

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a preprint version which may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/34468>

Please be advised that this information was generated on 2021-03-05 and may be subject to change.

Touring the ICIS Publication Management System (PMS v1.2)

E.D. Schabell
erics@cs.ru.nl

Radboud University Nijmegen, Institute for Computing and Information Sciences,
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands

Abstract

The *Publication Management System (PMS)* was initially developed and deployed for usage by the *IRIS* department within the *Radboud University Nijmegen*. It was born from a wish to provide extensive services in managing and reporting our publications. This paper takes the reader through a tour of the current version of *PMS*, from the basic services available to any user, on to specific functionality for our institutes members, through the API and finally leaves the reader with some examples of how to use the more advanced features *PMS* provides.

1 Introduction

The *Publication Management System (PMS)* was initially developed and deployed for usage by the *IRIS* department within the *Radboud University Nijmegen*. It was born from a wish to provide extensive services in managing and reporting our publications, see [Schabell et al., 2005] for a listing of fundamental features that *PMS* provides.

The latest release of the *ICIS PMS* was completed at the end of September 2006. It was a milestone release that migrated the usage from a single department (*IRIS*) to the institutional level (*ICIS*). In the following sections we will provide the user with a tour of the currently deployed *ICIS PMS* portal. We will start in Section 2 with a look at the portal and the general tools provided to every visitor (no authorization required). We will continue on in Section 3 with a look at the specific tools available to *ICIS* users once they have logged into the portal. After this, we will point out some of the functionality that the *PMS API* provides for more technical users in Section 4. Finally, in Section 5 we will provide some examples using the described *PMS API* functionality. Hopefully this will provide the user with a good overview and ensure that she makes the most of the current *ICIS PMS*.

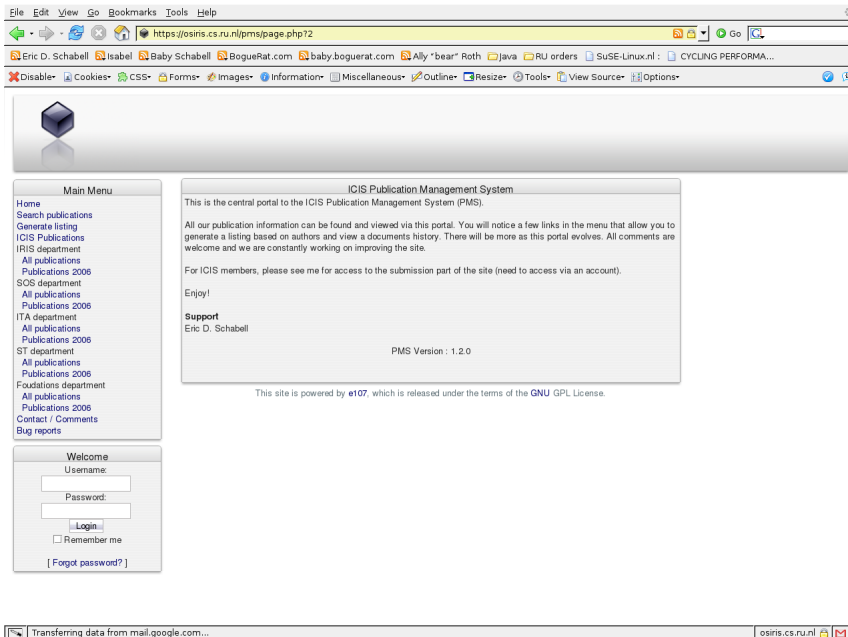


Figure 1: ICIS PMS portal

2 Basic tools

The tools discussed here can be found at the *ICIS PMS portal*¹ and are unrestricted in that we allow everyone to make use of them. We begin with the portal itself and walk the user through a list of the available tools she will find there.

2.1 Portal

The central entry point for the *ICIS PMS* can be located through the ICIS website² or by proceeding directly to the *ICIS PMS* site itself, see Figure 1. All relevant tools can be reached by their respective entries in the menu (top left). The user will initiate a given tool by clicking on the provided link.

2.1.1 Search publications

In order to provide for the browsing of our publications collection, a search function has been created. The user can search based on a publications title or abstract. The abstract search is a full text search. Upon submitting your search you will either be provided with a message that nothing was found or a list of publication keys that match your query. An example results set could look something like this for a query searching all titles matching - **Publication**:

| CITEKEY | TITLE | HISTORY |
|------------------------------|------------------------------------|----------------------|
| ===== | ===== | ===== |
| 2005-Schabell-Realization | IRIS Publication Management System | View |
| 1996-Systems-PublicationList | Publication List 1991-1996 | View |

The results shows two publication key links, their respective titles and two links that lead to the respective publications history (a log on the publications lifecycle within PMS). To view a

¹<http://osiris.cs.ru.nl/pms>

²<http://www.cs.ru.nl>

publications details the user can then click on the provided links.

2.1.2 Generate listing

This tool provides the user with a means to generate a listing of publications based not only on author names, but on specific research groups and even complete organizations. The user can make a selection from the presented pull-down menu of all author names as found in the *PMS*. After submitting her choice via the provided button, the user will be provided with two links. For example, if the user were to generate a listing based on the author named - **Schabell, E.D.**, she would be presented with the following two hyperlinks:

```
https://osiris.cs.ru.nl/pms/iris-diglib \
/src/report_publist.php?last=Schabell&first=E.D.&style=
```

```
https://osiris.cs.ru.nl/pms/iris-diglib \
/src/report_publist.php?last=Schabell&first=&style=}
```

The first hyperlink will show you all publications by the author, specifically using both the first and last names of the author. The second hyperlink will display a more generic publication list using only the authors last name as a filter.

2.1.3 ICIS technical reports

This item provides an overview of the *ICIS* technical reports. The view is split into years, with the current year being the default displayed upon viewing this item. Furthermore, there are diverse items listing the various departments and organizations that have requested some form of publication reporting. Most often you will see something that generates a bibtex formatted overview (links have format of: *Publications YYYY*). These are generated lists that can be directly imported into the *METIS* system as required by our institution.

2.1.4 Contact/Comments

As a way of directly contacting the author with comments or suggestions, this item is available for the users convenience. It should be noted that this is not the way to submit a feature request or report any bugs or mistakes the user might have found, see Section 2.1.5 for that.

2.1.5 Bug reports

This is the correct location to enter any feature requests or file any bugs that the user feels need some attention. This item takes the user to our *Trac*³ issue tracking site. If you wish to be kept informed of your issues status then please remember to replace the *anonymous* with a valid username or email address. Here you can view development activity via the **Timeline** tab, see our release **Roadmap**, even **Browse Source** or **View Tickets** to see what has already been submitted.

³<http://trac.edgewall.org>

3 ICIS member tools

As a member of *ICIS*, you will have a login that gives you access to some extended functionality within *PMS*. The first is integrated technical report submission process, the second allows you to submit non-technical report publications and the final one allows you to edit existing publications.

3.1 TechReport submission

The first and foremost change that was needed for this release was to provide for a seamless technical report submission process.⁴ Up to now, this process involved several steps for a writer of a technical report to actually obtain a technical report number for her publication. No more, this is now a one step submission process which results in not only a new technical report number but in the immediate availability of the writers publication online within the *ICIS PMS*.

The technical report submission form asks the user to provide the minimum data needed to create an entry in *ICIS PMS*:

- submitters naam
- research group
- authors
- annotation (optional)
- edit of the generated cite key (optional)
- publication content in PDF format (file upload)
- email address
- publication title
- note (optional)
- publication abstract

The technical report number is assigned automatically. A notification is sent to the departmental representative so that the proper paperwork for internal registration can be completed⁵, but the user can immediately view his publication entry and see the technical report number that has been assigned.

3.2 External submission

Outside of technical report submissions, it is possible to add non-technical report publications and publications that have not originated from our institute. This submission process starts with the user picking from the following list of available publication types:

- | | | |
|---------------|-----------------|-----------------|
| Book: | Chapter: | Conference: |
| - book | - inbook | - proceedings |
| - booklet | - incollection | - inproceedings |
| - inbook | - inproceedings | |
| - phdthesis | | |
| Educational: | Lecture Notes: | Report: |
| - manual | - manual | - booklet |
| - misc | - misc | - mastersthesis |
| - unpublished | -unpublished | - misc |
| | | - techreport |
| Professional: | Journal: | - unpublished |
| - article | - article | |
| - misc | | |

⁴It should be noted here that the technical report submission feature is the only feature of this system that is a requirement for all ICIS departments to use. This is now the only way to obtain a technical report number.

⁵It remains your responsibility to ensure that the proper registration paperwork is filled out by your departmental Management Assistant.

Each type will take the user through a step by step process of filling in the required and optional BiBTeX data elements for that specific type. It conclude with the optional possibility to upload the publications content in a PDF file.

It should be noted that it is possible to enter technical reports for our institute or departments that are missing from the current listings. This is how a user can get a technical report into *PMS* with an existing technical report number.

3.3 Edit publications

The editing of a publication begins with a search for the publication to be edited, as in Section 2.1.1. The user can search based on a publications title or abstract. The abstract search is a full text search. Upon submitting your search you will either be provided with a message that nothing was found or a list of publication keys that match your query. An example results set could look something like this for a query searching all titles matching - **Publication**:

| CITEKEY ===== | TITLE ===== | EDIT ===== |
|------------------------------|------------------------------------|---------------|
| 2005-Schabell-Realization | IRIS Publication Management System | Edit me! |
| 1996-Systems-PublicationList | Publication List 1991-1996 | Edit me! |

The results shows two publication key links, their respective titles and two links that lead to the start of editing the respective publications (follow the **Edit me!** link of a publication to start the editing process). To view a publication's details the user can then click on the provided links.

From this point onwards it is a two step process. First, you can edit the data as found in *PMS*. Upon proceeding to the next step by pushing the provided button at the bottom of the screen, the user will arrive at the second and last step in the editing process. Here the user is provided with an overview of her changes, if any, and may optionally upload new sources in a ZIP file and/or content in a PDF file.

Should you upload new files, they will replace existing files. The rest of the editing process provides the user with all the possibilities needed to change, update and even migrate a publication from one type to another.⁶

⁶Should you remove an actor (author, editor, organization, etc) from your publication, it will be removed. At the time of this writing there is no way to add a new actor (author, editor, organization, etc) to a publication other than contacting the administrator. This is a known issue and is scheduled for correction in the bug tracking system.

4 Api

For the more technical user of the *PMS* we will provide a short tour of the API and put some of the more advanced uses on display. We will present some of the functionality from the *DisplayManager*, *ReportManager* and *PubManager*, in that order. Finally, it should be noted that to make use of these API features will require access to the *PMS* database which is only possible by contacting the administrator. Please note that the provided code snippets are only for demonstrating the method calls, for more complete examples and usage of the API see Section 5.

4.1 DisplayManager

The *DisplayManager* class provides for displaying of the *PMS* data in HTML.

4.1.1 displayIdList

The most interesting method for an advanced user would have to be the *displayIdList* which will, when provided with an array of valid publication cite keys, deliver a formatted HTML page displaying the given publications. This method is defined statically, meaning that it is available through a single method call and no constructor call for the *DisplayManager* class as follows (code is excluded for filling the array with publication id's):

```
DisplayManager::displayIdList( $someArrayOfPublicationIds );
```

4.2 ReportManager

The *ReportManager* class is for all output formats, not just HTML, when returning data for usage from *PMS*. The two methods we will examine here are most likely to be the ones of most interest to advanced users. With these you can include publication information, in both a global and a more selective way, into any webpage. Both methods are not static and should be preceded by a constructor call for the *ReportManager*.

4.2.1 getHTMLPublications

The more global of the reporting methods, *getHTMLPublications* provides the more advanced user with the possibility to generate a list of publications in HTML format from a single user, a single research group, our entire institute, or every single publication in *PMS*. This can be augmented with a given year or the default of the current year will be used to produce a single year overview with links to the remaining years. The basic usage is as follows (only default values shown for the method call):

```
$manager = ReportManager();  
$manager->getHTMLPublications( $last='all', $first='', $year='' );
```

4.2.2 reportHTMLPubSingle

The most basic form of reporting and therefore probably giving the advanced user the most flexibility in designing her publications pages is the *reportHTMLPubSingle* method. It takes a single publication id as input and provides an overview of the publication data based on Bibtex type, the abstract if available, an eventual content download link, a Bibtex entry generation link, and an eventual URL link. The basic usage is as follows:

```
$manager = ReportManager();  
$manager->reportHTMLPubSingle( $singlePublicationId );
```

4.3 PubManager

This class is the central point of communication with the *PMS*, arranging requests for data between the various components and users as needed. Two essential methods will be discussed here to complete the advanced users toolbox, *getAllResources* and *getAllResourcesFromAuthor*.

This method is used to obtain a list of publication id's for a given author in the system. It simply requires the caller to supply the authors last name and optionally the first name. This method is defined statically, meaning that it is available through a single method call and no constructor call for the *PubManager* class as follows:

```
PubManager::getAllResourcesFromAuthor( $last , $first='' );
```

To obtain a list of publication id's, the *getAllResources* method will provide the advanced user with a tool that makes it easy to filter for different sets of id's. It takes a class of Bibtex publication, a year, and a research group name, all of which are optional variables. Should none be passed then the method will return the id's of all publications found within *PMS*.

It is possible to filter with one or more of the variables, to specify only a class of publication, a given year and/or a research group. This method is not static and should be preceded by a constructor call for the *PubManager* as follows (only default values shown for the method call):

```
$manager = PubManager();  
$manager->getAllResources( $class='', $year='', $group='' );
```

4.4 Digging deeper

For those users that would like to dig a bit deeper into the *PMS* project, there are several ways to obtain more information. Firstly, the entire project API is documented and the entire range of available services can be found online⁷. Secondly, you may look at the sources as they are freely available from the *CodeYard* repository:

Latest version:

```
svn co https://cvs.codeyard.net/svn/PMS/pms/src/trunk
```

⁷https://osiris.cs.ru.nl/pms/iris-diglib/src/diglib_classdocs

5 Some examples

After our discussion on the *ICIS PMS* API, we will provide some code examples for those users wishing to make use of these advanced features.

5.1 Display a single publication

A basic example is shown here for how it is possible to display a single publication. We start by deciding to allow the user to pass a single option, the publication id. To make our PHP webpage more flexible, we allow the user to also pass the publication id from the command line as our first argument. Taking a look at our code below, we see the first thing that is needed is to include two files, the database connection information and the *ReportManager*. We then proceed to determine if we are running as a webpage (pulling our publication id from the *REQUEST* array), or as a command line application (pulling our publication id from the arguments list). Finally we call our static method to display the single publication.

```
<?php
    include_once( 'dbconnect.inc.php' );
    include_once( 'ReportManager.php' );

    if ( $_REQUEST['id'] )
    {
        ReportManager::reportHTMLPubSingle( $_REQUEST['id'] );
    }
    elseif ( $argv[1] )
    {
        ReportManager::reportHTMLPubSingle( $argv[1] );
    }
?>
```

5.2 Display your publications

The easiest way to include a listing of your publications from *PMS* in your personal website for example would be to provide a link to the URL (this example is using the author name):

```
https://osiris.cs.ru.nl/pms/iris-diglib \
    /src/report-publist.php?last=Schabell&first=E.D.
```

If you would rather format a page yourself you can do something like the following example to personalize the results to your own sites look and feel. Taking a look at our code below, we see the first thing that is needed is to include two files, the database connection information and the *ReportManager*. Next we ensure that we get the current years listings as our default starting point. Finally, we call the *getHTMLPublications* with our name to produce a nice listing. You can include your own style sheet information somewhere in this page to create your look and feel.

```
<?php
    include_once( 'dbconnect.inc.php' );
    include_once( 'ReportManager.php' );

    $year = date( 'Y' );
    ReportManager::getHTMLPublications( "Schabell", "E.D.", $year );
?>
```

5.3 Add publications to project page

The following code illustrates how one can include individual publications into a project website, it shows how you can embed PHP into your HTML files. We start by presenting a *Results* heading for the list of publications we want to display. Next we need to include our database connection

information and the *ReportManager* method we want to use. Finally we setup our HTML list and display two publications using *reportHTMLPubSingle*.

```
<h3>Results:</h3>

<?php
    include_once( "dbconnect.inc.php" );
    include_once( "ReportManager.php" );
?>

<ol>
    <li><?php ReportManager::reportHTMLPubSingle( "2004-VanGils-TransSel" ); ?>
    <li><?php ReportManager::reportHTMLPubSingle( "2004-Bommel-MarketChallenges" ); ?>
</ol>
<hr>
```

5.4 Display technical reports

For a more extensive example that makes use of more of the API, we decide to display the institutes publications, but only the technical reports for all available years. For simplicity, we will leave out the simple HTML formatting tags and try to present only the necessary code for completing this example. We start deciding to allow any caller of this page to supply a personal style sheet, so we check for this and set it to our default style sheet should none be provided. Next we print out a header for the list to be generated and then include our needed database connection information. We also include several classes we will need, including the *MysqlDB* as we wish to run a customized query later in the code. Next we setup our variables by picking up any passed options and initialize our select query. Then we gather in our array of technical report id's and run the select query to get a listing of years in which our institute has published technical reports (this will be used to create the hyperlinked years that allow a user to click between individual year listings). The gathered query results is then used to print out the hyperlinked years at the top of the page. Finally we sort the technical report id's we found previously in the order of most recent number first and then display the requested years technical reports.

```
<?php

if ( strlen( $_REQUEST['style'] ) == 0 )
{
    print '<link type=\'text/css\' rel=\'stylesheet\' ';
    print 'href=\'http://osiris.cs.kun.nl/iris/web-docs/style.css\'>';
}
else
{
    print '<link type=\'text/css\' rel=\'stylesheet\' ';
    print 'href=\'\' . $_REQUEST['style'] . \'>';
}

print "<center><h2>ICIS Technical Reports:</h2></center>\n";

include_once( 'dbconnect.inc.php' );
include_once( 'PubManager.php' );
include_once( 'ReportManager.php' );
include_once( 'MysqlDB' );

// Variables.
$year          = $_REQUEST['year'];
$class         = $_REQUEST['class'];
$group         = $_REQUEST['group'];
$pubMgr        = new PubManager();
$report        = new ReportManager();
$yearListSelect = "select DISTINCT year from resource where class='Report' ";
$yearListSelect .= "AND ( research_group='icis' OR research_group='sos' OR ";
$yearListSelect .= "research_group='iris' OR research_group='fnds' OR ";
```

```

$yearListSelect := "research_group='st' OR research_group='ita' ) ";
$yearListSelect := "ORDER BY year;";

if ( strlen( $class ) == 0 ) { $class = "Report"; }
if ( strlen( $year ) == 0 ) { $year = "2006"; }
if ( strlen( $group ) == 0 ) { $group = "icis"; }

$myIds = $pubMgr->getAllResources( $class , $year , $group );
$resultsYearList = MySQLDB::runSelect( $yearListSelect );

// print year overview.
foreach( array_reverse( $resultsYearList ) as $target )
{
    if ( $year == $target->year ) { print "$year "; }
    else
    {
        print "<a href='" . ICIS_TR . " $class&year=" . $target->year;
        print "&group=$group&style=";
        print "http://osiris.cs.kun.nl/iris/web-docs/style.css'>";
        print $target->year . " </a>";
    }
}

// sort by tr number and display.
//
$myNumbers = array(); // will hold numbers pointing to id's.
$sorted = array(); // used to sort my tech numbers.
foreach ( $myIds as $item )
{
    $row = MySQLDB::runSelect(
        "SELECT tech_number FROM techreport WHERE resource_id='$item'";
    );
    list( $pre , $number ) = split( "-", $row[0]->tech_number );
    $myNumbers = array_merge( $myNumbers , array( $number => $item ) );
    $sorted = array_merge( $sorted , array( $number ) );
}
sort( $sorted );
$sorted = array_unique( $sorted );

print "<h2>$year</h2>\n";
foreach ( $sorted as $index )
{
    print $report->reportHTMLPubSingle( $myNumbers[$index] ) . "\n";
}

?>

```

As you can see this is a rather extensive manipulation of the various API tools provided can lead to almost any form of publication display that one could dream up.

5.5 Comments on generating listings

The alert reader would have noted that there were a few unexplained options in the hyperlinks presented in section 2.1.2. These are further detailed here to provide an advanced user with some tips to produce group or institutional listings and optional styling adjustments:

- last - can be the last name of an author, the name of a research group or icis as an institution.
- first - the authors first name, or empty if selecting a group or institution.
- style - can be empty or point to the URI of a valid style sheet to match your websites look and feel.

Here are a few example uses of the above options:

Displays organizations publications:
`https://osiris.cs.ru.nl/pms/iris-diglib \`
`/src/report_publist.php?last=icis&first=&style=`

Displays research group IRIS publications:
`https://osiris.cs.ru.nl/pms/iris-diglib \`
`/src/report_publist.php?last=iris&first=&style=`

Displays all publications in PMS:
`https://osiris.cs.ru.nl/pms/iris-diglib \`
`/src/report_publist.php?last=&first=&style=`

Displays IRIS publications with style sheet from IRIS website:
`https://osiris.cs.ru.nl/pms/iris-diglib \`
`/src/report_publist.php?last=iris \`
`&first=&style=http://osiris.cs.kun.nl/iris/web-docs/style.css`

These Uri's can easily be included in any website to integrate *PMS* publication reporting with existing sites.

6 Looking back

This concludes our tour through the current version of *PMS*. We have take you through the *ICIS PMS* portal, showing the various tools provided to both general users and to institute members. We have take a look at the *PMS API*, detailing some of the methods and services that could be of interest to a more advanced user. We concluded our tour by providing some extensive examples with actual code to demonstrate our claim of having a flexible publication management API.

As we have shown, *PMS* is rather flexible, enabling you to both store, present and manage your publications within a easy to use framework. It is possible to use the provided *ICIS PMS* portal links to display publications or to make use of the *PMS API* to generate almost any form of publication output one might need. It is our hope that these examples demonstrate that *PMS* is, as Wally likes to put it, "Prettier than a skunk sandwich and cooler than a hobo's mittens." [Adams, 2006]

References

- [Adams, 2006] Adams, S. (2006). *Dilbert.com - Strip of the Day*. Dilbert.com, <http://www.unitedmedia.com/comics/dilbert/archive/images/dilbert2006111110504.gif>. last checked: Nov 2006.
- [Schabell et al., 2005] Schabell, E., Proper, H. E., and van der Weide, T. (2005). IRIS Publication Management System – the first steps towards realization. Technical Report ICIS–R06008, Radboud University Nijmegen, Nijmegen, The Netherlands, EU.