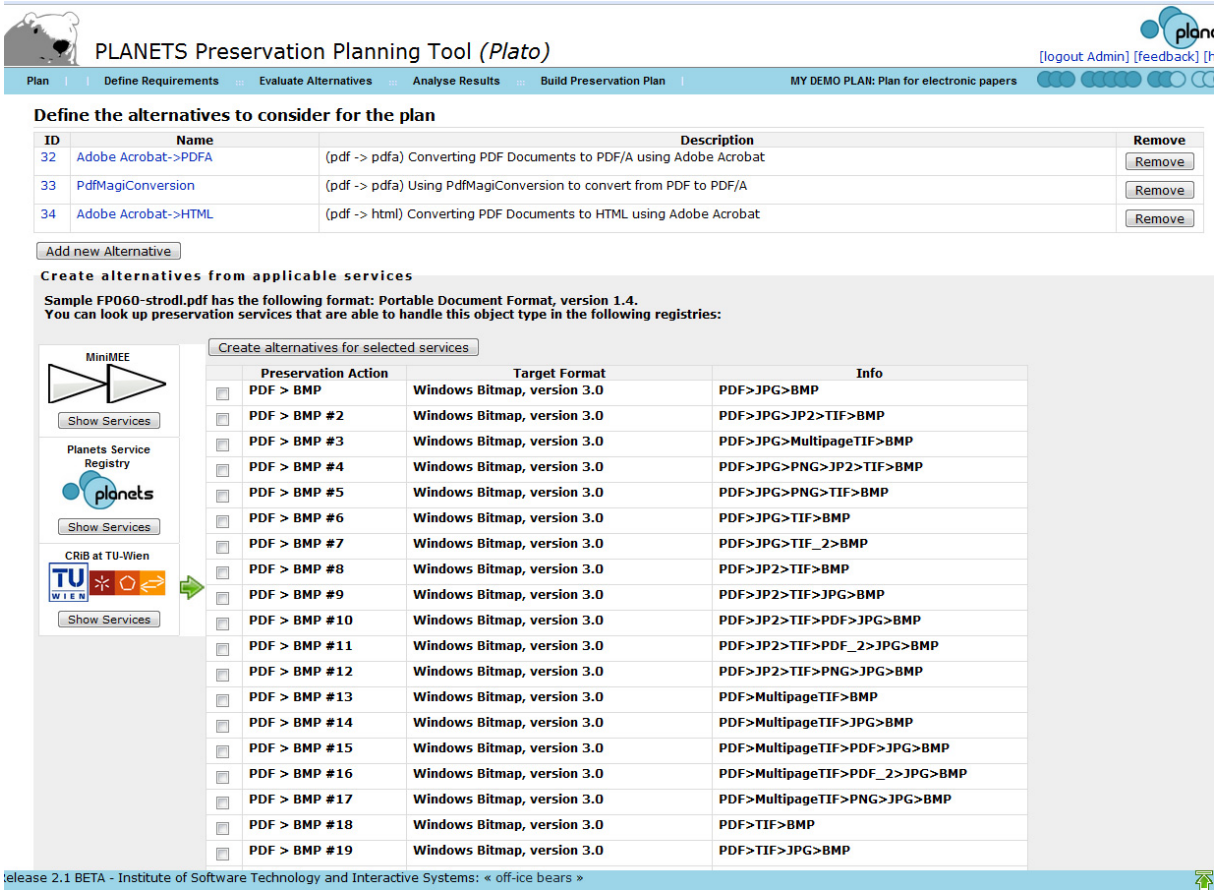


Figure 22 CRIB Service Registry: list of potential alternatives

The MiniMEE registry is a prototype that we have developed to show quality-aware migration services. The tools you find there are executed in a controlled environment. When running these, you will receive detailed information about their processor usage, memory consumption, and other runtime characteristics. Details are available on the documentation page.

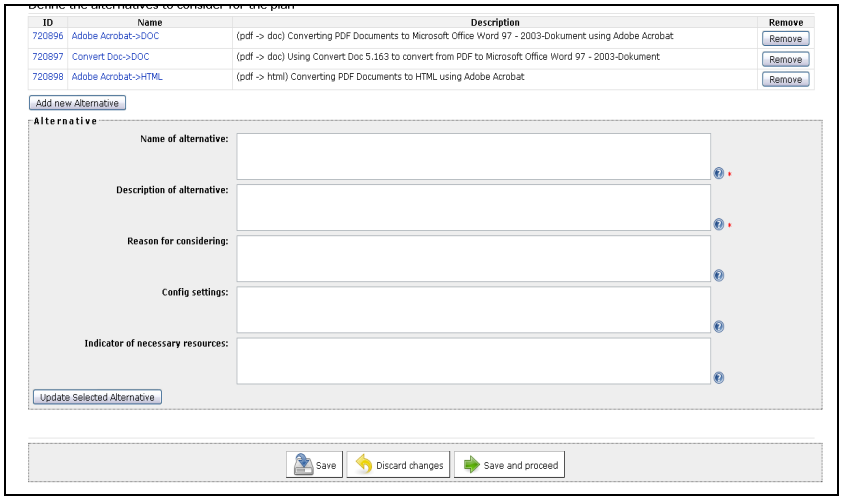


Figure 23 Add new Alternative

The natural first step of evaluation is to define the possible courses of actions to be taken into consideration. A variety of different strategies might be applicable; for each alternative action, a

complete specification of the entailed steps and the configuration of the software tool employed is desired. The discovery of potential actions that are applicable varies in complexity according to the type of content. Often, this implies an extensive search phase, investigating which tools are applicable to the type of objects at hand. Registries holding applicable preservation action tools can be consulted for reference and are potentially very beneficial to support the search. The outcome is a *shortlist* of potential candidates for performing preservation actions, which will be evaluated empirically during the next steps.

2.2 Go/No-Go

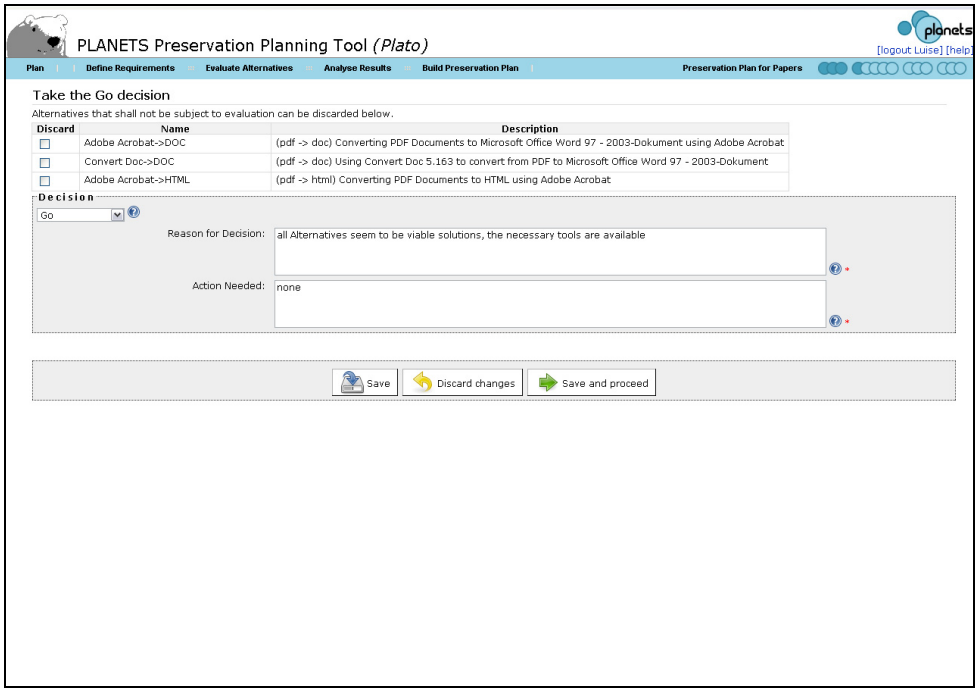


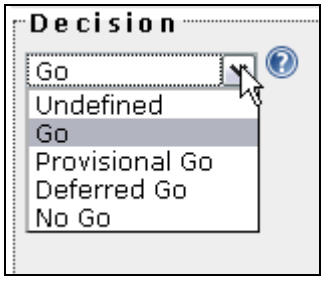
Figure 24 Take the Go decision

Before continuing with the experimentation procedure, this step reconsiders the situation at hand and evaluates whether it is feasible and cost-effective to continue the planning procedure. In cases where the evaluation is considered infeasible or too expensive, a reduction of candidate tools might be necessary. In order to proceed to the next workflow step, the user has to take the GO decision.

Type	Examples
Reason for Decision	• All Alternatives seem to be viable solutions, the necessary tolls are available
Action Needed	• none

Table 9 Examples for the boxes in Take the Go decision

2.2.1 Take the go decision



2.3 Develop Experiments

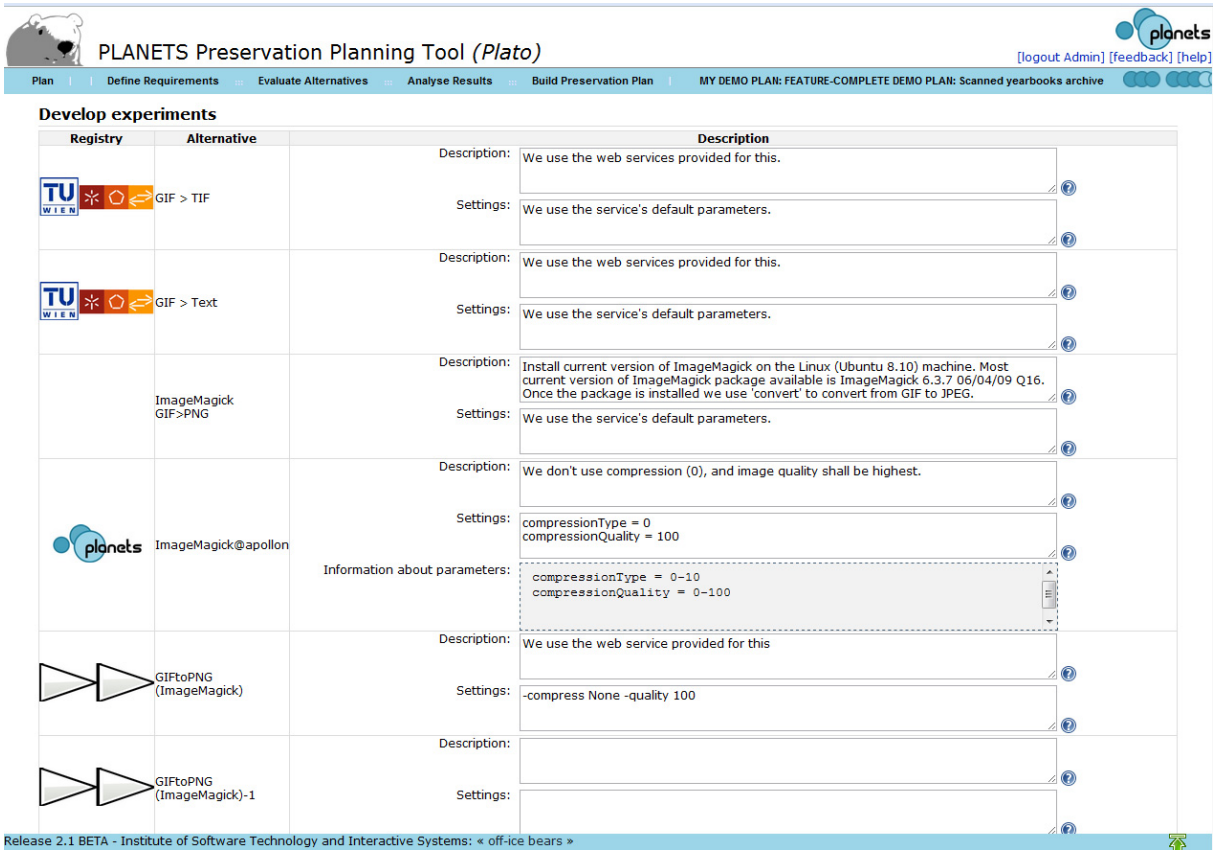


Figure 25 Develop Experiments

This step defines and documents the configuration of the tools on which experiments are carried out, and thus builds the basis for experiment execution in the next step. This includes setup procedures, a documentation of the hard- and software environment, and additional steps needed to carry through the evaluation of experiments.

Type: Alternative	Examples for Description
Alternative: Adobe Acrobat -> DOC	• Standard Settings of Adobe Acrobat, Windows XP Workstation with Service Pack 3 used for migration • Install current version of ImageMagick on

	the Linux (Ubuntu 8.10) machine. Most current version of ImageMagick package available is ImageMagick 6.3.7 06/04/09 Q16. Once the package is installed we use 'convert' to convert from GIF to JPEG.
Alternative: Doc-> DOC	• Standard Settings of Convert Doc, Windows XP Workstation with Service Pack 3 used for migration
Alternative: Acrobat -> HTML	• Standard Settings of Adobe Acrobat, Windows XP Workstation with Service Pack 3 used for migration

Table 10 Examples for Develop Experiments

2.4 Run Experiments

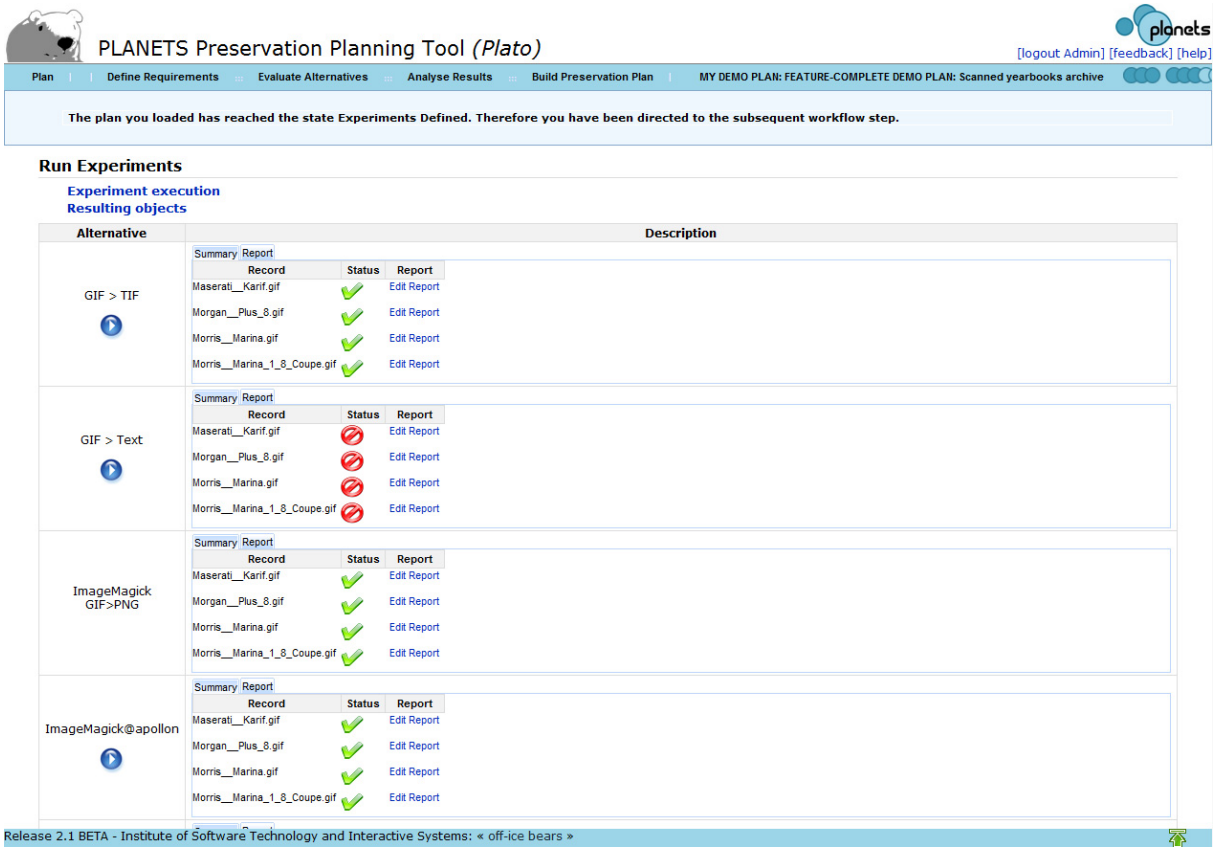


Figure 26 Run Experiments

In this step, all considered candidate tools for preservation actions are applied to the complete set of sample objects that have been defined in the first phase. This produces a series of experiment results that can be analysed and are stored for future evidence. In the case of object conversion, this means that the resulting output files shall be stored for further reference. When evaluating emulators, a documentation detailing the experience of rendering of the object is needed.

Type: Alternative	Examples for Description
Alternative: Adobe Acrobat -> DOC	• Standard Settings of Adobe Acrobat, Windows XP Workstation with Service Pack 3 used for migration
Alternative: Doc-> DOC	• Standard Settings of Convert Doc, Windows XP Workstation with Service Pack 3 used for migration
Alternative: Acrobat -> HTML	• Standard Settings of Adobe Acrobat, Windows XP Workstation with Service Pack 3 used for migration

Table 11 Examples for the boxes in Run Experiments

2.4.1 Results Files

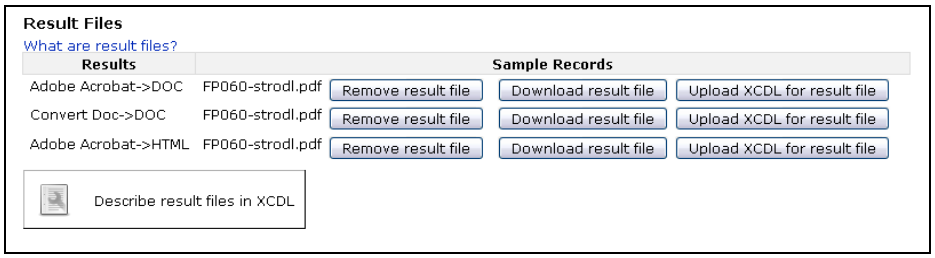


Figure 27 Result Files

2.5 Evaluate Experiments

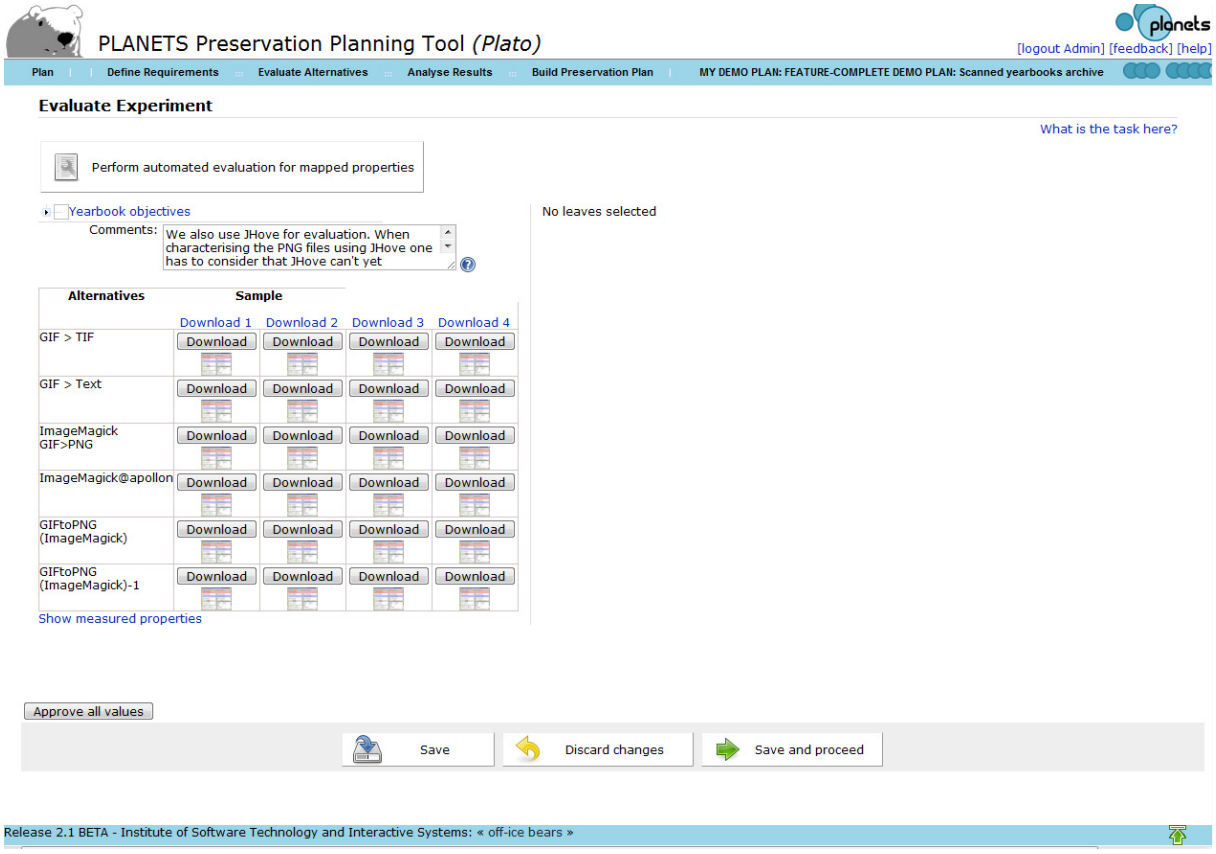


Figure 28 Evaluate Experiment

The evaluation of experiments is based on the requirements specified in the objective tree. All criteria on the leaf level of the objective tree are evaluated, taking into account the empirical evidence resulting from the experiments conducted.

The initial view of this page provides you with

- an objective tree on the upper left,
- a matrix of sample objects and experiment results on the lower left,
- and an empty space saying "No leaves selected" on the top right, as shown below.

The task of this page is to provide all evaluation values for every criterion that you have defined in the tree.

This means both experimental outcomes judging migration quality or the performance of an emulator, as well certain characteristics of the tool, the process, etc... Some of these evaluation values might be provided by automated measurement tools; we are working on increasing the percentage of these. However, currently you still have to enter many of them manually.

The tree on the top left provides a way of selecting areas in the tree to filter the list of leaves. If you click on the root, the right side will disaply all leaves; if you select a certain branch, it will display all leaf criteria in this branch.

You can expand the tree on the top left to get the screen below:

Now selecting a node will display all leaf criteria under this node and prompt you for evaluation values.

Figure 29 Requirements to evaluate

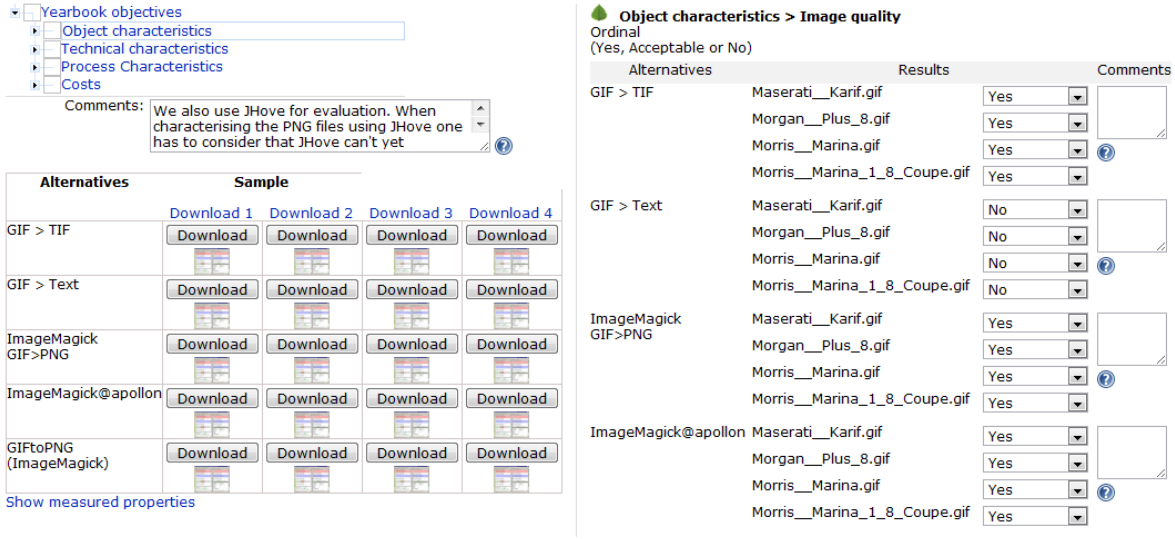


Figure 30 Evaluation of requirements

To support the evaluation, you have full access to the original sample objects and the results that were created during the experiments - either uploaded by yourself or generated by preservation

actions. You can download the original samples on the top of the table and each result file in the corresponding cell of the matrix.

You can further get visual support for judging the quality of migration actions by comparing the charateristics of input and output files extracted by Jhove. Clicking on the matrix in the lower left will bring up a screen that shows the characteristics of original and converted objects side by side, as shown below.

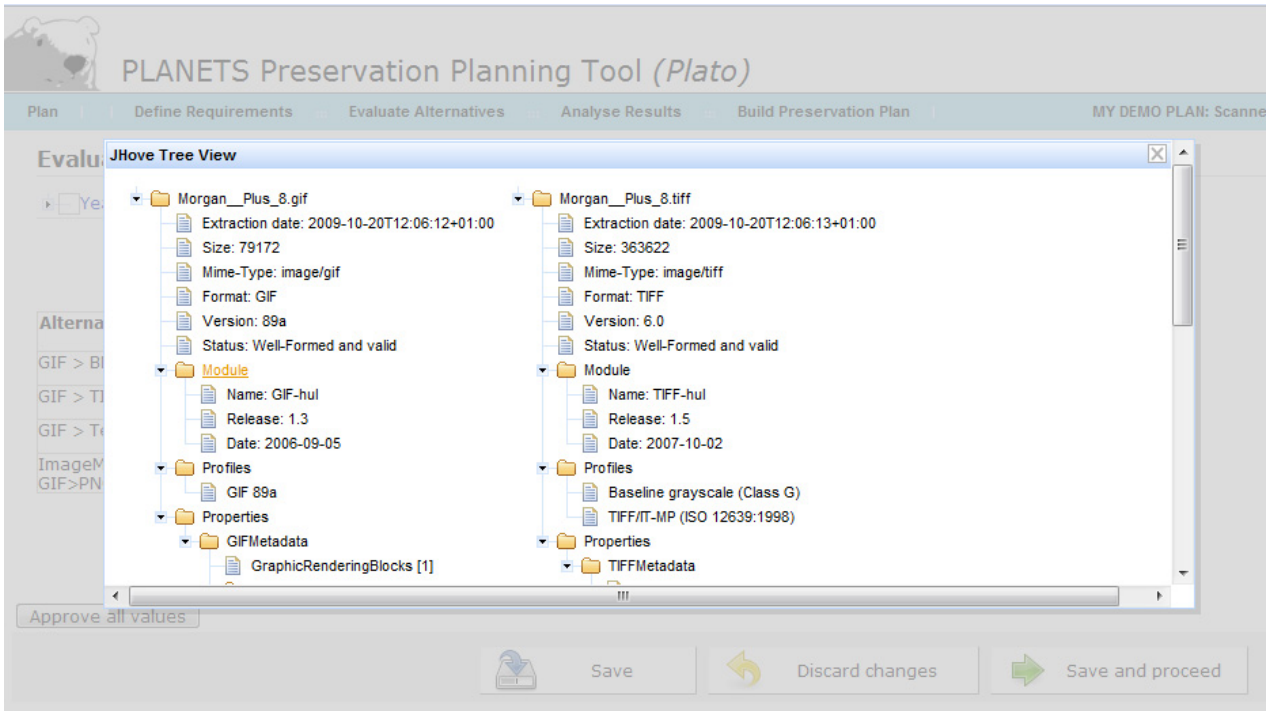


Figure 17 Jhove comparing original and converted objects

For reasons of traceability and auditing, Plato checks that each evaluation field has actually been entered. So you have to either manually specify a value for each field or *confirm* the evaluation data by clicking *Approve all values* before you can proceed to the next step.

3. Analyse Results

3.1 Transform Measured Values

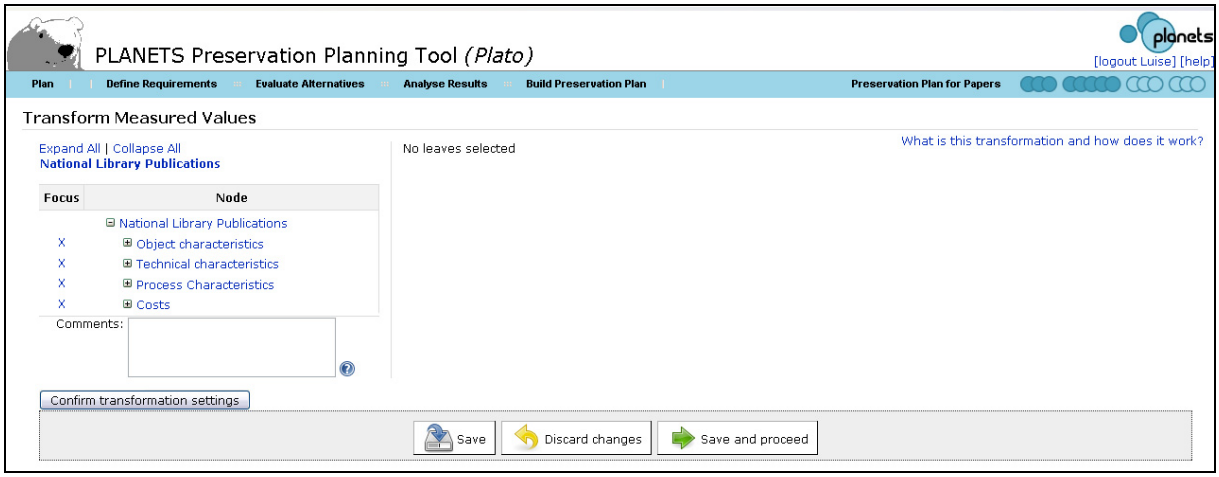


Figure 31 Transform Measured Values

The result of the previous step is an objective tree fully populated with evaluation values for all criteria. However, the measurements in this tree are of varying scales and thus cannot be aggregated and compared directly. Thus, transformation rules are defined which result in a mapping from all possible measurement scales to a uniform target scale. This scale usually consists of real numbers ranging from 0 to 5. The lowest value 0 denotes an unacceptable result, while 5 is the best possible evaluation value. Corresponding to the scales employed, we can distinguish two types of transformation settings: numerical thresholds and ordinal mappings.

- For ordinal values, a mapping is defined for each possible category, resulting in a value between 0 and 5 to be assigned. For a boolean scale, *Yes* might be mapped to 5, whereas *No* will often be mapped to a low value. In this case, a decision has to be made whether the negative result *No* should be acceptable or not, i.e. mapped to 1 or to 0.

- For numeric values, thresholds are defined for each integer number from 1 to 5. All numbers below the lowest threshold (or above the highest, in case of descending order) will then be transformed to 0. The calculation of values between the threshold is usually done using linear interpolation.

In both cases, the definition of *acceptance criteria* is an essential step, where decision makers have to clearly specify the constraints they are willing to accept. This further provides a gap analysis which

clearly points out both strengths and limitations of the candidates under evaluation.

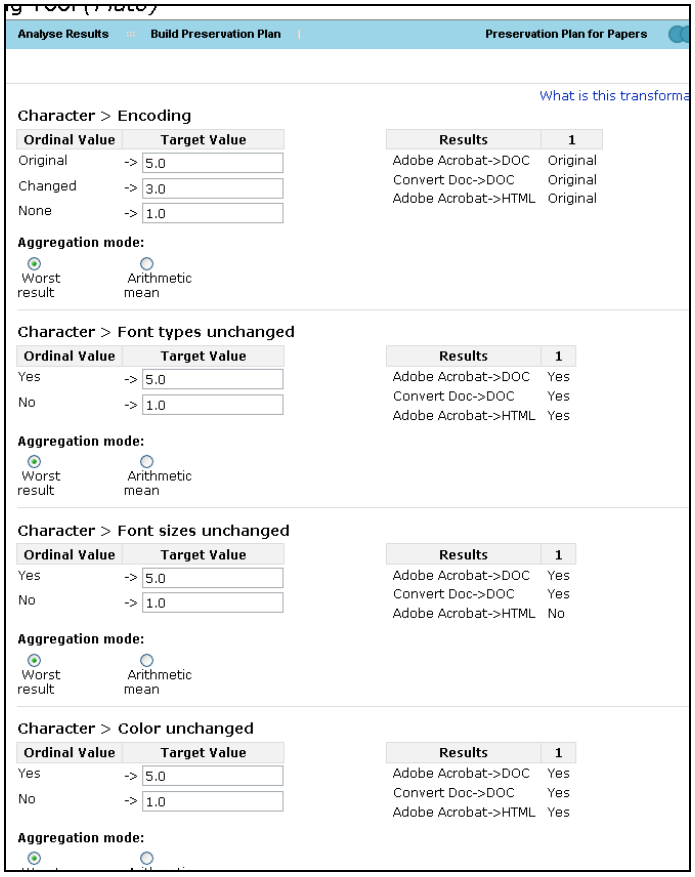


Figure 32 Transformation tables for different requirements

3.1.1 Why Transformation?

Values in the requirements tree are measured in different units (seconds, euro, bits and goodness values). To be able to aggregate the different values they have to be transformed to a uniform scale. Experience has shown that a scale with the resolution of discrete values 0-5 with 0 being an unacceptable value and 5 the best possible result works very well.

3.1.2 Transformation tables

For every leaf in the tree a transformation table has to be created:

Ordinal Values, Yes/Acceptable/No

For every ordinal or Yes/Acceptable/No value a numeric value has to be assigned. For an ordinal value that is considered better a higher numeric value has to be assigned than to an ordinal value that is considered worse. For an ordinal value that makes an alternative completely unacceptable '0'.

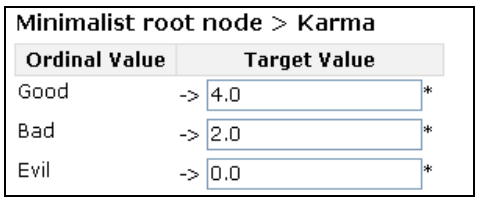


Figure 33 Transformation tables for ordinal scale

Numeric Values (Positive Number, Positive Integer, Int Range)
For threshold values numeric values are assigned. By selecting the appropriate 'Threshold stepping' it can be chosen, if the transformation should be linear (interpolating between the values) or if steps should be used.

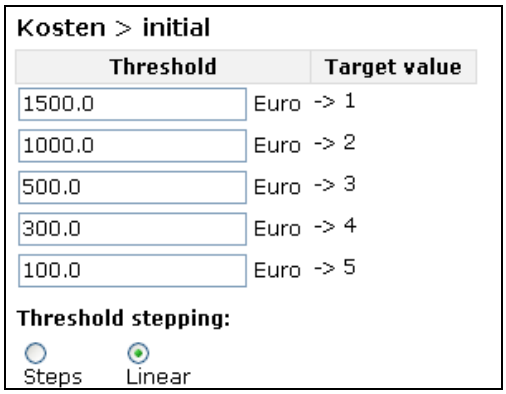


Figure 34 Transformation table for numeric scale

With "Steps" selected for Threshold Stepping, a measured value of e.g. 65 would get the same transformed value as 60 ('3'). If you choose "Linear" the value will be interpolated between the transformation values, so a value of '3,5' would be assigned.

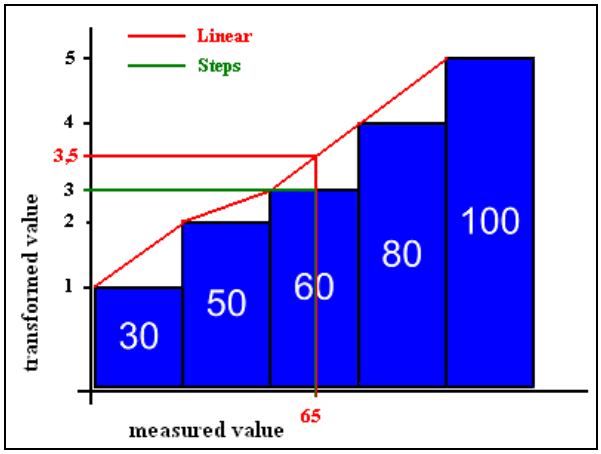


Figure 35 Interpolation of measured values

The measured values from the evaluation are shown next to the transformation tables.
To help you select the correct numeric values for the transformation, all the measured results are shown in a table to the right of the transformation table. For every alternative the sample records (numbered one to number of sample records) are shown in one line.

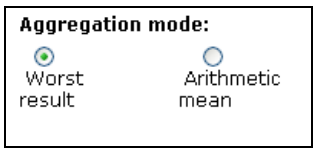


Figure 36 Aggregation mode for transformation

3.1.3 Examples for transformations for ordinal values and numeric values

Transformation of ordinal values

In this example the enumerations in a document are evaluated. If the enumeration is available and like it is in the original the value 'yes' is assigned. For an enumeration that is still supported but changed, 'acceptable' would be selected. If the enumeration is not supported anymore 'no' is chosen for this values.

The transformed values are '5' for 'yes', as this is the best possible result, '3' for acceptable and '1' for 'no'. We did not assign '0' to 'no', as it is not a knock-out criterion in our scenario.

Next to the transformation table we can see the results for every alternative for the one sample record we have in our fictional plan.

As an aggregation method 'Worst' is selected. If we would have more than one sample record and we have 'acceptable' for one record and 'yes' for every other record, the final value for this alternative would still be '3', as 'acceptable' is the worst result of all records for this alternative.

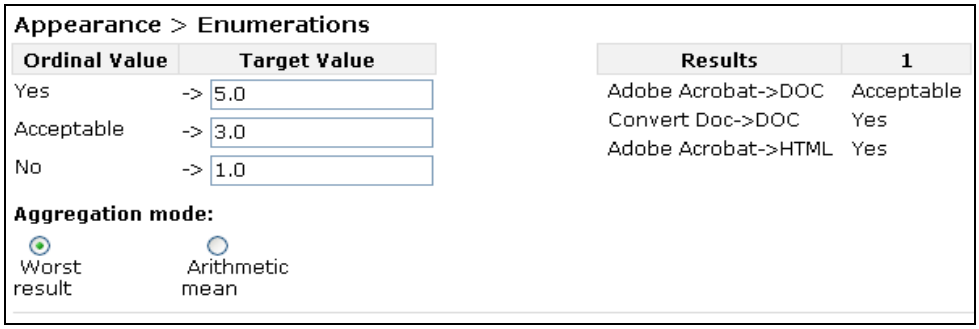


Figure 18 Ordinal transformation

Transformation of numeric values

This example shows how many MegaBytes per second an alternative is able to handle. The measured values are again shown next to the transformation table. '0.05' MByte/second is our best result, so we assigned '5' to this value. '0.01' was the worst result, so '1' is assigned. The other values are evenly distributed between these values.

'Steps' is selected for Threshold stepping, so if an alternative would be able to handle 0.019 MByte/second, the resulting value would still be '1'.

As Aggregation Method "Arithmetic Mean" has been selected. If we would have two sample records, 0.01 MByte/second for one of the records and 0.02 MByte/second for the other record the resulting transformed value would be the arithmetic mean of '1' and '2' -> '1.5'.

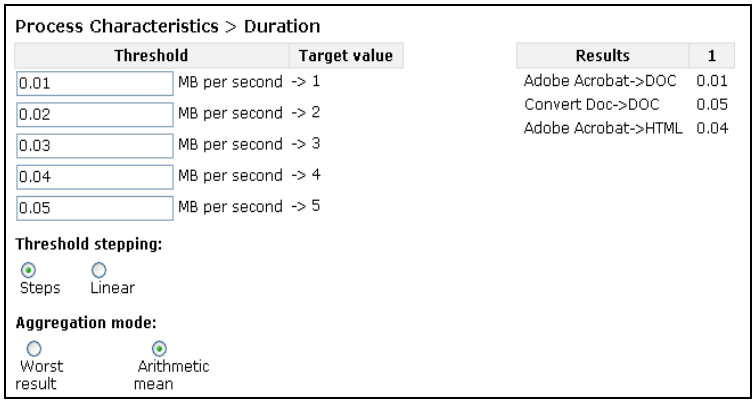


Figure 37 Transformation of numeric values

3.2 Set Importance Factors

This section explains the automatic weight balancing of objective trees in Plato.

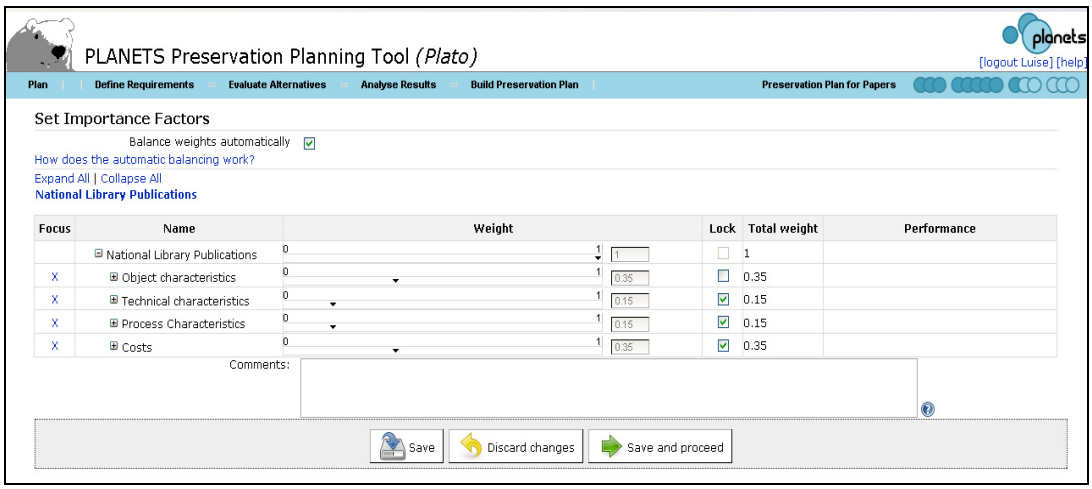


Figure 38 Set Importance Factors

When you reach this step the first time, Plato will automatically balance the relative importance of all nodes in the tree equally. Using the tree table editor, you can then set the relative importance factors of all siblings in a branch for all levels. The sum of weights on each level must always be equal to 1; this will be validated before you can proceed.

The column 'total weight' provides the overall influence of a node on the root value. For example, a node with weight 0.5 that is child of a node with weight 0,5 which is directly under the root will have a total weight of 0.5*0.5=0.25.

Automatic balancing

If the flag 'Balance weights automatically' is checked, Plato will activate an automatic balancing that relieves you of checking the sum yourself. Just use the sliders to increase weights of nodes that you deem important. The remaining weights will be balanced automatically. Each weight becomes locked once you adjusted it yourself, so that it is not rebalanced afterwards. This is indicated by the 'lock' column.

Example

1. Assume you have a root node with four nodes - A,B,C,D. When you reach the page, all weights are set to 0.25.

2. Now you adjust the weight of node A to 0.4 - the remaining weights are set to 0.2 each and node A is locked.

3. Now you adjust the weight of node D to 0.4. The effect is that node D becomes locked, too, while B and C are set to 0.1 each. Node A is not affected, because it is already locked.

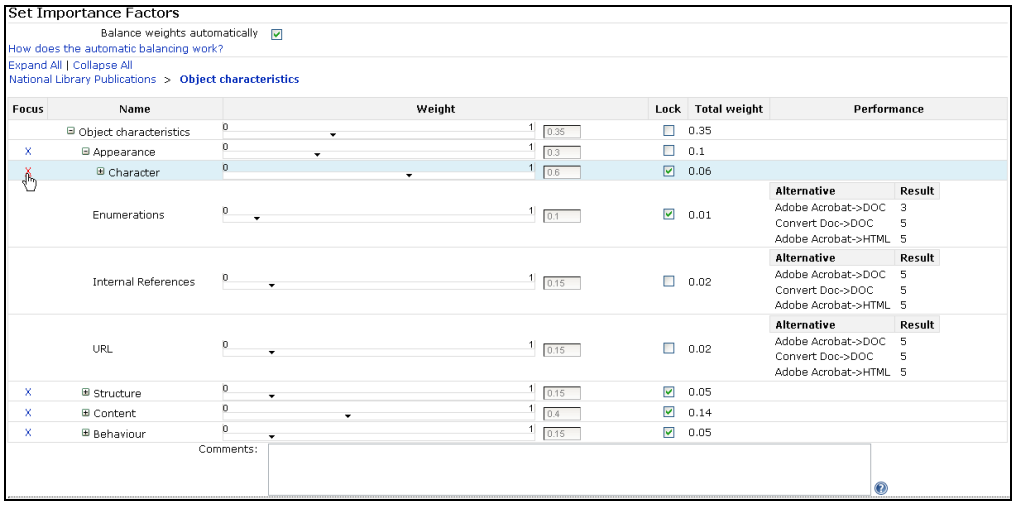


Figure 39 Set Importance Factors

3.3 Analyse evaluation results

The final step of the evaluation phase considers the complete evidence base of information produced during the previous phases of the workflow and analyses the performance of the candidate components in the experiment evaluation to arrive at a conclusion and recommendation for the best tool to employed, and the corresponding configuration. Alternatives are ranked by their evaluation values which are aggregated over the tree hierarchy using two different methods.

- Weighted multiplication is used to filter alternatives which exhibit unacceptable evaluation values at the criterion level, as these have been mapped to a target value of 0 during transformation and thus result in a total performance of 0.

- On the remaining alternatives, weighted addition is used to directly compare the performance of tools on all levels of the tree hierarchy.

The analysis and comparison of the alternatives considered can be guided significantly by a graphical visualization as provided by the planning tool described in Section 6. As a result of the evaluation, the preservation planner makes a decision and recommendation for a tool to be selected. The method allows for the selection of multiple components that are considered to be complementary. For example, many conversion tools for electronic documents have problems with entirely preserving the layout as it was displayed in the original environment, whereas migrating a document to an image loses the future potential for full-text search access. In some cases it might be desirable to combine both approaches and thus select multiple tools for the incorporation into a preservation

system. As an essential element of the recommendation, the reasons underlying it are documented, together with the expected effects of applying this strategy on the set of objects at hand. For example, it may be known that the easy *editability* of objects will be lost as a direct cause of converting them to a format such as PDF/A. As this might not be a requirement, or not be assigned significant weight, it might not influence the decision in a significant way. However this reasoning needs to be documented as part of the decision making procedure.

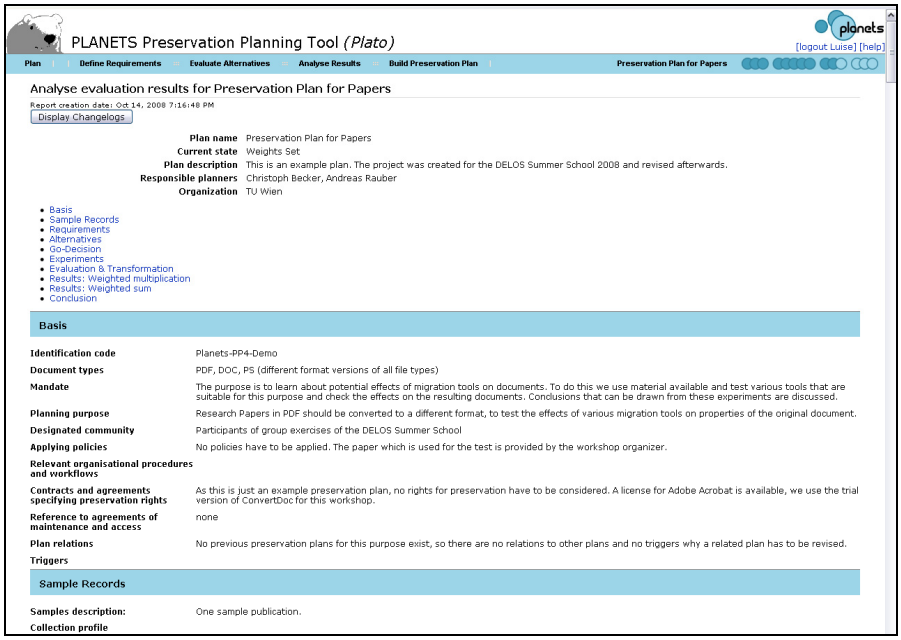


Figure 40 Analyse evaluation results for Preservation Plan for Papers

The change logs stored with the preservation plan can be displayed by clicking on 'Display Chang'logs'. The change logs are displayed next to the respective fields or group of fields that have been altered.

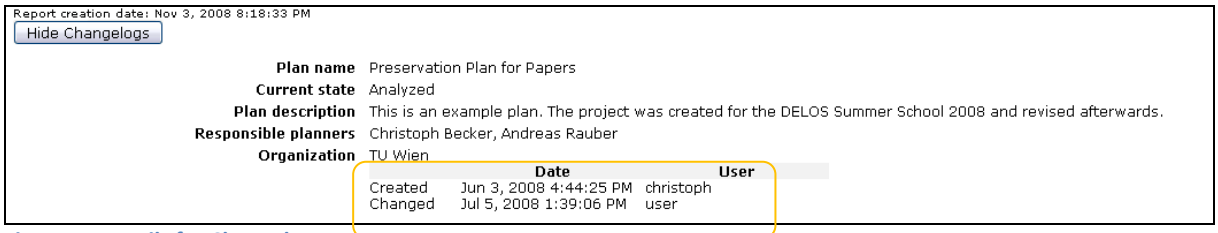


Figure 41 Details for Changelogs

4. Build Preservation Plan

In the fourth and final phase of the planning workflow, a preservation plan is created, based on the decision for a preservation action. It specifies a series of concrete steps or actions, along with organisational responsibilities, rules and conditions for executing the preservation action on the collection.

*A **preservation plan** defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection). The Preservation Plan takes into account the preservation policies, legal obligations, organizational and*

*technical constraints, user requirements and preservation goals and describes the preservation context, the evaluated preservation strategies and the resulting decision for one strategy, including the reasoning for the decision. It also specifies a series of steps or actions (called **preservation action plan**) along with responsibilities and rules and conditions for execution on the collection. Provided that the actions and their deployment as well as the technical environment allow it, this action plan is an executable workflow definition.*

4.1 Create Executable Plan

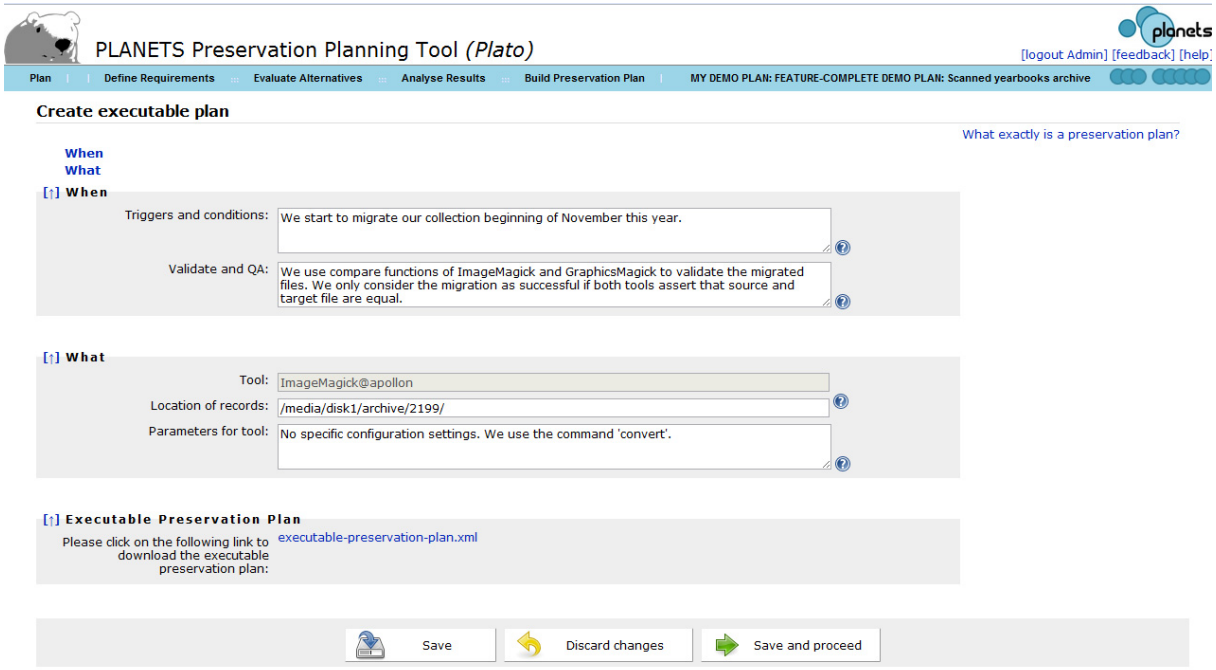


Figure 42 Create Executable Plan

This step of the workflow defines the triggers for the execution and the conditions under which the preservation action will be carried out. Hard- and software requirements as well as dependencies to other systems are documented. To enable the execution of the preservation plan, tool settings and details about the location of the collection on which the action is to be performed are defined. To quality assure the performed actions, a subset of the criteria used for evaluating solutions can be selected. These criteria will then be extracted using characterisation services on the objects after the performed action and validating that defined thresholds of these criteria are met. The necessary documentation that has to be recorded for performing the action is also defined in this step.

4.1.1 When

Triggers and conditions for execution Hard-and software requirements, other dependencies

Type	Examples
Triggers and conditions	We start to migrate our collection beginning of November this year.
Validate and QA	We use compare functions of ImageMagick and GraphicsMagick to validate the migrated files. We

	only consider the migration as successful if both tools assert that source and target file are equal.

Table 12 Examples for the boxes in Create Executable Plan / When

4.1.2 What

- Executable Preservation Plan (e.g. executable workflow, programme) that will execute the preservation action on the digital objects and automated mechanisms for validating results of preservation action [includes <ID>]
- Other actions needed (reporting/documenting…)

Type	Examples
Location of records	/media/disk1/archive/2199/
Parameters for tool	No specific configuration settings.

Table 13 Examples for the boxes in Create Executable Plan / What

4.1.3 Executable Preservation Plan

If you have chosen a suitable action as the alternative of choice, Plato will generate an executable preservation plan that you can deploy directly. You can download it as an XML as shown in the screenshot above.

4.2 Define Preservation Plan

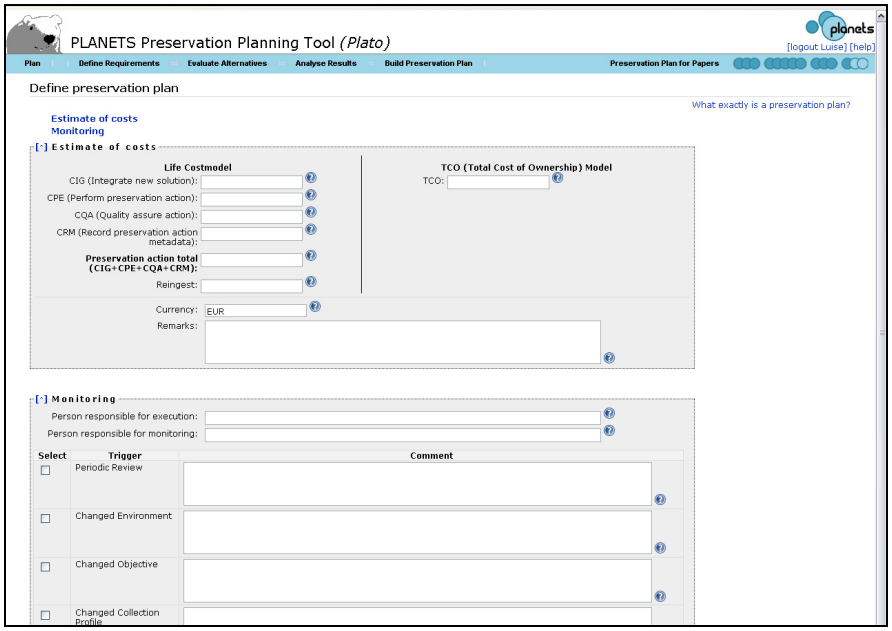


Figure 43 Define preservation plan

While most parts of the preservation planning workflow take care of the technical aspects of the preservation plan, this step mainly defines the organizational details.

Cost factors influence the decision on a specific alternative. In this step, a more detailed calculation of costs using an approved cost model is performed. Cost models that can be used are for example Life2 or the Total Cost of Ownership (TCO) model. While an estimate of the costs may be fine for evaluating the alternatives, the costs have to be determined as accurate as possible in this step.

The assignment of responsibilities is also documented in this step. Monitoring the process of applying the preservation actions has to be done by a different role than executing the preservation plan. It also has to be monitored if an event occurs that makes it necessary to reevaluate the plan. Possible triggers for this are either a scheduled periodic review, changes in environment such as new available tools detected through technology watch, changed objectives (changed target community requirements) or a changed collection profile (e.g. new objects in the collection). Another possible trigger is that certain thresholds on the evaluation are no longer met by applying the preservation action.

4.3 Validate Preservation Plan

Figure 44 Validate plan

In the final stage the whole documentation about the preservation plan has to be reviewed. The process of evaluating different alternatives and taking a decision for a recommended tool based on the evaluation and the creation of the preservation action plan, the documentation about the basic framework of the institution and parameters under which the plan is valid are verified again.

The validated plan has to be approved by the person responsible for approval. Once the plan is approved, no more changes on the plan can be done without revising the whole plan.