

```
only need subs from MRtoolbarLib.pl" if ($?)
check to see if javascript
checkForTmpFileDeletion();
StartIt(); &DisplayHash(\
open DEBUG, ">debug.txt";
\n");
rname = $in('USER');
ctID = $in('PROJECTID');
```

FootPrints “Under The Hood”: Supporting and Extending FootPrints In-House

Presented by:
Mark Montague
The University of Michigan
markmont@umich.edu

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Supporting

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Supporting



FootPrints “Under the Hood” Part I: Supporting FootPrints In-House

Presented by:
Mark Montague
The University of Michigan
markmont@umich.edu



Introduction

- **Content:**
 - How FootPrints works internally.
 - How to support and troubleshoot FootPrints yourself without calling UniPress.
 - Part II, “Extending FootPrints In-House”:
10am Tuesday in Tech Forum 2.
- **Audience:**
 - System administrators.
 - Some familiarity with Perl.
 - Some knowledge of CGI technologies.
- **Questions: At the end or during the “Birds of a Feather” session Tuesday at 12:15pm.**



Platform assumptions

FootPrints	Version 7.0c. Material in this presentation may not apply to other versions.
Operating system	Linux/Unix. Shell commands will need to be changed for Microsoft Windows servers, but everything else covered in this presentation should remain applicable.
Database	Any SQL-based database supported by UniPress for use with FootPrints. Some material in this presentation will not be applicable to organizations using the FootPrints internal (GDBM) database.
Web server	Apache. But most material in this presentation is independent of which web server is used.



Transparency

- FootPrints “does not require any programming, database administration, or consulting”. But...
- FootPrints is written almost entirely in Perl.
(See <http://www.perl.org/>)
- Benefits for organizations that want to go “under the hood”:
 - **Openness / transparency:** knowing how FootPrints works enables you to use it more effectively.
 - **Immediacy:** in-house support can be much quicker (minutes/hours instead of hours/days).
 - **Specificity:** get exactly what is needed; leverage knowledge of how the rest of your organization operates.
 - **Flexibility:** support FootPrints in-house when it’s appropriate, refer other support issues to UniPress.



WARNING!

You can get yourself into trouble very quickly with the material in this presentation. Although UniPress provides excellent technical support, the material covered in this presentation goes beyond what they provide as standard support.

Do everything on a separate test server first. Only repeat it on your production FootPrints server after it has been tested thoroughly, proven safe, and you are confident you know what you are doing.

Always make backups of both your filesystem and database before each change so you can go back to a working installation should you inadvertently break your server. (See pages 470 - 472 of the FootPrints 7.0 Reference Manual). In the worst case, be prepared to reload your server and re-install FootPrints.

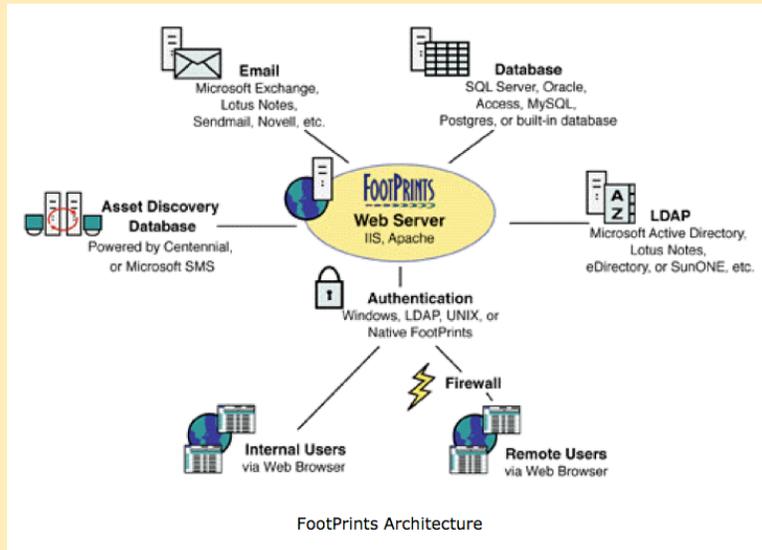


Support methodology

- Problems are not common, but FootPrints is large and complex.
- How FootPrints is configured and used varies tremendously between organizations.
- Since we can't predict more than a small fraction of support needs, our strategy is:
 - Develop a good knowledge of how FootPrints works internally.
 - Develop investigative skills (we'll give a few general hints and examples, but this is mostly up to you).

Architecture (generic)

(from page 463 of the FootPrints 7.0 Reference Manual)

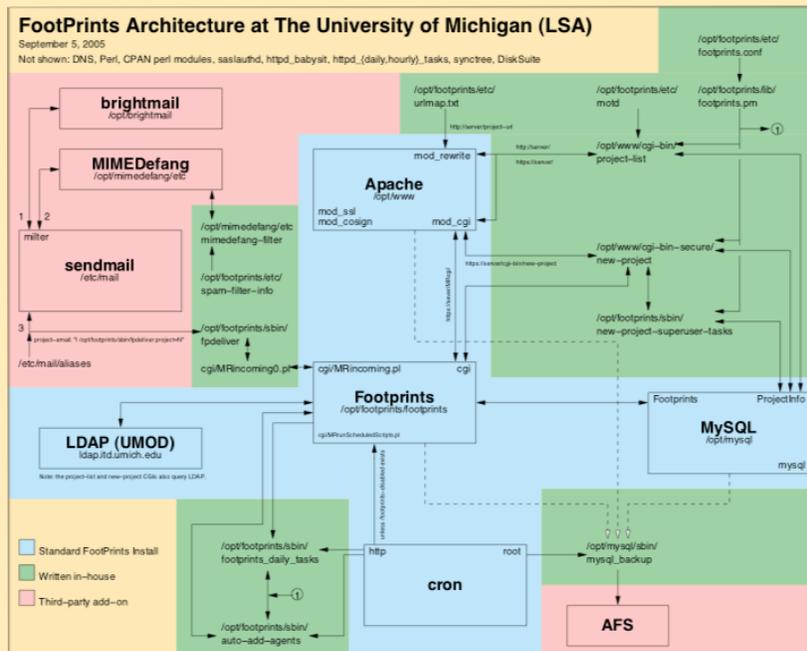


Architecture (example)

FootPrints Architecture at The University of Michigan (LSA)

September 5, 2005

Not shown: DNS, Perl, CPAN perl modules, saslauthd, httpd_babyait, httpd_(daily, hourly)_tasks, synctree, DiskSuite



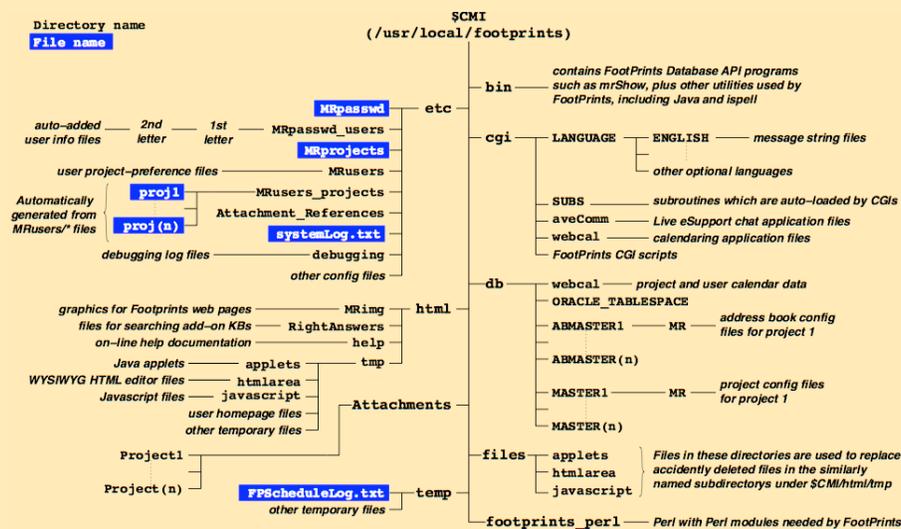


Filesystem structure

- FootPrints is usually installed in `/usr/local/footprints`
- At the University of Michigan, `/usr/local` is often on a fileserver, so FootPrints gets installed in the directory `/opt/footprints/footprints`
- FootPrints finds out where it is installed by looking at the variable `$CMI` in the file `$CMI/cgi/MRlocalDefs`
 - In a default installation of FootPrints, this file is `/usr/local/footprints/cgi/MRlocalDefs`
 - Near the top of the file is usually a line such as:
`$CMI = "/usr/local/footprints";`
- `$CMI` is used frequently in this presentation to mean “the directory where FootPrints is installed”.

Filesystem structure

(see also page 462 of the FootPrints 7.0 Reference Manual)





Metacharacter escaping

- Special characters in data such as ‘:’, ‘/’, or spaces can cause ambiguity, conflicts, or security problems with
 - Perl (during evaluation or string interpolation)
 - Databases (particularly when using unbound SQL parameters)
 - File formats (file separators)
- **FootPrints’ solution:**
 - “Fix Fields”: Replace special characters with three character codes before storing the data.
 - Data is usually stored in fixed form.
 - “Unfix fields”: Replace the three character codes the original special characters before using the data.



Metacharacter escaping

- Each metacharacter escape sequence (three character code used by FootPrints) starts with two underscores to indicate that it’s actually a special character that has been fixed.
- The third character is a letter that indicates what the special character actually is. Examples:
 - “ **b**” is a space (“blank space”).
 - “ **a**” is an apostrophe.
- Example: the project name “Mary’s Test Project” gets stored in FootPrints as “Mary **as** **b**Test **b**Project”.



Metacharacter escaping

Code	Character	Code	Character
<code>__b</code>	space	<code>__f</code>	/ (forward slash)
<code>__a</code>	' (single quote)	<code>__F</code>	\ (backslash)
<code>__q</code>	" (double quote)	<code>__Q</code>	?
<code>__t</code>	` (single backquote)	<code>__e</code>]
<code>__m</code>	@	<code>__E</code>	[
<code>__d</code>	. (period)	<code>__g</code>	>
<code>__u</code>	- (dash)	<code>__G</code>	<
<code>__s</code>	;	<code>__B</code>	!
<code>__c</code>	:	<code>__W</code>	{
<code>__p</code>)	<code>__w</code>	}
<code>__P</code>	(<code>__C</code>	=
<code>__3</code>	#	<code>__A</code>	+
<code>__4</code>	\$	<code>__I</code>	(vertical bar)
<code>__5</code>	%	<code>__M</code>	, (comma)
<code>__6</code>	^	<code>__Ux_</code>	Unicode character with value 'x'
<code>__7</code>	&		
<code>__8</code>	*		
<code>__0</code>	~ (tilde)		



File formats - YAML

- FootPrints stores some information as YAML files.
- YAML is a data serialization format -- it translates Perl variables into human readable text, and vice-versa.
- YAML permits FootPrints to read and write configuration files with a single line of Perl code.
- Example: `$CMI/db/MASTER1/MR/LogoSettings.txt`

```
--- #YAML:1.0
FILE_NAME: umbskresgeloگو.jpg
LOGO_CLICKABLE: ''
SEARCH_FRAME_LOGO_URL: http://www.bus.umich.edu/KresgeLibrary
```

- For more information, see <http://www.yaml.org/>



Configuration files - `$CMI/cgi/MRlocalDefs`

- `$CMI/cgi/MRlocalDefs` is a fragment of a Perl script.
- Read by every FootPrints CGI script.
- Contains variables that control FootPrints' behavior -- usually site-specific things which most customers won't need (and hence which are not configurable via FootPrints' web interface).
- Less important now than it used to be -- UniPress has added web pages to control many things that used to require editing `MRlocalDefs`.



Using `MRlocalDefs`

- Tables 1 - 6 in the handout contain tables of most `MRlocalDefs` variables.
- Some `MRlocalDefs` variables may be obsolete or restricted to use in specific environments.
- Before editing `MRlocalDefs`, read the FootPrints source code to find out how to properly use the variable (how to set it, what values are legal, etc.). For a list of the files to check under Linux/Unix, run

```
find /usr/local/footprints/cgi -name '*.pl' -print |  
xargs grep -l 'EMAIL_MARKER_LINE'
```

(leave out the leading `$`, `@`, or `%` character from the variable's name when searching).



MRlocalDefs examples

- Send everyone to `http://www.example.com/support` when they log out, except members of project 3, who should be sent to `http://www.example.com/finance` instead:

```
$ALTERNATE_LOGOUT_URL    = 'http://www.example.com/support';  
$ALTERNATE_LOGOUT_URL{3} = 'http://www.example.com/finance';
```

- Add a button to the left toolbar that lets users create issues in the “Building Maintenance” project (project 5) without needing to switch to it first:

```
$CREATE_IN_OTHER{5} = 'Maintenance Work Order';
```



Configuration files - \$CMI/etc

- The `$CMI/etc` directory contains data files used by FootPrints, usually things that are not specific to any one project:
 - Project option information (for all projects)
 - User account information and preferences
 - License information
 - Debugging log files
- Table 7 if the handout contains a list of the files in `$CMI/etc` and a brief description of each one.
- Note that some files may not be listed in the table, some files may be restricted to specific environments, and some information may be outdated or incorrect -- use with care.



Configuration files - \$CMI/etc/MRprojects

- \$CMI/etc/MRprojects stores project options and preferences.
- Text file - each line contains information for a single project. Fields within a line are separated by colons.
- Note: a lot of project configuration, particularly anything related to project fields, is in the directories \$CMI/db/MASTER_x/MR (where _x is the project number), not \$CMI/etc/MRprojects
- Looking in MRprojects is a quick way to check how a project is configured without adding yourself to it.
- Sometimes it's faster or easier to edit MRprojects than to use the web interface, especially when changing the same thing in multiple projects.



Configuration files - \$CMI/etc/MRprojects

Example line from \$CMI/etc/MRprojects (wrapped on to multiple lines for readability):

```

project number project database directory project admin email project name
4:/opt/footprints/footprints/db/MASTER4:ITCS UMCE User Services
Test:0:0:0:consulting.tools@umich.edu:0:0:
(status = Referred or status = Closed) and priority >= 1
::0:10:1:4:::::1:0:0:1:Username:ASSIGNEESANDADMINS:0:::::Closed:
:-1:0:0:1:_REQUEST_:0:1:0:C:::1:ON:1:1::::Open
address book number criteria for sending mail address book auto status for email
primary key reply to closed ticket
    
```

Table 8 in the handout contains a list of all MRprojects fields with a brief description of each one.



Configuration files - `$CMI/etc/MRprojects`

- Find out a project's project number:

```
grep -i "Help Desk Requests" /usr/local/footprints/etc/MRprojects
```

- List the project administrator's email address for each project:

```
perl -an -F':;' -e 'printf("%-50s %s\n", $F[2], $F[6]);' \
  < /usr/local/footprints/etc/MRprojects
```



Configuration files - `$CMI/etc/MRpasswd` and `$CMI/etc/MRpasswd_users/*/ */*`

- `$CMI/etc/MRpasswd` stores information about regular (non-auto-added) users and their non-project-specific preferences.
- `$CMI/etc/MRpasswd_users` is a directory tree that contains files with information about auto-added users and their non-project-specific preferences.
 - For scalability reasons, each auto-added user has their own file.
 - Files are stored under a double-layer directory hash based on the first two letters of the user's username.
 - An auto-added user with the username "alice" would have her information in the file
`$CMI/etc/MRpasswd_users/a/l/alice`
- Text files - each line contains information for a single user. Fields within a line are separated by colons.



Configuration files - \$CMI/etc/MRpasswd and \$CMI/etc/MRpasswd_users/*/*/*

Here is a sample line from \$CMI/etc/MRpasswd (wrapped on to multiple lines for readability):

```

username  encrypted password  user type  real name  default project  email address
markmont:qO.M8iXGj019w:4:Mark Montague:1:markmont@umich.edu:
0:::1:::SHOW::DISABLED:0:auto:ON:0:OFF:xsmall:show::OFF::
applet
checkbox method  spell check preferences  toolbar static/dynamic  WYSIWYG editor  font size
    
```

Table 9 in the handout contains a list of all MRpasswd fields with a brief description of each one.



Configuration files - \$CMI/etc/MRusers/

- Each non-auto-added user on the system has a file in the \$CMI/etc/MRusers directory.
 - The name of the file is the same as the username of the user.
 - The project preferences for a user with username “jonsmith” would be in the file \$CMI/etc/MRusers/jonsmith
- Text file - each line contains information for a single project. Fields within a line are separated by colons.
- All auto-added users share a common set of project preferences:
 - \$CMI/etc/MRusers/sharedForAutoAddedUsers
 - Auto-added users cannot customize their project preferences unless one of the project’s agents first converts them into a regular user.



Configuration files - \$CMI/etc/MRusers/*

IMPORTANT!

- If you make any changes to the files in the \$CMI/etc/MRusers directory, you also need to make changes to the files in the \$CMI/etc/MRusers_projects directory.
- What changes to make and why will be covered in a few slides.



Configuration files - \$CMI/etc/MRusers/*

Here is a sample line from \$CMI/etc/MRusers/markmont (wrapped for readability):

```

project number  user type  # issues on homepage  frames/non-frames  homepage view
1:2:20:0,1:mr:UNUSED:UNUSED:UNUSED:UNUSED:UNUSED:UNUSED:0:0::S::
My__bwork::priority,assignees,date,status,title::
project__administrator__bdefault:0:CUSTOMER
homepage saved search  user's role  prefill ticket contact info  homepage field layout
    
```

Table 10 in the handout contains a list of all MRusers fields and a brief description of each one.



Configuration files - \$CMI/etc/MRusers_projects/*

- FootPrints often needs a list of all users who are in a particular project.
- To generate this list, FootPrints reads all of the files in the \$CMI/etc/MRusers directory and notes which users have lines for the project in question in their files.
- Since it's not efficient to read all of the project preferences for all users each time to figure out who is in the project, FootPrints stores the results in \$CMI/etc/MRusers_projects/proj_x where x is the project number of the project in question.
- The next time FootPrints needs to know what users are in the project, it'll check this file first before re-reading all of the information in all of the files in \$CMI/etc/MRusers



Configuration files - \$CMI/etc/MRusers_projects/*

- **IMPORTANT:** Any change you make to files in the directory \$CMI/etc/MRusers_projects will not be permanent -- the next time a user changes one of their preferences for the project, or a user gets added to or removed from the project, the project's file in this directory will be removed.
- **IMPORTANT:** Changes you make to files in \$CMI/etc/MRusers won't take effect until you remove the \$CMI/etc/MRusers_projects files that are associated with the lines you changed in the MRusers files.
 - For example, if you edit \$CMI/etc/MRusers/alice and change Alice's preferences for projects 1 and 3, you should remove the files \$CMI/etc/MRusers_projects/proj1 and \$CMI/etc/MRusers_projects/proj3



Configuration files - User types

- Lines in `$CMI/etc/MRpasswd`, `$CMI/etc/MRusers/*`, and `$CMI/etc/MRusers_projects/*` contain fields for user type.
- Before FootPrints 6.0, possible user types included 0, s, S, a, A, 1, U, e, E, 2, 3, and 4.
- Starting with FootPrints 6.0, user roles take the place of most of the user types. Each user has a role which specifies what they can and cannot do in a project. This reduces the valid user types to:
 - 0 shared customer
 - A unique customer
 - 2 agent
 - 4 system administrator



Configuration files - `$CMI/db/MASTER*/MR/*`

- Each project has a directory `$CMI/db/MASTERx/MR` (where x is the project number).
 - Example: `/usr/local/footprints/db/MASTER12/MR`
- This directory contains the project's "configuration", particularly project field information, field dependencies, escalations, and so on.
- (Project "options" or "preferences", in contrast, tend to be kept in `$CMI/etc/MRprojects` or in other files in the `$CMI/etc` directory -- although this can often be somewhat of an arbitrary distinction.)
- Table 11 in the handout contains a list of files in the `$CMI/db/MASTER*/MR` directories with a brief description of each one.



Configuration files - \$CMI/db/ABMASTER*/MR/*

- Each address book has a directory \$CMI/db/ABMASTER_x/MR (where x is the address book number).
- This directory contains the address book's configuration, particularly address book field information, LDAP configuration, and so on.
- Table 12 in the handout contains a list of files in the \$CMI/db/ABMASTER*/MR directories with a brief description of each one.



Configuration files - \$CMI/db/*MASTER*/MR/Schema

- The schema file is the most important file in the \$CMI/db/MASTER_x/MR and \$CMI/db/ABMASTER_x/MR directories.
- This file contains the definitions for the project fields and address book fields, respectively.
- Each line in these files has five fields:

Field number	Description
0	Project or address book field name
1	Type (char, enum, date, http, or the name of a field of type enum)
2	Permissions (not used if the field has advanced permissions -- roles are used instead)
3	Not used (always zero?)
4	More type information (multi, time, datetime, check, tracking)



Configuration files - \$CMI/db/*MASTER*/MR/Schema

A sample schema file for a project:

```
Printer__bmodel866 deleted -1
Machine__bname char 88 0
Operating__bssystem Operating__bssystem_choices 88 0
Operating__bssystem_choices enum Microsoft__bWindows__b2000
Microsoft__bWindows__bXP Microsoft__bWindows__b_Pother__p
Solaris__b2__d5__d1 Solaris__b8
MacOS__b10__d3__b_PPanther__p MacOS__b10__d4__b_PTiger__p
Linux Other
Public__bticket Public__bticket_choices 88 0
Public__bticket_choices enum Yes (No)
Follow__uUp__bDate date 0 0
Submitted__bviaChoices enum Web Email Other Telephone
Submitted__bvia Submitted__bviaChoices 6 0 tracking
```



Database access

- FootPrints stores all issue data in a database (except for attachments).
- You can access this database using standard database access tools (e.g., the “mysql” command line database shell program, Crystal Reports, etc.)
- The database name is “Footprints” (note the lowercase ‘p’).
- FootPrints uses the database user “FP” to access the database.
 - You can create additional users and grant them access to the database and its tables using your database software’s administration tools.



Database access

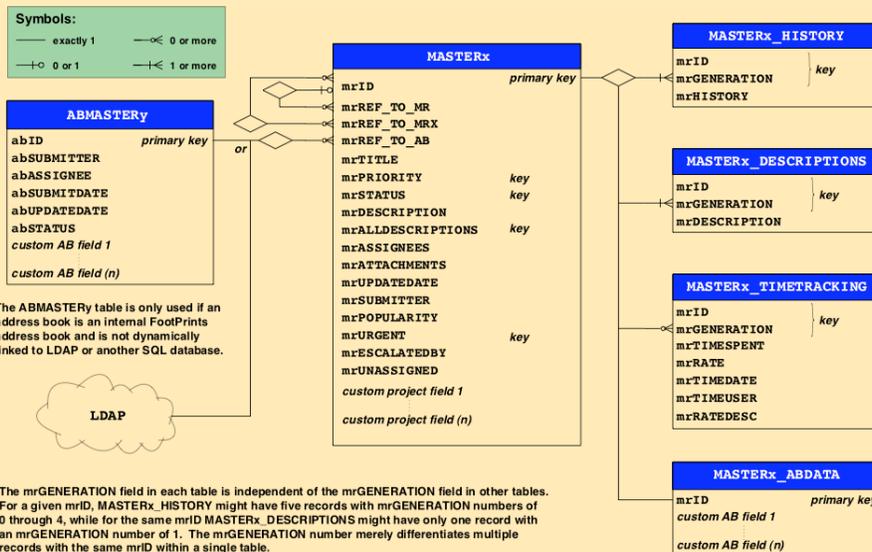
- The default database password for the FootPrints database user “FP” is easy to guess. Therefore, be sure to do at least one of the following:
 1. Run the database on the FootPrints web server machine and make sure it cannot be accessed at all from the network (e.g., start MySQL with the "--skip-networking" option). And/or,
 2. Configure the database server machine’s firewall rules to strictly limit access to the database to the FootPrints server machine. And/or,
 3. Lock down the database permissions so the database cannot be accessed remotely. For example, if you are using a MySQL database, run the following commands:


```
mysql -u root -p
DELETE FROM mysql.user WHERE user = 'FP' AND host = '%';
DELETE FROM mysql.db WHERE User = 'FP' AND Host = '%';
quit;
```

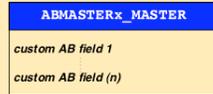
 ...and then do a full audit to determine what users can still access any tables, and from where. And/or,
 4. Change the password and put it, encrypted, into \$CMI/etc/MRDBpasswd

Database schema - Project and address book

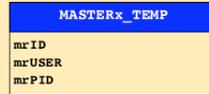
See pages 463 - 468 of the FootPrints 7.0 Reference Manual for descriptions and types.



Database schema - Miscellaneous tables



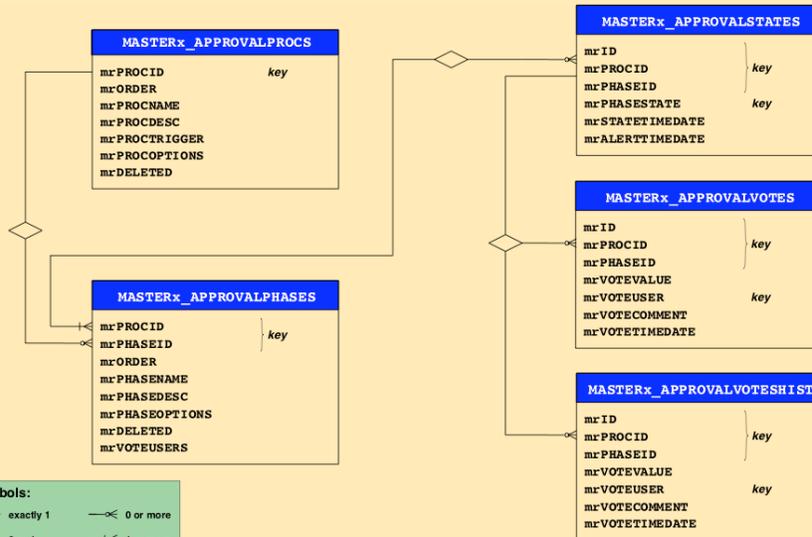
Master Contact Record table



Temporary table used for JOINS

Database schema - Change management

See pages 463 - 468 of the FootPrints 7.0 Reference Manual for descriptions and types.





Using the database directly

- The database can be queried directly to get information, bypassing the FootPrints web interface.
- Example: FootPrints doesn't currently allow agents to do an advanced search via the web based on who is on a issue's "CC:" list. The following query will show all issues in a project for which the address "alice@example.com" has been carbon copied:

```
SELECT mrID, mrASSIGNEES from Footprints.MASTER2 where  
mrASSIGNEES like '%CC:alice@example.com%';
```



Using the database directly

- Example: A web page that queries FootPrints' database directly in order to make service statistics available to people outside of FootPrints:

https://mailmove.lsa.umich.edu/csg-fp-cgi/csg_stats.pl

See Example 1 in the handout for a minimalistic example of code to produce such a web page.



Managing attachments

- Starting with FootPrints 7.0, files attached to issues are no longer stored in `$CMI/html/tmp/Attachments` and hence can no longer be requested directly through the web server (bypassing FootPrints) by guessing the project number and attachment filename.
- Attachments are now accessed via `$CMI/cgi/MRdownloadAttachment.pl`
 - Ensures the user has logged in.
 - Will only provide the attachment for an issue if the user has the permission to view the issue details.
 - Better management of MIME types and the download process.
- Where attachments are stored can now be configured under **Administration -> System -> Attachments** or the text file `$CMI/etc/AttachmentsRoot`



How the FootPrints CGIs work

- Users normally log in to FootPrints by going to the URL `http://server.example.com/footprints`
- This serves the file `$CMI/html/index.html`
- ...which redirects the user to `http://server.example.com/MRcgi/MRentrancePage.pl`
- ...which corresponds to running the CGI script `$CMI/cgi/MRentrancePage.pl`
- `MRentrancePage.pl` will either log the user in immediately (if login cookies are being used) or will display the login page where the user can enter their username and password.



How the FootPrints CGIs work

- All “active content” pages will have URLs of the form `http://server.example.com/MRcgi/name-of-page` which invokes the script `$CMI/cgi/name-of-page`
- Static content (such as images and help pages) is served from the following URLs and locations:

Content for...	Is served from...
<code>http://server.example.com/footprints/<u>file</u></code>	<code>\$CMI/html/<u>file</u></code>
<code>http://server.example.com/MRimg/<u>file</u></code>	<code>\$CMI/html/MRimg/<u>file</u></code>
<code>http://server.example.com/tmp/<u>file</u></code>	<code>\$CMI/html/tmp/<u>file</u></code>
<code>http://server.example.com/help/<u>file</u></code>	<code>\$CMI/html/help/<u>file</u></code>

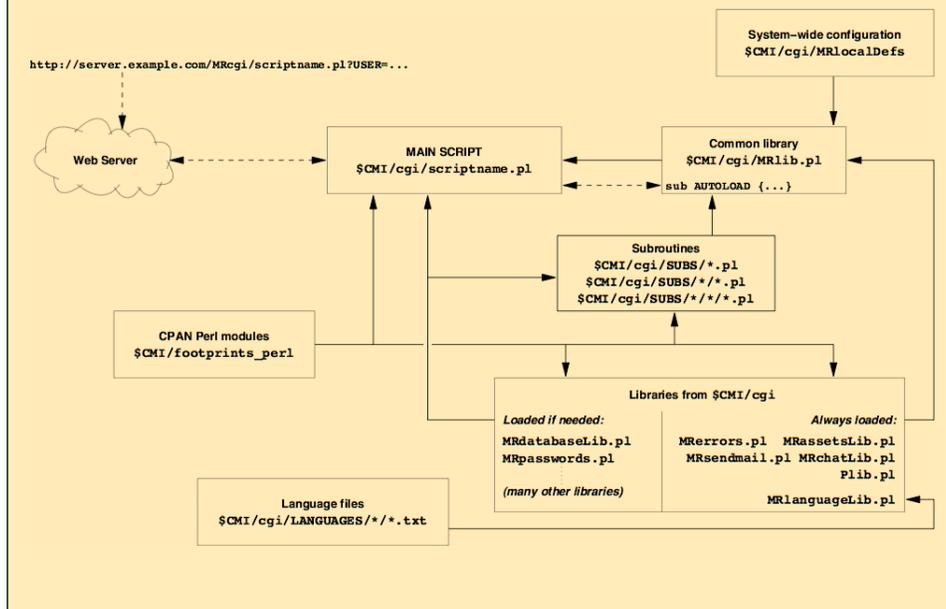


How the FootPrints CGIs work

- A FootPrints page such as `http://server.example.com/MRcgi/MRentrancePage.pl` (`$CMI/cgi/MRentrancePage.pl`) is almost never a complete script that can run as-is.
- As the script runs, it loads several other “pieces”.
 - Some pieces are loaded as soon as the CGI starts
 - Others are only loaded if/when they are needed.
- A complete CGI usually consists of:
 - The main CGI script: `$CMI/cgi/name`
 - Standard CPAN Perl modules such as `Net::LDAP`
 - Configuration files (partial scripts), for example `$CMI/cgi/MRlocalDefs`
 - FootPrints functionality libraries such as `$CMI/cgi/MRchatLib.pl`
 - FootPrints subroutines from the directory `$CMI/cgi/SUBS`
 - Language files (partial scripts): messages strings in English, French, etc.

How the FootPrints CGIs work

Files which form a FootPrints CGI script



How the FootPrints CGIs work - MRlib.pl

- `$CMI/cgi/MRlib.pl` is loaded by almost all FootPrints CGIs and does the following:
 - Loads `MRlocalDefs`, setting reasonable values for any variables which have been omitted from `MRlocalDefs`.
 - Initializes a lot of global variables.
 - Defines several subroutines (the most important of which is `AUTOLOAD`).
 - Loads several FootPrints libraries containing a large number of subroutines.



How the FootPrints CGIs work - MRlib.pl

- Perl runs a special subroutine named **AUTOLOAD** whenever a script tries to call a subroutine that hasn't yet been defined.
- **FootPrints defines AUTOLOAD in MRlib.pl**
 - It turns the subroutine name into a file name by adding "\$CMI/cgi/SUBS/" to the beginning and ".pl" to the end.
 - Any sequences of two underscores in the name of the subroutine are replaced with a filesystem separator character in the file name ('/' on Linux/Unix systems, '\' on Microsoft Windows).
 - If the file exists, FootPrints will have Perl load it (thus hopefully defining the subroutine) and the subroutine will then be called.
 - If the file does not exist, FootPrints will display an error message to the user.



How the FootPrints CGIs work - MRlib.pl

- **AUTOLOAD examples:**

Footprints will look for the undefined subroutine named...	In the file...
CheckLogin()	\$CMI/cgi/SUBS/CheckLogin.pl
Roles__GetUserRole()	\$CMI/cgi/SUBS/Roles/GetUserRole.pl

- **The benefit of using the AUTOLOAD mechanism is that most subroutines get loaded only if and when they are actually needed.**
 - Greatly reduces the amount of memory needed by each script.
 - Increases each script's startup speed.
 - Makes the FootPrints code more modular and easier to maintain.



How the FootPrints CGIs work - language files

- FootPrints CGIs do not directly contain messages or text that gets displayed to the user.
- Instead, FootPrints CGIs use variables, known as “language strings”, for each message that it needs to display:

```
print $STR::MRregister_command_SubtaskCreateSingle;
```

- These language strings are defined in separate files that just contain the messages that each CGI needs to display:

```
$STR::MRregister_command_SubtaskCreateSingle =  
  "Subtask was created";
```

- This enables FootPrints to support multiple languages (English, Portuguese, etc.) without having to change the FootPrints CGIs themselves -- new language string files are just added for each new language.



How the FootPrints CGIs work - language files

- Language strings are stored in special directories, grouped by language:

```
$CMI/cgi/LANGUAGE/name-of-language/name-of-file
```

- Each CGI or script will usually have its own language string file that has the same name as the CGI but with a “.txt” extension instead of “.pl”.

- Examples:

- \$CMI/cgi/LANGUAGE/PORTUGUESE/MRentrancePage.txt
- \$CMI/cgi/LANGUAGE/ENGLISH/MRentrancePage.txt



How the FootPrints CGIs work - language files

- CGIs will usually call the subroutine `importLanguageFile()` in order to load the correct language string file containing all of the messages that the CGI might need to display.
- `importLanguageFile()` chooses the correct language file based upon:
 - The system-wide language preference:
Administration -> System -> Languages
 - The project's language preference:
Administration -> Project -> Project Options
 - The user's language preference:
My Preferences



How the FootPrints CGIs work - language files

- Finding where a language string is used in FootPrints is a two step process:
 1. Find the name of the string by searching for the string in the language string file:


```
grep -i "select a modifier" \
  /usr/local/footprints/cgi/LANGUAGE/ENGLISH/*
```

In this case, the name is `$STR:MRsearch_page_116`
 2. Find where the string with this name is used in the code:


```
find /usr/local/footprints/cgi -name "*.pl" -print | \
  xargs grep STR:MRsearch_page_116
```

We search all files, not just the one indicated in the variable name (`MRsearch_page.pl`) because the string could be used in a subroutine under `$CMI/cgi/SUBS`.



How the FootPrints CGIs work - mod_perl

- `mod_perl` runs certain FootPrints scripts as `mod_perl` “servlets” instead of as CGIs.
- This places a lot of new restrictions and additional requirements on the FootPrints code:
 - The scripts that get run under `mod_perl` need to be in separate Perl packages and avoid using global variables or state, instead declaring all variables locally.
 - Since all FootPrints scripts share the same subroutines under `$CMI/cgi/SUBS` all of the FootPrints code has to be aware of the various Perl packages and namespace issues.
- Example of how the Perl packages affect the code:

```
$STR::MREditUser_Page_PageHeader_0
```

↑ ← ↑
variable type package name variable name



How the FootPrints CGIs work - example

- To see how a FootPrints CGI works overall, we’ll take a look at `$CMI/cgi/MREditUser_Page.pl`
- This is the script that is run when an agent or project administrator clicks on
Administration -> Project -> Edit Agents
- See Example 2 in the handout for an annotated version of the code.



The FootPrints frameset

- The regular FootPrints web interface uses four frames.
- `MRhomepage.pl` sets up the frames. If you view the HTML for any standard FootPrints web page, you'll see something like this near the end:

```
<frameset rows=85,* border=0 framemargin=0 frameborder=0>
  <frameset cols=*,1 border=0 frameborder=0>
    <frame src=/tmp/markmont1homepage8807.html marginheight=0 marginwidth=0
      scrolling=no noresize>
    <frame src=/tmp/markmont_homepageCheckbackFrame8807.html scrolling=no
      noresize>
  </frameset>
  <frameset cols=130,* border=0 framemargin=0 frameborder=0
    name=toolbarAndTicks>
    <frame src=/tmp/markmont3homepage8807.html marginheight=0 marginwidth=0
      frameborder=0 scrolling=auto name="toolbar" noresize>
    <frame src=/tmp/markmont2homepage8807.html framemargin=1 frameborder=0
      scrolling=auto name="ticks" noresize>
  </frameset>
</frameset>
```



The FootPrints frameset

- The files in the frameset generated by `MRhomepage.pl` are stored in the directory `$CMI/html/tmp`
- Each user who is logged in to FootPrints has their own personal set of these four files.
- `pid`, below, is the process id of the copy of `MRhomepage.pl` that generated the files for the user.

Filename	Frame	Contents
<code>username1homepagepid.html</code>	top left	logo, search box, project menu
<code>username2homepagepid.html</code>	bottom right	issue list, search results, etc.
<code>username3homepagepid.html</code>	bottom left	toolbar
<code>username4homepagepid.txt</code>	not displayed	contains copy of URL used to generate the home page
<code>username_homepageCheckbackFramepid.html</code>	top right, hidden (1 pixel wide)	Javascript to periodically call <code>MRcheckBack.pl</code> to see if anything needs to be done



Scheduled tasks

- Each minute, the FootPrints server automatically runs the script `$CMI/cgi/MRRunScheduledScripts.pl`.
 - On Unix/Linux, the cron daemon is what runs **MRRunScheduledScripts.pl**.
 - On Microsoft Windows, a special service named **FPSchedule** runs the script.
- **MRRunScheduledScripts.pl** will not run if it detects that another copy (from the previous minute) is still running.
- Proper operation of FootPrints requires **MRRunScheduledScripts.pl** run regularly, so most other scripts will check and log a warning or error to the FootPrints system log if it has not run in the last 5 minutes.
- **MRRunScheduledScripts.pl** usually invokes other scripts as appropriate to perform the necessary tasks.



Scheduled tasks

MRRunScheduledScripts.pl does the following:

- Run any auto-run reports that need to be run (every hour on the hour)
- Run any escalations that need to be run (check every 3 minutes)
- Send any outgoing backgrounded **SMTP** or **MAPI** email
- Send any outgoing backgrounded mass emails
- Get any incoming **IMAP** or **POP** email (every 3 minutes) or incoming **MAPI** email (every minute)
- Send any change management email reminders that need to be sent (every 3 minutes)
- Log any emails that could not be sent (once an hour)
- Send any calendar reminder emails (every 5 minutes)
- Create issues automatically from recurring calendar appointments (once a day)
- Un-assign deleted agents from their issues, possibly assigning others
- Automatically create master contact records from address book contacts
- Run any scheduled one-time purge or archive jobs
- Run any scheduled recurring purge or archive tasks (once a day)
- Remove any dump files from purges if the **MRlocalDefs** variable **\$DUMP_FILE_RETENTION_DAYS** is set (once an hour)
- Rotate large log files and delete any old log files (once an hour)
- Clean up any temporary files left behind by other scripts (once an hour)
- Merge the day's security logs into the archived security logs file (once a day)
- Merge today's log file with full log file (Microsoft Windows only, once a day)
- Do any scheduled database data import
- Do any scheduled address book data import



Scheduled tasks

MRRunScheduledScripts.pl also does the following:

- Create checkback refresh files for any users whose next checkback needs to do something (due to a broadcast message, escalation alert, error alert, timeout, etc.)
- Check for and kill hung processes (Unix/Linux only -- this is done by the **FPWatch** service on Microsoft Windows machines)
- Make sure the SQL service is up (Microsoft Windows only)
- Re-index any Oracle **CLOB** fields (Oracle databases only, every 30 minutes or when specified by the **MRlocalDefs** variable **\$REINDEX_INTERVAL**)
- Upgrade the database schema when new versions of FootPrints are installed (e.g., add generation 0 to retrofit the **MASTERx_DESCRIPTION** table)
- Make sure "**xhost +localhost**" has been run so that the Java classes can create charts for FootPrints (Unix/Linux only)
- Writes current license usage information to **\$CMI/etc/licenseUsage.txt** if **\$LICENSE_USAGE_REPORTING** is set in **MRlocalDefs** (every 6 minutes)



Log files

- In order to effectively support FootPrints and resolve problems, it is necessary to have detailed information about what FootPrints is doing internally.
- Log files provide this information.
- There are several types of log files that provide useful information about FootPrints' operations. Some are created by FootPrints itself, others originate from the other software upon which FootPrints relies:
 - Footprints system log
 - FootPrints scheduled tasks log
 - FootPrints security & query logs
 - FootPrints debugging logs
 - Web server logs
 - Database server logs
 - Email server logs
 - Operating system logs



Log files - FootPrints system log

- FootPrints saves general informational messages, warnings, and errors to `$CMI/etc/systemLog.txt` (special text format).
- This log can also be viewed by going to **Administration -> System -> Logs**
- FootPrints will give any system administrators who are logged in a pop-up telling them about any new errors or warnings, unless they have “Administrative Error Alerts” set to “Ignore Errors” in their preferences.
- If the file grows larger than 1 MB, FootPrints will rename it to have the date (and possibly a sequence number) in its name, and then start a new log file. (See next slide).
- FootPrints system log files are kept for 7 days after they are last modified.



Log files - FootPrints scheduled tasks log

- FootPrints runs `$CMI/cgi/MRRunScheduledScripts.pl` once each minute.
- A complete record of everything this script does is saved to `$CMI/temp/FPSchedule_log.txt` (regular text)
- If the file grows larger than 1 MB, FootPrints will rename it to have the date (and possibly a sequence number) in its name, and then start a new log file:
 - `FPSchedule_log_9_11_2005.txt` (oldest for this date)
 - `FPSchedule_log_9_11_2005_1.txt` (2nd oldest)
 - `FPSchedule_log_9_11_2005_2.txt` (3rd oldest)
 - ...and so on.
- FootPrints schedule log files are kept for 3 days after they are last modified.



Log files - FootPrints security & query logs

- FootPrints can keep records of who performs what actions. This is a new feature in FootPrints 7.0.
- Security and query logs can be enabled and configured by going to **Administration** → **System** → **Logs**. The configuration is stored in the YAML file `$CMI/etc/SecurityLogging.txt`
- There is no default location for the logs -- a system administrator has to specify a directory. The security logs are kept in CSV (comma separated value) format and are named using each user's username; query logs use a colon-delimited-field format and are named `Queries.username`
- The security and query logs get moved into an **ARCHIVE** subdirectory each day and then kept for a length of time specified by the system administrator.



Log files - FootPrints debugging logs

- FootPrints can save detailed debugging information about specific things it does (incoming email processing, LDAP queries, etc.).
- This functionality can be configured either by going to **Administration** → **System** → **Debug Manager** or by adding variables to `MRlocalDefs` (see Table 5 in the handout).
- FootPrints debugging log files are regular text files and are usually stored in the directory `$CMI/etc/debugging`. See the `MRlocalDefs` variables table for the actual file names and locations.
- The debugging log files can be sent to UniPress by going to **Administration** → **System** → **Logs**



Log files - FootPrints debugging logs

- If any FootPrints debugging log file grows larger than 1 MB, FootPrints will rename it to have the date (and possibly a sequence number) in its name, and then start a new log file.
- FootPrints debugging log files are kept for 3 days after they are last modified.
- If a CGI dies, hangs, or otherwise fails to run completely, “Sub Logging” (subroutine logging) is the best type of debugging to enable. It will show you a complete trace of what subroutines the script calls in what order. Sub logging is also very useful for learning how a CGI normally operates.



Non-FootPrints log files

- The web server log files are an excellent source of information about which FootPrints pages are being accessed and from where. Note that Apache is often configured to keep separate log files for accesses and errors.
- The database server log files are a good place to check for error messages if you suspect database-related problems. This does not apply to the FootPrints internal (GDBM) database, only external databases.
- Email server log files can be useful, if you have access to them. Incoming and outgoing email may be handled by separate servers and have separate log files, especially under Linux/Unix.
- Operating system log files.



Errorpages

- Usually, when a problem occurs, the FootPrints CGIs will call a subroutine named `Errorpage()`
- **Errorpage()** does the following:
 - Logs the error to the FootPrints system log.
 - Displays the error message to the user.
 - Include debugging information hidden as comments in the HTML source of the error message displayed to the user. These may also be displayed in as text on the web page that the user can see.
- **The debugging information includes:**
 - A Perl stack trace showing where the problem occurred in the code.
 - A list of all environment variables.
 - A few FootPrints-session specific variables.



Web troubleshooting tips

- Know where your web server logs are and how to check them.
- Use a web browser intended for web developers that includes features such as:
 - JavaScript console
 - JavaScript debugger
 - Information about the current page and frame information (not just the page's/frame's HTML source)
 - Form information.
- **Mozilla Firefox is a browser that supports all of these features.**
 - An extension to Firefox that provides additional helpful features is "Web Developer":
<http://chrispederick.com/work/firefox/webdeveloper/>



Web troubleshooting tips

- If you suspect that your web server is invoking the FootPrints CGIs improperly, it may be useful to display the CGI's environment (Linux/Unix primarily).
- Add the following lines of code near the top of the script, usually right after the comment that says "MAIN CODE":

```
open(DEBUGLOG, ">>/tmp/web-debug.txt");
print DEBUGLOG "\n--- $0 called at " . localtime() . "\n";
foreach my $var (sort(keys(%ENV))) {
    my $val = $ENV{$var};
    $val =~ s|\n|\n|g; # Escape newlines
    $val =~ s|"|\\"|g; # Escape double quotes
    print DEBUGLOG "{$var}=\\"{$val}\n";
}
close(DEBUGLOG);
```

- Or, you can just call `Errorpage()` instead if you don't need the script to finish running.



Web troubleshooting tips

- To show what form data a script is receiving, add the following code right after the script's call to `ReadParse()`:

```
open(DEBUGLOG, ">>/tmp/web-debug.txt");
print DEBUGLOG "\n--- $0 called at " . localtime() . "\n";
foreach my $var (sort(keys(%in))) {
    my $val = $in{$var};
    $val =~ s|\n|\n|g; # Escape newlines
    $val =~ s|"|\\"|g; # Escape double quotes
    print DEBUGLOG "{$var}=\\"{$val}\n";
}
close(DEBUGLOG);
```

- This is very useful to see what input the CGI is getting from the previous web page, and hence what the CGI is going to do.



Web troubleshooting tips - JavaScript

- The problem might not be with the web server or with FootPrints CGIs directly, but instead might be with JavaScript code generated by the FootPrints CGIs.
- JavaScript can be seen in the HTML source of FootPrints' main frame.
- Mozilla Firefox ships with an excellent JavaScript console that will tell you about JavaScript warnings and errors. The "Web Developer" extension for Firefox will provides you easy-to-see JavaScript status icons for each page.
- For more serious needs, Mozilla Firefox also has an extension that adds a complete JavaScript debugger.



LDAP troubleshooting tips

- Turn on LDAP debugging, perform a few LDAP-related actions in FootPrints, and then check the file `$CMI/etc/debugging/LDAPDebug.txt`
- LDAP debugging can be turned on via the **Administration -> System -> Debugging Manager** page, or by setting "`$DEBUGLDAP = 1;`" in `MRlocalDefs`.



LDAP troubleshooting tips

- If you are using an LDAP address book, check the contents of the file `$CMI/db/ABMASTER/MR/ldap.cfg`. You will see something like this:

```
ldap.itd.umich.edu          server
389                        port
ou=People,dc=umich,dc=edu  base
0                           options
Pachla                      lastname (for testing)
```

- Verify that the configuration is valid by using the following command:

```
ldapsearch -h server -p port -b "base" \
"(& (sn=lastname) (objectclass=person))"
```

This will show you the same thing FootPrints gets back from the LDAP server when doing a query. If this command does not work, then either the configuration is wrong or there is a problem with the LDAP server.



Email troubleshooting tips

- Check the system mail logs (may be on a different machine if your FootPrints server is not also a mail server).
- Turn on email debugging, send an email to FootPrints, and then check the files:


```
$CMI/etc/debugging/ImapMailDebug.txt
$CMI/etc/debugging/PopMailDebug.txt
$CMI/etc/debugging/GetOutlookMapiMailDebug.txt
$CMI/etc/debugging/InMailDebug.txt
```
- `$CMI/cgi/MRincoming.pl` is the script responsible for processing incoming email after it is retrieved from the mail server by one of the retrieval scripts.
- `MRincoming.pl` will often save problem emails in the directory `$CMI/temp`. Each message will be saved in its own file whose name begins with "IncomingMailErr_".



SQL database troubleshooting tips

- Check the database log files. These may be on another machine if the FootPrints server machine is not also the database server.
- Try to connect to the database and do manual queries using your normal database client (the “mysql” command line shell, Crystal Reports, etc).
- Try using the command-line APIs documented in chapter 11 of the FootPrints 7.0 Reference Manual, after setting up the necessary environment variables.
 - “`$CMI/bin/mrQuick issue-number`” is a good choice, since it is fairly low-level (basic) yet still returns useful information.
 - The code that does the real work is `$CMI/cgi/SUBS/mrQuick.pl` Turn on sub logging if desired.



General troubleshooting methodology

- Ideally, a case study would be included at this point in the presentation, giving an example of how user’s report about a problem with FootPrints was investigated and fixed.
- UniPress has fixed all the bugs I know about, so we’d have to deliberately revert some code in FootPrints 7.0c back to how it existed in FootPrints 6.6 in order to “un-fix” it and give us something to investigate and fix. But doing this would be artificial and unrealistic.
- So instead we’ll give a general troubleshooting methodology which will hopefully be applicable to a wide range of common situations.



General troubleshooting methodology

- **Step 1: Explore the problem.**
 - If the problem was reported by a user, make sure you can replicate the problem and verify that the user's description of the problem is accurate.
 - In what situations does the problem occur? Is it limited to a specific web browser? If it only occurs sometimes (as opposed to always), what are the conditions necessary to trigger the problem?
 - Once you do fix the problem, how will you know it is really fixed? Devise a test.



General troubleshooting methodology

- **Step 2: Determine which FootPrints scripts are involved.**
 - If the problem involves the web interface:
 - Using Mozilla Firefox, log in to FootPrints and go to the page "just before" where the problem occurs (assuming that the problem occurs after you press the "GO" button on that page).
 - Right-click in the main frame and select **This Frame -> View Frame Info**
 - Note the name of the script that generated the page (under "**General -> Address**").
 - Note the name of the script that will run when you click "GO" by examining "**Forms -> Form Action**". There may be multiple forms on the page, be sure to check the right one.
 - Note the values of the form fields for that form (you'll need to click on the form name first).



General troubleshooting methodology

- **Step 3: Determine how the scripts work.**
 - Familiarize yourself both with the script that generates the page where the “GO” button is as well as the script that runs when you click “GO”.
 - There’s no easy way to do this at first, but it will get easier each time you do it.
 - Turning on sub logging and then reproducing the problem can be a big help.
 - Adding instrumentation statements to the code -- such as “**print**” statements -- can also give you valuable information.
 - Make a backup copy of the script(s) first.
 - If **StartIt()** has been called, you can print HTML comments containing the information you want and view the information in your web browser.
 - Otherwise, write the information to a temporary file.



General troubleshooting methodology

- **Step 4: Form a hypothesis and test it.**
 - Narrow down as much as possible the location in the code the problem could be occurring.
 - Back up the original scripts before modifying them.
 - Try a fix and test it to see if it works. If it doesn’t work, undo the change. Repeat until fixed.
 - If the problem is web browser related, be sure to search the web for discussions of the same problem in other web-based applications.
 - Reference books on Perl, HTML, JavaScript, CGIs, your SQL database, and LDAP could be very helpful for finding and fixing the problem.



General troubleshooting methodology

- **Step 5: Clean up and complete the fix.**
 - Remove any instrumentation code or remnants of previous attempted fixes.
 - Test extensively to be sure that the problem really is fixed and that the fix does not create any additional problems.
 - If the problem and fix were specific to a particular web browser, it's particularly important to test the fix in all other common web browsers.
 - Look for other code in FootPrints that might have the same problem; fix it too.
 - Create a patch for the problem (how to do this is covered in Part II of this presentation, "Extending FootPrints In-House").
 - Document the problem.
 - Send the patch and documentation to UniPress for inclusion in the next release of FootPrints.



Final words

- **Supporting FootPrints in-house (particularly troubleshooting problems in-house) requires a lot of familiarity with the FootPrints code.**
- **The best way to develop this familiarity is to leap in and start working with FootPrints' internals. A lot of time and effort will be required at first, but the investment will pay off in the medium term and less time and effort will be required.**
- **Perseverance and thoroughness are important factors for success (not to mention luck!).**
- **Ask questions and seek advice at the BoF, on the FootPrints User Group, or contact me at footprints-server-admins@umich.edu**



Questions?



**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Extending

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Extending



FootPrints “Under the Hood” Part II: Extending FootPrints In-House

Presented by:
Mark Montague
The University of Michigan
markmont@umich.edu

UNIPRESS SOFTWARE 2 0 0 5
FOOTPRINTS
User Conference



Introduction

- **Content:**
 - External extensions to FootPrints.
 - Internal extensions/customizations to FootPrints.
 - Examples.
 - We will *not* cover the XML/SOAP interface since that is being covered in the 2 pm Monday and 10:15 am Wednesday Advanced Tech Track sessions.
- **Audience:**
 - System administrators or programmers.
 - Familiarity with how FootPrints works.
 - Familiarity with Perl or a similar scripting language.
 - Some knowledge of CGI technologies and HTML.
- **Questions: At the end or during the “Birds of a Feather” session Tuesday at 12:15pm.**



Platform assumptions

FootPrints	Version 7.0c. Material in this presentation may not apply to other versions.
Operating system	Linux/Unix. Shell commands will need to be changed for Microsoft Windows servers, but everything else covered in this presentation should remain applicable.
Database	Any SQL-based database supported by UniPress for use with FootPrints. Some material in this presentation will not be applicable to organizations using the FootPrints internal (GDBM) database.
Web server	Apache. But most material in this presentation is independent of which web server is used.



Types of extensions

- **There are two main types of extensions to FootPrints: external and internal.**
 - *External extensions:* Extensions that are outside of the FootPrints code provided by UniPress. External extensions can make use of FootPrints as a system or re-use pieces of the FootPrints code. External extensions can typically be used without logging into FootPrints.
 - *Internal extensions:* Extensions that add to or change FootPrints' code. Internal extensions typically require a user to be logged in to FootPrints in order to use them and are usually accessed from within FootPrints itself.



External extensions - Location

- **An external extension may need to reside on the FootPrints server if it:**
 - Requires direct access to files maintained by FootPrints (for example, configuration files or attachments).
 - Re-uses FootPrints code that resides on the FootPrints server.
- **Otherwise, external extensions can usually reside anywhere, including on the FootPrints server itself.**



External extensions - Examples

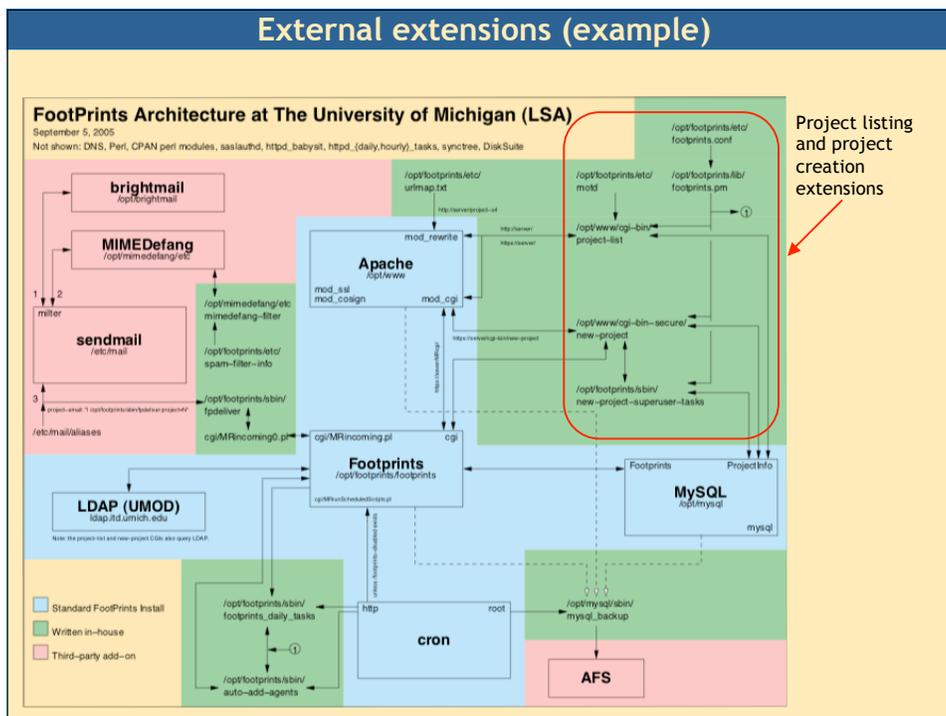
- **View active issues**
 - Directly queries the FootPrints database and displays basic information for all issues in a project that meet certain criteria (for example, open issues which are marked as public).
 - Can reside on any web server, or could even be turned into a command-line tool.
 - Example 1 in the handout shows some sample code that does this.
- **Submit new issue**
 - Uses FootPrints' built-in HTML form processing functionality to let people submit issues to a project without accessing FootPrints itself.
 - Can be embedded into non-FootPrints web sites.
 - Could alternatively be implemented via XML/SOAP.



External extensions - Examples

- **Project listing page**
 - Provides a list of all “public” projects on the FootPrints server so users can select the one they want.
 - Resides on the FootPrints server because it needs to access various FootPrints configuration files (such as **\$CMI/etc/MRprojects**).
 - Uses a database to keep track of additional information about each project, such as a short description, whether a project should be displayed in the list, etc.
- **Project creation page**
 - Allows certain users (e.g., managers) who are not system administrators to create new projects themselves, instantly.
 - Resides on the FootPrints server but could be enhanced to allow it to reside elsewhere.

External extensions (example)





External extension types

- There are several different methods for implementing external extensions, each representing a different type of extension:
 - Use the FootPrints XML/SOAP web services API
 - Use the FootPrints database API
 - Access the FootPrints database directly
 - Emulate a web browser
 - Emulate a web server
 - Reuse FootPrints libraries and subroutines
 - Access FootPrints configuration files directly
- A single extension may be able to employ several of these strategies at once, using the easiest way to accomplish each thing it needs to do.



External extension types - XML/SOAP web services API

- See pages 499 - 538 of the FootPrints 7.0 Reference Manual for information about the XML/SOAP API for FootPrints, including example code.
- Also see <http://www.unipress.com/footprints/webservices/>
- External extensions using SOAP/XML web services can run on any machine; they are not restricted to running just on the FootPrints server itself.
- XML/SOAP will likely increasingly displace other external extension types in future versions of FootPrints.
- For more information, refer to the 2 pm Monday and 10:15 am Wednesday Advanced Tech Track sessions, which cover the XML/SOAP web services API in more detail than we could provide here.



External extension types - Use the FootPrints database API

- This strategy makes use of high-level “FootPrints database commands” provided by UniPress in the `$CMI/bin` directory.
- These commands represent an “intermediate level” of abstraction in between the FootPrints CGIs (higher level) and accessing the database used by FootPrints directly (lower level).
- See pages 483 - 496 of the FootPrints 7.0 Reference manual for a list of all of the commands and how to use each of them.
- Extensions which employ this strategy must reside and run on the FootPrints server.



External extension types - Use the FootPrints database API

- Simple example of how to use the FootPrints database API:

```
#!/bin/sh
CMI="/usr/local/footprints"
CMMASTER="$CMI/db/MASTER1"      # Project location of project 1
ABMASTER="$CMI/db/ABMASTER1"  # Addressbook location of AB 1
export CMI CMMASTER ABMASTER
PATH="$CMI/bin:$PATH"
mrQuick 2                        # Show basic info about ticket 2
```

- The shell script above produces the following output:

```
Query results are listed below...
2   5   2005-09-18 23:03:59   footadmin   Open
      2005-09-18 23:03:59   Database API demo
Number of entries in this report: 1
Number of total matching entries: 1
```



External extension types - Access the database directly

- Extensions can send SQL statements directly to the database server, bypassing all FootPrints code.
- Extensions which employ this strategy can run on any machine.
- See Example 1 in the handout for some minimalistic code that demonstrates this type of extension to FootPrints.
- A real-world example is available at https://mailmove.lsa.umich.edu/csg-fp-cgi/csg_stats.pl



External extension types - Emulate a web browser

- The external extension to FootPrints pretends to be a user with a web browser and sends custom-crafted HTTP requests over the network to the FootPrints server.
- This can be easily done by using the Perl module `LWP::UserAgent`. Similar modules/classes/libraries are available for other languages.
- External extensions to FootPrints that employ this strategy can run on any machine, not just the FootPrints server.
- The XML/SOAP API will eventually remove the need for external extensions to emulate a web browser.
- For an example showing how `LWP::UserAgent` can be used, see `$CMI/cgi/MRsendClones.pl` and `$CMI/cgi/MRreceiveClone.pl`



External extension types - Emulate a web server

- **The external extension pretends to be a web server:**
 - Sets up the environment (i.e., the `%ENV` hash in Perl) the same way a web server would. Any input would typically be put in the **QUERY_STRING** environment variable.
 - Runs a FootPrints CGI script as a child process and captures any output.
 - Examines the HTML output generated by the FootPrints CGI to determine whether the operation succeeded and/or extracts any desired information from the output.
- **This strategy for implementing an external extension to FootPrints is used extensively by the “project creation” CGI written at the University of Michigan which allows non-system-administrators to create new FootPrints projects.**



External extension types - Emulate a web server

- **This type of extension must be reside on the FootPrints server itself.**
- **See Example 3 in the handout for some minimalistic code that demonstrates this type of extension to FootPrints.**



External extension types - Reuse FootPrints libraries and subroutines

- External extensions (CGIs, scheduled tasks, etc.) can be written to use the subroutines included in FootPrints' `$CMI/cgi/SUBS` directory.
- Some “jumping through hoops” (manually setting global variables, “faking” the extensions’ environment, etc.) may be necessary in order to get everything to work properly.
- Note that although the extension is “pulling in” FootPrints code to use, the extension is external to FootPrints itself -- that is, the extension is not accessible from the FootPrints web interface, it resides outside of the `$CMI` directory, and so on.



External extension types - Reuse FootPrints libraries and subroutines

- Extensions which employ this strategy must run on the FootPrints server.
- The CGI written at the University of Michigan that allows non-system-administrators to create new projects uses this technique in order to provide a “FootPrints look and feel” to its web page.
- See Example 4 in the handout for some code that demonstrates this type of extension to FootPrints.



External extension types - Access FootPrints configuration files directly

- Extensions can read and modify the files in `$CMI/db` and `$CMI/etc` directly, bypassing all FootPrints code.
- Extensions which employ this strategy must run on the FootPrints server.
- See Example 5 in the handout for some code that demonstrates this type of extension to FootPrints.



Internal extensions

- Internal extensions, changes, and customization of FootPrints most often involve simply changing the FootPrints CGI scripts that are provided by UniPress.
- In some cases internal extensions can involve creating new FootPrints CGIs -- in this case, though, existing FootPrints CGIs still need to be modified in order to make the new FootPrints CGIs accessible via the web.



External extensions - Change management

- **External extensions usually don't require much, if any, change management:**
 - External extensions are usually self-contained files.
 - They can usually be copied onto any additional servers or machines that need them.
 - External extensions are usually not touched by FootPrints upgrades.



Internal extensions - Change management

- **Internal extensions do require change management:**
 - Internal extensions involve changes to code shipped with UniPress. These changes will be undone during any upgrade to a new version of FootPrints.
 - If all FootPrints servers (e.g., development server, test server, production server) will all have the same internal extensions, the extensions can be added to a new server by copying the affected files. But there are several drawbacks to this approach:
 - It does not allow other people to see *just* the things that were changed -- changes are mixed in with UniPress code.
 - It's not possible to easily pick and choose which internal extensions a given server should have.
 - If a problem is discovered or needs change, it is desirable to be able to remove only one internal extension without affecting anything else.



Internal extensions - Change management

- The following slides describe the FootPrints code change management process used at the University of Michigan.
 - It is based on the “**diff**” and “**patch**” tools.
 - These tools come standard with all Linux/Unix systems, although if you are using something other than Linux, you may want to install the latest version from <http://www.gnu.org/> as the GNU versions of these tools are more powerful than vendor-specific versions.
 - “**diff**” and “**patch**” are also available for Microsoft Windows:
<http://gnuwin32.sourceforge.net/packages/diffutils.htm>
<http://gnuwin32.sourceforge.net/packages/patch.htm>
- You can develop your own change management process and/or use different tools, if you prefer.



Internal extensions - Change management

- It is highly recommended that you have a separate test server for FootPrints which you use to develop and test all of your internal extensions to FootPrints. You should *not* use your regular production server:
 - It can take hours, days, or weeks to completely write and test internal extensions to FootPrints.
 - It is very likely that whatever part of FootPrints the new internal extensions affects will be broken (unusable) until you finish development.
 - Data corruption could occur in the database and/or configuration files which could be difficult to track down and fix manually. This is especially likely if users try to use the new functionality you’re adding before the code for it is complete.



Internal extensions - Change management

- **Step 1: Create a “pre-modification copy” of FootPrints**
 - This is the “before” snapshot.
 - Copy the entire **\$CMI** directory. Give the copy a descriptive name such as “**footprints-pre-global-link-change**” (if you’re changing FootPrints’ Global Link functionality, for example).
 - Note that on Linux/Unix you should use “**tar**” to make the copy instead of “**cp -r**” since “**cp**” will not handle symbolic links correctly.
 - If you want this to be a full backup that you can revert to in the event of serious problems, be sure to make a complete backup of the database that FootPrints uses too.
 - You’ll only need the database backup if you need to restore the **\$CMI/db** or **\$CMI/etc** directories. If you make a mistake that breaks one of the CGIs, you can simply copy an unmodified version of the CGI from the “pre-modification” copy of FootPrints you made.



Internal extensions - Change management

- **Step 1: Create a “pre-modification copy” of FootPrints (continued)**
 - Example:

```
cd /usr/local
mkdir footprints-pre-global-link-change
(cd footprints ; tar cf - *) | \
(cd footprints-pre-global-link-change ; tar xpf -)
```



Internal extensions - Change management

- **Step 2: Implement your internal extension**
 - Change whatever files under `$CMI/cgi` need to be changed in order to implement the new internal extension.
 - Test thoroughly.



Internal extensions - Change management

- **Step 3: Create a patch that captures the changes needed for your internal extension to FootPrints**
 - Use the “`diff`” program to compare the contents of all files under the `$CMI/cgi` directory with the files in the “before” snapshot you made in step 1, saving the results to a file. For example,

```
cd /usr/local
diff -urdN \
    footprints-pre-global-link-change/cgi footprints/cgi \
    > global-link-change.patch
```
 - Be sure to compare only the “`cgi`” subdirectories since if you include the `$CMI/etc`, `$CMI/db`, and `$CMI/html` subdirectories, you’ll get a lot of unrelated changes that were made by FootPrints itself during its normal course of execution.



Internal extensions - Change management

- **Step 4: Inspect and clean up the patch**
 - Edit the file created in step 3 and make sure all the changes you made in step 2 are present and correct.
 - Remove any changes that were made by FootPrints -- for example, during its normal course of execution, FootPrints might make some changes to `$CMI/cgi/MRlocalDefs` (you won't want these to be included in your patch because `MRlocalDefs` is usually specific to individual servers and it would confuse the "patch" program).



Internal extensions - Change management

- **Step 5: Apply the patch to your production FootPrints server**
 - Copy the patch file to your production FootPrints server, putting it in your `/usr/local` directory.
 - Do a complete backup of both `$CMI` and the database.
 - Use the "patch" program to apply the patch:

```
cd /usr/local
patch -p 0 < global-link-change.patch
```
 - Please refer to the documentation for the "diff" and "patch" programs on your system for more information.
 - The commands above can often also be used to re-apply the extension to the server after upgrading to a new version of FootPrints.



Internal extensions - Managing upgrades

- Internal extensions to FootPrints can cause maintenance problems when upgrading to new versions of FootPrints.
 - The “**patch**” program works by looking for certain lines in the original file and then adding your changes at the same place.
 - If a new version of FootPrints contains significant changes to a file that you modified for a previous version of FootPrints -- for example, if a section of the file was re-written for the new version -- the “**patch**” program or other tool you need may not be able to find the areas in the new file where the changes for your internal extension to FootPrints need to be made.
 - The upgrade from FootPrints 6.6 to FootPrints 7.0 causes such a problem for just about every file that comes with FootPrints due to the sweeping changes in variable names that were required when adding support for **mod_perl**.



Internal extensions - Managing upgrades

- Unlike external extensions to FootPrints, you should plan on internal extensions to FootPrints requiring an on-going maintenance commitment as new versions of FootPrints are released.
- You may need to do some of the work of the “**patch**” program yourself by making some of the changes by hand.
- In the worst case, you may need to completely re-implement your internal extension for the new version of FootPrints.
- Keep all this in mind when deciding whether to change FootPrints, especially when contemplating large scale changes. Don't get your users used to customizations that you won't be able to commit to for future versions of FootPrints.



Internal extensions - Managing upgrades

- The maintenance burden of internal extensions can be minimized by considering whether the change you are making are general enough that other organizations would find them useful, too.
- If you think that other organizations are likely to benefit by your change, spend a bit of extra time when implementing your internal extension to FootPrints for the first time so that it is well-written, complete, and takes into account cases that other organizations might encounter which might not be relevant to your organization.
- You can then submit the extension to UniPress and ask them to include it in the next release of FootPrints. This way, UniPress is, in effect, taking on the responsibility of making sure that new functionality added by your extension will continue to work in future versions of FootPrints, relieving you of the burden of maintaining the extension yourself.



Internal extensions - Managing upgrades

- For internal extensions that are specific to your organization and would not be useful to others, you can minimize the maintenance burden by keeping the changes you made as small and as simple as possible.
- Consider whether a change really needs to be made before committing to making and maintaining it, and reject any requests for large, unnecessary, or complex changes. Determine whether an external extension to FootPrints could achieve the same result, as external extensions need to be updated less often than internal ones, and when they do need to be updated, the changes are usually smaller.
- Finally, if you don't want to implement an internal extension yourself, you can often pay UniPress to do the work and implement it for you.



Internal extensions - Examples

- **Change the default type for quick searches**
 - At the University of Michigan, several FootPrints agents reported that the “quick search” in the top frame of FootPrints web pages did not give the desired results.
 - It defaults to a Title search.
 - The agents were expecting a Keyword search.
 - Many agents did not know about manually changing the search type, and those who did know would sometimes forget to do it.
 - So it was decided to change the default selection for the quick search from “Title” to “Keyword”. This is a trivial two-line change to `$CMI/cgi/SUBS/MRhomepage/printTopFrame.pl`

Example - Changing the default search type

```
diff -urdN footprints-004-default-search/cgi/SUBS/MRhomepage/printTopFrame.pl →
footprints/cgi/SUBS/MRhomepage/printTopFrame.pl
--- footprints-004-default-search/cgi/SUBS/MRhomepage/printTopFrame.pl 2005-06-29 →
15:03:52.000000000 -0400
+++ footprints/cgi/SUBS/MRhomepage/printTopFrame.pl 2005-09-19 →
20:31:42.257879000 -0400
@@ -251,8 +251,8 @@
        # first row is radio buttons
        print FFILE1 "<tr><td colspan=2 align=center valign=middle><font
color=\"$FP::searchframetext\">";
-       print FFILE1 "$STR::MRhomepage_25<INPUT TYPE=RADIO NAME=WSEARCH VALUE=TITLE CHECKED> →
&nbsp;";
-       print FFILE1 "$STR::MRhomepage_95<INPUT TYPE=RADIO NAME=WSEARCH VALUE=KEYWORD> &nbsp;";
+       print FFILE1 "$STR::MRhomepage_25<INPUT TYPE=RADIO NAME=WSEARCH VALUE=TITLE> &nbsp;";
+       print FFILE1 "$STR::MRhomepage_95<INPUT TYPE=RADIO NAME=WSEARCH VALUE=KEYWORD CHECKED> →
&nbsp;";
        print FFILE1 "$STR::MRhomepage_27<INPUT TYPE=RADIO NAME=WSEARCH VALUE=NUM>";
        if ($FP::CUSTOM_WSEARCH_FIELD)
        {
```

→ Indicates that the line is wrapped and continues *without* a new line being started in the patch file.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Wrapped reports (**Reports** -> **Custom** -> **Wrapped**) support line breaks in the reports but not page breaks.
 - Page breaks are useful if you want to print out the reports and have each issue on its own page, but FootPrints doesn't support page breaks out of the box.
 - Let's add page break functionality to wrapped reports!
 - Only 41 lines of code will need to be added or changed to add the new functionality, but these lines are scattered through a number of files, and knowing *where* these lines should be is by far the hardest part of creating the extension.
 - **Caution:** this functionality was ported to FootPrints 7.0 from FootPrints 6.6 with only a little testing, so use it at your own risk.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Many things under FootPrints have this pattern:
 - Front-end user interface page.
 - Back-end script that does the work.
 - Communication or glue connecting the front- and back-ends.
 - Our strategy:
 - Add a "page break" button to the "Create a New Report" page.
 - Get the "Create a New Report" page to pass the page break information to the back-end script that generates the report (in this case, it involves how report definitions are stored internally).
 - Modify the back-end script that creates the report so that it put the page breaks where the front-end script tells it to.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Step 1: Add a “page break” button to the “Create a New Report” page.
 - Go to **Reports -> Custom -> Wrapped**
 - Using Mozilla Firefox, right click and select **This Frame -> View Frame Info**, and get the script name from the “General” tab.
 - The script we need to modify is **\$CMI/cgi/MRsearch_page.pl**
 - From this point, it’s a matter of figuring out how the script works. We note that this script has a lot of dedicated subroutines in the directory **\$CMI/cgi/SUBS/MRsearch_page**
 - Page breaks “should” be similar to line breaks, so we model the changes we make on the existing line break code.
 - See example 6 in the handout for the changes made.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Step 2: Get the “Create a New Report” page to pass the page break information to the back-end script that generates the report.
 - Based on what we’ve learned by studying **MRsearch_page.pl**, we make any changes we know need to be made.
 - Since we’re modeling the page break code on the line break code, we do a case-insensitive search for the word “**line**” on all of the **MRsearch_page.pl** code and all of the subroutines it uses to try and find things that need to be added or changed that we don’t know about. This is imprecise and time-consuming but surprisingly effective.
 - We’re bound to miss some things, discover problems during testing, and have to come back later to make more changes.
 - See example 7 in the handout for the changes made.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Step 3: Modify the back-end script that creates the report so that it put the page breaks where the front-end script tells it to.
 - Go to **Reports -> Custom -> Wrapped**
 - Using Mozilla Firefox, right click and select **This Frame -> View Frame Info**, and get the back-end script name from the “Forms” tab.
 - The script we need to modify is **\$CMI/cgi/MRdirectSearch.pl**
 - From this point, it’s a matter of figuring out how the script works. We note that this script has a lot of dedicated subroutines in the directory **\$CMI/cgi/SUBS/MRdirectSearch**
 - Page breaks “should” be similar to line breaks, so we model the changes we make on the existing line break code.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Step 3 (*continued*): Modify the back-end script that creates the report so that it put the page breaks where the front-end script tells it to.
 - The changes to the back-end script are the “meat” of the extension: we need to come up with an algorithm for figuring out where page breaks are supposed to be (and hence starting/ending table rows at the correct places). We keep track of this through a new hash, **%newPageAfterRow**, whose keys are the row numbers and values are true if we should have a page break at that point.
 - We implement page breaks by printing an HTML line break but setting a CCS2 property to make the web browser begin a new page *when printing the report*:


```
<br style="page-break-before: always;" />
```
 - See example 8 in the handout for the changes made.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Step 4: Get **MRdirectSearch.pl** to get the page break information passed to it by the front-end script. This is “Step 2 *redux*”.
 - Based on what we’ve learned by studying **MRdirectSearch.pl**, we make any changes we know need to be made.
 - Since we’re modeling the page break code on the line break code, we do a case-insensitive search for the word “**line**” on all of the **MRdirectSearch.pl** code and all of the subroutines it uses to try and find things that need to be added or changed that we don’t know about. This is imprecise and time-consuming but surprisingly effective.
 - We’re bound to miss some things, discover problems during testing, and have to come back later to make more changes.
 - See example 9 in the handout for the changes made.



Internal extensions - Examples

- **Add page breaks to wrapped reports**
 - Step 5: Test everything exhaustively, fix any problems, document everything, create the patch, and clean up.



Final words

- While FootPrints does not require any programming, it's powerful enough and open enough to permit you to extend it via programming if needed.
- Extending FootPrints in-house requires a lot of familiarity with the FootPrints code and a significant amount of time and effort.
- The best way to develop this familiarity is to leap in and start working with FootPrints' internals.
- You'll need to assess whether the benefits your organization receives are worth the time and effort required -- both initially as well as in terms of on-going maintenance costs.
- Ask questions and seek advice at the BoF, on the FootPrints User Group, or contact me at footprints-server-admins@umich.edu



Questions?



**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Tables

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Tables

Note

This table does not include all of the variables it should, although it is more complete than it was in October 2004.

Table 1. MRlocalDefs variables - web interface

Variable	Description
\$AB_ADMIN_SYSADMIN_ONLY	only system administrators can do address book administration, if set
\$AGENT_CONTACT_DEFAULT_CHECKED	controls whether the "Update contact" checkbox is checked by default on the Create Issue and Edit Issue pages
\$ALLOW_AGENTS_TO_VIEW_DELETED	let agents view deleted issues
%ALLOW_AGENTS_TO_VIEW_DELETED	hash of project numbers for which agents are allowed to view deleted issues
\$ALLOW_PROJ_ADMIN_TO_PURGE	let project administrators purge and/or archive issues
\$ALL_CONCURRENT_LOGOUT	enable "logout all concurrent agents" button
%ALTERNATE_LOGOUT_URL_GROUP	where to send group users after logging them out, keyed by project
\$ALTERNATE_LOGOUT_URL_GROUP	ditto, if no entry for the above
%ALTERNATE_LOGOUT_URL_INTERNAL	where to send agents after logging them out, keyed by project
\$ALTERNATE_LOGOUT_URL_INTERNAL	ditto, if no entry for the above
%ALTERNATE_LOGOUT_URL	where to send anyone who logs out if none of above apply (keyed by project)
\$ALTERNATE_LOGOUT_URL	ditto, if no entry for the above
\$ALTERNATE_REAL_NAME_SORT	names are of the form "Smith, Bob" not "Bob Smith"
\$ALWAYS_SHOW_AUTOADD	whether autoadded users should be shown in the select-user popups
\$CHAT_DISABLED_FOR_THESE_USERS	whitespace separated list of users for whom not to allow chat
\$CONCURRENT_IDLE_TIMEOUT	concurrent users may be logged out at any time if idle for more than this many minutes
\$CONCURRENT_INACTIVE_TIMEOUT	set higher than idle timeout to log out concurrent users idle for this many minutes first
\$CUSTOMER_CANT_SEE_ISSUE_DETAILS	in combination with \$CUSTOMER_DEFAULT_TO_EDIT_INSTEAD_OF_DETAILS, prevents customer from seeing issue details
\$CUSTOMER_CONTACT_DEFAULT_CHECKED	whether the "update your contact information" checkbox is checked by default for customers
\$CUSTOMER_DYNAMIC_SQL_LINK	enables Dynamic SQL Link functionality for customers creating issues
%CUSTOMER_QUICK_CREATE_ONLY	hash of project numbers for which customers should only be allowed to create quick issues
%CUSTOMER_QUICK_CREATE_ON_LOGIN	hash of project numbers for which customers should be sent to the quick issue create page immediately upon logging in
\$CUSTOMER_SEARCH_ABFIELDS	enable customers to do searches on address book fields

Variable	Description
\$CUSTOMER_DEFAULT_TO_EDIT_INSTEAD_OF_DETAILS	send customers straight to the edit page in places where they would normally see issue details
\$DEFAULT_ADD_TO_INTERNAL_KB	makes “Internal Knowledge Base” the default choice when creating solutions
\$DEFAULT_TIME_TRACKING_MINUTES_CREATE	default number of minutes for the time tracking fields when creating a new issue
\$DEFAULT_TIME_TRACKING_MINUTES_EDIT	default number of minutes for the time tracking fields when editing an issue
\$DESCRIPTION_COLS	override the project preference for number of columns in input boxes for multi-line fields
\$DESCRIPTION_ROWS	override the project preference for number of rows in input boxes for multi-line fields
\$DISABLE_CHAT	turn off chat for everyone
%DISABLE_CHECKBACK	hash of users for whom not to do checkbacks
\$DISABLE_CHECKBACK	don’t do checkbacks for anyone
@DISABLE_CREATE_BUTTON_PROJECTS	don’t put a “Create Issue” button in the toolbar for any of these projects
\$DISABLE_WAIT_SPLASH	Turn off the “Please Wait…” progress bar splash window on homepages and search result pages
\$DONT_SHOW_OLDDESC_ON_EDIT	don’t include a “View Current Description” button on the issue edit page
\$DOWNLOAD_ATTACHMENTS_INLINE	display attachments in browser as if they are normal links to real files, as FootPrints used to do before version 7.0 (all users)
%DOWNLOAD_ATTACHMENTS_INLINE	hash of usernames for which to display attachments in browser as if they are normal links to real files, as FootPrints used to do before version 7.0
\$DO_NOT_CHECK_INCLUDE_IN_CHILD	for master or global issues, make the “include description in” subtasks / global links checkbox unchecked by default
\$DROPDOWN_FOR_CREATE_AND_UPDATE_CONTACT	replace the “create contact” checkbox with a drop down menu for creating / updating / using current contact instead
\$EDIT_ALL_TIME_TRACKING	allows agents to add time for any other agent
%EDIT_PAGE_FIELD_LOCKING	hash of project numbers for which to enable field locking on the edit page
\$FAILED_LOGIN_MESSAGE	custom message to display when a user fails to log in; useful for suggesting an alternate login URL
\$FIRST_CALL_RESOLUTION_EXTRA_STATUS	another status in addition to “Closed” which should count for an issue being Resolved in the First Call Resolution report
\$FOCUS_ON_DESC_FOR_EDIT	when displaying the edit page, put the cursor in the description box so the user can immediately begin typing a new description without having to click there first
\$fontface	what font to use for most things
%GROUP_USER_TOP_FRAME_LINKS	up to 4 custom links in top frame next to logo
\$HIDE_DELETE_BUTTON	hide the delete-issue button for all but system administrators
\$HIDE_INDIVIDUAL_USERS_SECTION	do not display the “Individual Users” pseudoteam in assignee boxes

Variable	Description
<code>%INTERNAL_USER_TOP_FRAME_LINKS</code>	up to 4 custom links in top frame next to logo
<code>\$logoLogin</code>	filename for logo on login page
<code>\$logoMain</code>	filename for logo in top frame
<code>\$MAX_USER_PROJ_LENGTH</code>	truncate displayed user/project names in top frame
<code>\$NO_APPLET_FOR_CHECKBACK</code>	do checkbacks, but don't use Java applet for it
<code>\$NOCONTACTBOX</code>	disables "update your contact information" checkbox
<code>\$NO_FEEDBACK_LINK</code>	don't show the UniPress feedback link for system administrators
<code>\$PROJ_NAME_COLUMN</code>	enables project-name column for cross project listings
<code>%SHOW_DETAILS_DATA_AS_URL</code>	replaces details data with hyperlink, keyed by field
<code>\$SIMPLE_TOOLBAR</code>	turn off the dynamic toolbar for everyone
<code>@SIMPLE_TOOLBAR</code>	turn off the dynamic toolbar for these users
<code>\$textarea_cols</code>	how wide text entry boxes are (default: 85 characters)
<code>\$textarea_rows</code>	how tall text entry boxes are (default: 10 lines)
<code>\$TOPFRAMESIZE</code>	how tall top frame is (default: 85 pixels)

Note

This table does not include all of the variables it should, although it is more complete than it was in October 2004.

Table 2. MRlocalDefs variables - LDAP

Variable	Description
\$DETAILS_BASED_ON_DN_LDAP	extract LDAP search base to use from the search's DN
\$DONT_STOP_FOR_LDAP_BIND_ERRORS	continue with LDAP searches even if the LDAP bind fails
@EXACTSEARCH	do exact, not approximate, searches on these LDAP fields to avoid LDAP server timeouts
@MULTIPLE_LDAP_BASES	use these LDAP search bases for all projects
\$NO_MULTIPLE_VALUES_LDAP	return only first value in each LDAP field

Note

This table does not include all of the variables it should, although it is more complete than it was in October 2004.

Table 3. MRlocalDefs variables - incoming email

Variable	Description
\$AGENT_EMAIL_EDIT_IF_CC	if an agent is a permanent CC or has ever received email for an issue, let them edit the issue
\$AGENT_EMAIL_EDIT_IF_CONTACT	if an agent is the contact for an issue, let them edit the issue
\$ALLOW_AGENT_TO_SET_STAT_TO_REQUEST_VIA_EMAIL	let agents change an issue's status to "Request" via email
\$ASSIGN_INCOMING_BASED_ON_TO	allows auto-assignment to teams based on which email address a message is sent to; see \$CMI/etc/AutoAssignIncomingEmail.yaml
\$BOUNCE_BLOCKED_INCOMING_MAILS	email address to which a notification should be sent anytime an incoming email gets blocked
%BYPASS_EMAIL_FILTERS	allows overriding of no-accept rules for certain senders, subjects, or bodies
\$CONTACT_KEYWORD	Command for overriding contact name (defaults to "contact\s*=")
\$CREATE_NEW_ISSUE_IF_BAD_EDIT_NUMBER	if issue specified in the subject does not exist, create a new issue
\$DONT_SEND_MAIL_FOR_PROJPREFS26	Don't include the sender explicitly on notifications when anyone can edit any issue via email
\$EMAIL_FIELD_SEPARATOR	for incoming emails (defaults to "=")
\$FIX_BRACKETED_URLS_IN_EMAILS	remove angle brackets ("<>") from around URLs in incoming messages (the brackets can cause problems for FootPrints)
%INCOMING_MAIL_DENY_EDIT_NUMBERS	per-project strings of issues that can't be edited via email
\$INCOMING_MAIL_ERRORS_DO_NOT_INCLUDE_BODY	only headers will be included
@INCOMING_MAIL_ERRORS_INCLUDE_HEADERS	if only including headers, limit the ones included to these
\$MULTILINE_CONTINUATION_CHAR	for incoming emails (defaults to "\\")
@OUTLOOK_ORIGINAL_MESSAGE_TEXT	markers for start of previous message
\$POP_PORT	(MS Windows only?)
\$SENDER_KEYWORD	Command for overriding sender name (defaults to "sender\s*=")
\$USE_SECOND_FROM_HEADER	Ignore first "From:" header in message (for mail gateways that add an extra one)

Note

This table does not include all of the variables it should, although it is more complete than it was in October 2004.

Table 4. MRlocalDefs variables - outgoing email

Variable	Description
\$ALLOW_REPLY_TO_ESCALATION_MAILS	replies will append to issue
\$ALWAYS_SHOW_DESCRIPTION_STAMP_IN_EMAILS	normally user/date info is shown only if multiple descriptions are in the mail
\$APPEND_STATUS_TO_ESCALATION_EMAIL_SUBJECT	puts "STATUS=xxx" at the end of the subject line where "xxx" is the current status of the ticket
\$CUSTOMER_PREPEND_EMAIL_SUBJECT	put these words at start of subject for mails going to customers
%DATE_FORMAT_FOR_EMAIL	hash of project numbers giving the date format to use for each project: 'SYSTEM', 'american', 'european', or 'iso'
\$DONT_USE_BACKGROUND_MAIL	send emails immediately instead of scheduling for later
\$EMAIL_MARKER_LINE	"reply above this line" first line
\$EMAIL_MARKER_LINE_EXTRA	"reply above this line" second line
\$EMAIL_SUBJECT_RECORD_NAME	what to put in the subject lines of emails instead of "ISSUE"; can only be letters and numbers.
\$OUTGOING_MAIL_CHARSET	use this character set for outgoing mails
\$REPLY_DEFAULT	reply-to address to use if none specified in \$CMI/etc/MRMultiIncoming or \$CMI/etc/MRmail_reply
\$SEND_NORMAL_MAILS_FOR_ESCALATIONS	use normal format for escalation emails instead of escalation format
\$SENDMAIL_PATH	where to find the sendmail program (usually /usr/lib/sendmail)
\$SMTP_PORT	(MS Windows only?)
\$SHORT_WIRELESS_ESC_EMAIL	don't put project/issue info in wireless emails
\$WIRELESS_PAGER_MAIL_ONLY_REPLY_INFO_IN_SUBJECT	subject is only "ISSUE=xxxPROJ=yyy"
\$WIRELESS_PAGER_MAIL_TOTAL_CHARACTER_LIMIT	restrict wireless and pager email messages to this length
\$USE_BACKGROUND_MAIL	send emails at regularly scheduled times rather than immediately

Note

This table does not include all of the variables it should, although it is more complete than it was in October 2004.

Table 5. MRlocalDefs variables - logging and debugging

Variable	Description
\$AUTO_REPORT_DEBUG	write debugging information about auto-run reports
\$AUTOLOAD_TIME_LOGGING	logs each subroutine which is autoloaded
\$DEBUG	general low-level debugging variable, effect varies from page to page
\$DEBUG_AB_HOME	write debugging information to <code>\$CMI/cgi/debug.txt</code> about address book home page
\$DEBUGABLOAD	write debugging information about importing address book data
\$DEBUG_ASSETS	write debugging information about FootPrints' interaction with Centennial Discovery
\$DEBUG_CHANGEMANAGE	write debugging information about change management to <code>\$CMI/etc/debugging/CMDebug.txt</code>
\$DEBUG_CHAT	write debugging information about chat package execution
\$DEBUG_CHECKBACKGROUND	adds debugging information to web pages when doing checkbacks
\$DEBUG_CHECKBACK_APPLET	causes client side checkback Java applet to enter debugging mode
\$DEBUGDBLOAD	write debugging information about importing issues data
\$DEBUG_DESC	write debugging information about the description when creating new issues to <code>\$CMI/cgi/debug.txt</code>
\$DEBUG_DISCOVERY	same as <code>\$DEBUG_ASSETS</code>
\$DEBUG_FIRST_CALL_RESOLUTION_REPORT	includes debugging information in the first call resolution report for the user specified by this variable
\$DEBUG_FLASHBOARD	puts debugging information about the flashboard into the flashboard
\$DEBUG_HISTORY_REPORT	includes debugging information in the history report for the user specified by this variable
\$DEBUG_IMAP	(obsolete - replaced by <code>\$DEBUG_INCOMING</code>)
\$DEBUG_INCOMING	write debugging information about retrieving POP and IMAP email, and about processing of the retrieved messages
\$DEBUG_INCOMING_NO_STAMPLINE	cause <code>\$DEBUG_INCOMING</code> not to write timestamp/pid lines
\$DEBUG_LANSURVEYOR	write debugging information about FootPrints' interaction with LANSurveyor; set to 'full' to avoid 2000 char limit
\$DEBUGLDAP	write debugging information about LDAP operations
\$DEBUG_NTMAIL	write debugging information about retrieving incoming POP mail under Microsoft Windows
\$DEBUG_OLINE	enable <code>O_LINE</code> debugging (debugging what has changed?) when editing issues via the web
\$DEBUG_OUTLOOK	(obsolete - replaced by <code>\$DEBUG_INCOMING</code>)
\$DEBUG_POP	(obsolete - replaced by <code>\$DEBUG_INCOMING</code>)

Variable	Description
\$DEBUG_PROJECT_COPY	writes debugging information about copying an existing project to the file \$CMI/etc/debugging/ProjectCopyDebug <N> .txt
\$DEBUG_PROXY	write debugging information about HTTP proxying
\$DEBUG_REASSIGN	write debugging information about changing assignments when editing issues
\$DEBUG_RECEIVED_VS_CLOSED_REPORT	includes debugging information in the received vs closed report for the user specified by this variable
\$DEBUG_RESOLUTION_RATE_REPORT	includes debugging information in the resolution rate report for the user specified by this variable
\$DEBUG_RESULTS	put spell check page into debug mode
\$DEBUG_ROLES	write debugging information about user roles
\$DEBUG_ROUND_ROBIN	writes debugging information about round robin assignments to the file \$CMI/etc/debugging/RoundRobinDebug <N> .txt
\$DEBUG_SENDMAIL	write debugging information about outgoing email
\$DEBUG_SENDMAILNT	write debugging information about outgoing email (Microsoft Windows)
\$DEBUG_SLA_DUETIME	display low-level debugging information about service level agreement deadlines
\$DEBUG_SMS	set to 'FILE' to write information about FootPrints' interaction with Microsoft System Management Server to \$CMI/etc/debugging; anything else display the information
\$DEBUG_SPLIT	display low-level spell check debugging information
\$DEBUG_TIME_OUT	write debugging information about subtracting time spent in certain statuses to \$CMI/cgi/debug_time_out.txt (when doing searches)
\$DEBUG_TURN_AROUND_REPORT	includes debugging information in the turn around report for the user specified by this variable
\$DISABLE_SECURITY_LOGGING	turn off security and query logging
\$DONT_WARN_ABOUT_DEBUGGING	supress debugging warning pop-up shown when agents log in
\$INCLUDE_PASSWORD_IN_DEBUG	write the passwords out instead of censoring them
\$LINK_DEBUG	display debugging information about dynamic issue linking when editing issues
\$LOG_FILE	defaults to \$CMI/etc/FPlog.txt
\$LOG_FILE_LOCKING	log who locks what files when
\$LOG_FILE_SIZE_LIMIT	defaults to 1 MB
\$LOGGING_ON	writes a lot of information, including file open/locks/unlocks and errors for background processes, to file specified by \$LOG_FILE
\$NO_PASSWORDS_IN_DEBUG_MANAGER	don't let system administrators set \$INCLUDE_PASSWORD_IN_DEBUG via the Debug Manager
\$NO_LOG_SENT_MAIL	don't keep list of sent messages in \$CMI/etc/sent_mail_log.txt
\$SCRIPTDEBUG	writes %in and stack trace for each script that is run
\$SUB_LOGGING	enables logging for calls to subroutines (time, names)

Variable	Description
<code>\$\$SUB_LOGGING_CALLER_DETAILS</code>	include in the subroutine log who called each subroutine
<code>\$\$SUB_LOGGING_IO</code>	enables logging of subroutine arguments and return values
<code>\$\$SUB_LOGGING_FILESIZE_LIMIT</code>	defaults to 2 MB
<code>\$\$SUB_LOGGING_NO_BUFFER</code>	write each subroutine log line immediately
<code>\$\$SUB_LOGGING_PAGES</code>	space separated string of pages for which to do sub logging; empty string means all
<code>\$\$SUB_LOGGING_USERS</code>	string of space separated users for whom to do sub logging; empty string means everyone
<code>\$TIMEDEBUG</code>	displays low-level debugging information about MRCalendarLib.pl
<code>\$TIME_LOGGING</code>	causes <code>\$CMI/etc/debugging/TimeLogging.txt</code> file to be rotated/removed automatically
<code>\$TIMEZONE_DEBUG_ON</code>	causes timezone conversion error messages to be displayed
<code>\$debugText</code>	display debugging information about use of language files

Note

This table does not include all of the variables it should, although it is more complete than it was in October 2004.

Table 6. MRlocalDefs variables - other

Variable	Description
\$ACCESS_DELAY	wait after updating issue before sending email in order to give Microsoft Access enough time to update the data
%ADD_TIMESTAMP_TO_FIELDS	hash of project IDs, each value is a reference to an array of field names for which FootPrints should add timestamps; see \$CMI/db/MASTER <x>/MR/timestampFields
\$AGENT_EDIT_MASTERSUBMEMBER	allow users to edit tickets they otherwise couldn't solely because they have the or edit privilege on another ticket in the master/subtask relationship
\$AGENT_SUBMIT_STATUS	is the default status for tickets submitted by agents
\$AGENT_VIEW_MASTERSUBMEMBER	allow users to view tickets they otherwise couldn't solely because they have edit privilege on another ticket in the master/subtask relationship
\$ALLOW_OTHER_PERLS	don't complain if the version of Perl is not what FootPrints is expecting
\$ALTERNATE_TABLESPACE	Oracle only: use a different tablespace instead of "FOOTPRINTS"
\$ALWAYS_CC	Make "always CC" the default on issue creation/edit pages
\$ALWAYS_COPY_TICKET_ON_ESC	copy an issue when escalating, even if it has already been copied by an escalation
\$ASSETS_BASE_URL	URL to use for asset management software, overrides the value specified in the file \$CMI/etc/MRdiscoveryConfig.txt
\$ASSETS_USERAGENT_TIMEOUT	controls time-out of HTTP requests from FootPrints to the asset management and other software
\$AUTO_ADD_NO_RESTRICT	permits system administrators to configure auto-adding users to projects even if the address books and primary keys do not match
\$AUTO_ASSIGN_STAY_REQUEST	don't change the status of requests to 'Open' when auto-assigning
\$BALLOT_SECRET	password for creating the nonce used to validate votes
%AUTO_FILL_FIELD_SCRIPT	key is field name, value is reference to hash containing information about Javascript used to automatically fill in the fields
\$SAVECOMM_ON_EXTERNAL_SERVER	prevents FootPrints from accessing the AveComm chat configuration file \$CMI/cgi/global.asa if it does not reside on the FootPrints server
\$BACKTRACKVIEWMINE	enabled old "'mine' means 'assigned to customer'" functionality
\$CCS_ON_HTML_FORM	put a CC field on the HTML form that gets emailed to users
\$CHECKBACK_IDLE_TIME_LIMIT	value is in minutes from 1-15; defaults to 5 minutes. checkbacks will no longer happen for users after they are idle for more than 5 times this long
%CREATE_IN_OTHER	add button to toolbar to quickly create new issues in other projects, key is project number, value is button text
\$CUSTOMER_CANT_REOPEN	make customer email replies to closed tickets result in an error
@CUSTOMER_NAME_FIELDS	defaults to ('First_bName', 'Last_bName')

Variable	Description
\$DBI_RETRY_COUNT	how many times to re-try failed database transactions (does not apply to normal queries, only to atomic transactions)
\$DBI_RETRY_SECONDS	how long to wait between each database transaction re-try (wait a random amount of time up to this number of seconds)
\$DEFAULT_PRECISION	changes the default precision in the field editor (normally 20)
\$DEFAULT_REPORT_WRAP	overrides the project preference for the number of characters at which to wrap descriptions and multiline data
\$DEFAULT_SCALE	changes the default scale in the field editor (normally 2)
@DELETE_FILES_DAILY	list of additional files to delete each day
\$DISABLE_ATTACHMENTS_FOR_CUSTOMERS	don't let customers add attachments to issues
\$DISABLE_XHOST_CALL	stops FootPrints from running " xhost +localhost " (xhost call needed for charts to work under Unix)
\$DONT_ALLOW_ATTACHMENT_PATTERN	reject attachments which match this regular expression
\$DONT_ASSIGN_AGENT_FOR_REQUEST_EDIT	prevent agents editing requests from automatically being assigned to the issue
\$DONT_USE_SCHEDULE_SERVICE	disable all automatically run scripts
\$DOT_FORWARD_SUPPORTED	enable <code>.forward</code> files
\$DO_NOT_CHANGE_UPDATEDATE_ON_EDIT_OF_CLOSED	when editing closed tickets, keep the old update date and time
\$DO_NOT_SEARCH_FOR_CUSTOMER_REAL_NAME	use customer usernames everywhere instead of real names
\$DUMP_FILE_RETENTION_DAYS	automatically delete dump files from the \$CMI/db/MASTER <N>/MR/DumpsFromPurges directory (created when purging and/or archiving issues) after this number of days
\$EMAIL_DENY_MUST_MATCH_EXACTLY	disable partial email address matching
\$EMAIL_MATCHING	disables "rough matching" of email addresses if set to 'exact'
\$ENABLE_AUTO_CONVERT_TO_GLOBAL	when creating a new global link to an issue, if the issue being linked to is not already a global issue, automatically make it a global issue (and then do the link) instead of giving an error
\$ENABLE_LINK_TO_CLOSED_GLOBAL	when creating a new global link to an issue, allow the user to type in the issue number of a closed global issue in addition to giving the normal drop down menu of open global issues
\$ESCALATED_AS_COPY_SUBMITTER	when an escalation copies an issue, create the copy as the user who originally submitted the issue instead of using "Escalated" as the user.
\$EXACT_ESCALATION_TIMES	make every escalation run every time
%FPAB_EXACTSEARCH	hash of project numbers containing hash of address book field names for which exact searches (instead of substring searches) should be done
\$FP_BASE_URL	override contents of \$CMI/etc/URLroot
@GLOBALLINK_STATUS_CHANGE_VALUES	change global links statuses only when it is one of these
\$GLOBAL_DBD_PASSWORD	specifies an alternate password (unencrypted) to use instead of the easy-to-guess default database password
\$GLOBAL_DBD_STRING	specifies a special DSN to use when connecting to the database

Variable	Description
\$GLOBAL_DBD_USERNAME	specifies a special username to use when connecting to the database instead of the default username that FootPrints usually uses
\$HIDDEN_PRIORITIES	string containing space separated list of priorities that can only be set by escalations
\$HISTORY_ALT	in the history report, show the current assignees instead of the assignees from the time when the issue went into the ending status
\$JAVA_DIR	where Java is installed
\$KEEP_CLOSED_FOR_REPLY_TO_CLOSED	customer can reply via email to closed tickets but they remain closed
\$KEEP_CLOSED_FOR_REPLY_TO_CLOSED_PATTERN	if the email sender's name matches this regular expression, set \$KEEP_CLOSED_FOR_REPLY_TO_CLOSED
\$LICENSE_FILE_DUPLICATES_CHECK	each login removes duplicates from license file
%MANDATORY_FOR_CLOSED	make some optional fields mandatory for closed tickets, key is project number, value is reference to an array of field name strings
%MASS_UPDATE_STATUS_EXCLUDE	don't allow statuses to be changed to these via quick action checkboxes, key is project id, value is reference to array of status name strings
\$MRPROXY_SECURE	enable extra security check for Discovery URLs
\$MULTISERIES_CHART_OTHER_CUTOFF	defaults to 10
\$MYSQL_HOST	hostname for MySQL database server
\$NO_ASSIGN_TO_EMAIL_SUBMITTER	keep issues submitted via email by agents to being assigned to the agent who submits it
\$OEM	defaults to "FootPrints"
\$PROD_NAME	defaults to "FootPrints"
\$PROXY_USERAGENT_TIMEOUT	controls timeout for FootPrints' HTTP proxy mechanism
%REQUEST_IN_OTHER	like %CREATE_IN_OTHER, but for customers not agents
@SCHEDULE_MANUALLY	list of scripts that should not be run automatically
\$\$SOLUTION_PRIORITY	all solutions get this priority
\$SMS_NAMESPACE	Explicitly specify the SMS namespace instead of querying SMS for it ("\\Machine" + "\\root\SMS\site_" + sitecode)
\$USE_DESCRIPTION_TABLE_JOINS	disable creation of mrALLDESCRIPTIONS database column when upgrading from 6.0 to 6.5 and when creating new projects; also disable use of this column when searching
\$VOTING	turns on voting to allow auto-assigned agents to approve/disapprove of issues
\$X_SERVER_BIN	directory of X Windows binaries

Table 7. \$CMI/etc directory contents

File or directory name	Description
.License	List of users currently logged in, their MRPs, and their project numbers (Unix/Linux only)
ABlist.txt	information about all address books (number, location, name, etc.)
AddedGenerationZero.txt AddingGenerationZero.txt	flag file to indicate that a generation 0 for each project has been (is being) added for each mrID to MASTERx_DESCRIPTIONS (part of upgrade from FootPrints 6.0 to 6.5)
Attachment_References	directory containing file for each project saying what attachments that project references (formerly \$CMI/html/tmp/Attachments/References)
Attachment_Root	text file containing the name of the directory where attachments for all projects are stored
AutoAssignIncomingEmail.yaml	Only used if \$ASSIGN_INCOMING_BASED_ON_TO is set in MRLocalDefs. Contains a hash mapping email addresses to team names and from email addresses, and reply email addresses. If an incoming email is set to one of the email addresses (keys), it will be auto-assigned to the specified team.
DefaultKBSearch.txt	knowledge base search query to use if no project-specific query exists
Dictionaries	directory containing a file for each user of words that user has added to their spell-checking dictionary
FPMEhold FPMEhold.txt FPMEincoming FPMEinlastrun.txt FPMEoutgoing FPMEoutlastrun.txt	files and directories for getting and sending Microsoft MAPI email
FPPasswdlog	file used for logging if \$LOG_FILE is not defined but \$LOGGING_ON is set
Flashboard	directory containing flashboard files for each user
GroupRO_Default_Message.txt GroupRS_Default_Message.txt	welcome pop-ups for the home pages of RO, RS, etc., users
InMailvacation.txt	file of phrases that indicate vacation or out-of-office messages; incoming mail with these phrases in their title are rejected
IncomingIntro	contains text to email to new mail users and include in help messages (FootPrints 6.0 and later)
LDAPAutoAddProfiles	associates auto-add profiles with LDAP DN criteria.
License	contains the FootPrints license code
License.last	previous (old) license code
MRDBpasswd	stores an encrypted password for FootPrints to use in accessing the SQL database; if this file does not exist, an easy-to-guess default password is used instead.
MRInternalFields.txt	contains system-wide names for internal FootPrints fields
MRInternalFields_default.txt	contains defaults for system-wide names for internal FootPrints fields
MRchat_config	contains system-wide chat configuration
MRdatefmt	contains system-wide default date format
MRdefaultLanguage	contains the system-wide default language

File or directory name	Description
MRdiscoveryConfig.txt	configuration file for asset tracking add-on
MRlinksFile	maps Direct Link (DL) IDs and Short Link names onto query string parameters or full URLs
MRmailN	contains project number of project to which incoming email should be delivered by default
MRmail_debug	tells whether email debugging is enabled
MRmail_domain	contains the default mail domain name
MRmail_name	contains the default "From" name for those projects not listed in MRMultiIncoming
MRmail_reply	contains the default reply address for those project not listed in MRMultiIncoming contains email configuration for each project
MRmassmail_config	contains email batch size and delay parameters
MRoutgoing_enabled	enables/disables outgoing email for the system
MRpasswd	contains user information and preferences
MRpasswd_users	directory tree containing user information for auto-added users
MRprojects	lists each project and options
MRproxyConfig.txt	contains proxy information for asset management system
MRsend_intro	contains text to email to new mail users and include in help messages (pre FootPrints 6.0)
MRsendmail_name	contains name of program to invoke on Unix systems to send email
MRusers	directory containing file for each user with user's per-project membership and preferences
MRusers_projects	directory containing file for each project listing that project's users and the users' preferences; automatically generated from files in MRusers directory
MRwireless_pager_list	contains a list of domains for which wireless/pager email formatting should be used
Quick_Descriptions	directory containing a YAML file for each username which holds that user's quick descriptions.
Search	contains definitions of all searches for all projects
SearchList	contains short list of searches for all users and projects
SecurityLogging.txt	YAML file containing the security and query logging settings from the Administration —> System —> Logs page.
Semaphore	temporary file used for doing query logging for shared customer accounts
SpamWatch.txt	controls whether FootPrints will reject incoming mail with the X-Spam-Warning, X-Spam-Flag, or X-Spam-Status headers
URLroot	base URL used for all FootPrints accesses
V60toV65completed.txt V65toV66completed.txt	flag file to indicate that all changes that need to be made when upgrading from 6.0 to 6.5 (or 6.5 to 6.6) have been made
WarnAddress	list of users to usernames to notify of errors when project ID is not known
adminErrorNotify.txt	contains username of administrator to be listed on error pages

File or directory name	Description
allchPathFile.txt	path of AllChange software
archive	directory of files for each scheduled archiving job
attachmentsConfig.txt	system-wide attachment settings (file/directory size, extensions, etc.)
autorun_reports.txt	list of cross-project auto-run reports
aveCommConfig.txt	aveComm remote-control and chat configuration
aveCommPending.txt	list of pending aveComm chats
aveCommPopupHash	list of usernames for whom chatting is disabled
backErrs	log file containing error messages from background processes (Microsoft Windows only)
backgroundMails.txt	list of all mail messages waiting to be sent
badMails.txt	list of backgrounded mail messages which could not be sent due to FootPrints problems
chat	directory containing chat data files
cookie.cfg	controls whether "remember my password" cookies are enabled
cookieIDs	list of cookie IDs for "remember my password" cookies
crossProjectSearches	directory containing files with definitions of cross project searches
debug_mode.txt	contains list of users for which homepage/report debugging is enabled
debugging	directory containing files generated by turning various sorts of debugging on
deleteUsers.dat	list of deleted users scheduled to have their assignments changed to another user
foot.lic	like .License but for Microsoft Windows systems
imap.txt	IMAP account information for incoming email for each project
lastprojnum	highest used project number
ldap.cfg ldap2.cfg	list of LDAP bases to use
ldapattr.txt	LDAP attribute to use for binding
ldaplogin.txt ldaplogin2.txt	DN and password for binding to LDAP
licenseUsage.txt	if \$LICENSE_USAGE_REPORTING is set in MRLocalDefs, a license usage report will be added to the end of this file every 6 minutes.
logViewTimes.txt	list of when each system administrator has viewed the system-wide FootPrints log
logoSettings.txt	system-wide defaults for project/product logo
massmail.txt	list of mass emails scheduled to be sent
newInstall	the presence of this file causes the user to be sent to the project wizard when they log in
no_accept.txt	system-wide list of partial email addresses from which incoming email should be rejected

File or directory name	Description
no_send.txt	system-wide list of partial email addresses to which no outgoing email should be sent
ntdomain.cfg ntdomain2.cfg	contains Microsoft Windows NT domain authentication information
only_accept.txt	system-wide list of partial email addresses; if the file contains any addresses, email incoming email will be rejected unless it matches one of the addresses in the file
only_send.txt	system-wide list of partial email addresses; if the file contains any addresses, outgoing email will only be sent if it matches one of the addresses in the file
passwordQuestions.txt	list of questions for users to respond to with hints to retrieve forgotten passwords
pop.txt	POP account information for incoming email for each project
popups	directory containing notifications for users and administrators
projectLogins	directory containing a file for each user with the last 5 projects they logged into
pwauth.cfg	file containing details on what type of authentication FootPrints is using
rates	directory containing a file for each user with billing rate information
real_names	directory containing a file for each project mapping usernames onto real names
remote_control.txt	contains message sent to users when an agent requests remote control of their workstation
roleLicenseCheck.txt	cached result of license check to see if roles are supported
roles	directory containing file for each system-wide role which has been modified by the server administrators
sampleBase.txt	standard LDAP base name
sent_mail_log.txt	list of mail messages that have been sent
skippedIncomingMessages.txt	list of message-ids and other details of messages that should not be retrieved
smsConfig.txt	Microsoft SMS configuration information
systemFonts	list of fonts that Java knows about
systemLog.txt	FootPrints system-wide log file
time	files for each user containing their active and idle times
tmpSurvey	temporary file used by \$CMI/cgi/MRSurveyLib.pl
unsentMails.txt	list of background mails not sent due to errors
watch	directory containing a file for each "job" started by FootPrints in order to keep track of processes and kill them if they run too long (Unix/Linux only)
watchLog.txt	log file containing a list of processes which are killed
wizardtemplates	directory tree containing text files defining project creation wizard templates

Table 8. \$CMI/etc/MRprojects fields

Field	Description	Default
\$ProjPrefs[0]	project ID	
\$ProjPrefs[1]	CMMASTER (database directory)	
\$ProjPrefs[2]	project name	
\$ProjPrefs[3]	is project deleted? (0 no, 1 yes)	
\$ProjPrefs[4]	(now part of roles – used to be “taking (0 off, 1 on)”)	
\$ProjPrefs[5]	(now part of roles – used to be “edit (0 off, 1 on)”)	
\$ProjPrefs[6]	project administrator email	
\$ProjPrefs[7]	notify: see \$CMI/cgi/MRchangeUrgent.pl for codes	
\$ProjPrefs[8]	definition of urgent	
\$ProjPrefs[9]	criteria for sending mail	
\$ProjPrefs[10]	(now part of roles – used to be “customer priorities”)	
\$ProjPrefs[11]	mail to ProjAdmin	
\$ProjPrefs[12]	maximum priority	10
\$ProjPrefs[13]	assignment options: 0/1/2/3/4 = Creator/None/All/Creator’s Team/Creator and Creator’s Team	0
\$ProjPrefs[14]	AB MASTER number	
\$ProjPrefs[15]	assign request (0 or 1)	
\$ProjPrefs[16]	group view all tickets	
\$ProjPrefs[17]	used by Portera to store project owner’s user_id	
\$ProjPrefs[18]	used by Portera to store project billing id	
\$ProjPrefs[19]	auto time tracking: 0=manual, 1=automatic	
\$ProjPrefs[20]	customer name format	
\$ProjPrefs[21]	used by Portera to allow editing of original description	
\$ProjPrefs[22]	default priority	1
\$ProjPrefs[23]	time tracking: 1=off, 0=optional, 2=mandatory	
\$ProjPrefs[24]	email field validation	
\$ProjPrefs[25]	primary key field of Address Book	Email_bAddress
\$ProjPrefs[26]	who can edit issues via incoming email	
\$ProjPrefs[27]	is a key required for group login	
\$ProjPrefs[28]	(now part of roles – used to be “are group users allowed to chat?”)	
\$ProjPrefs[29]	(now part of roles – used to be “is chat allowed in this project?”)	

Field	Description	Default
\$ProjPrefs[30]	(now part of roles — used to be “is remote control allowed in this project?”)	
\$ProjPrefs[31]	default language	
\$ProjPrefs[32]	auto assign tech to a request after they’ve replied	
\$ProjPrefs[33]	auto-select status after tech responds to request	Open
\$ProjPrefs[34]	FootPrints/AllChange Integration	
\$ProjPrefs[35]	mail for endusers, same as the old \$ProjPrefs[9]	
\$ProjPrefs[36]	number of columns for project/ab field display	3
\$ProjPrefs[37]	auto assign on create+edit or create only (0 = create, 1 = create & edit); defaults to 1	1
\$ProjPrefs[38]	auto assign: overwrite or append to existing list (0 = overwrite, 1 = append)	
\$ProjPrefs[39]	default screen size/resolution for group users	LARGE
\$ProjPrefs[40]	auto-select status after customer responds via email	
\$ProjPrefs[41]	manually enter start date and time for time tracked	
\$ProjPrefs[42]	(now part of roles – used to be “limit Agents to seeing their assignments (0 off, 1 on)”)	
\$ProjPrefs[43]	use work calendar for reports on ticket age and time to close	
\$ProjPrefs[44]	who to mail by default for new solutions	
\$ProjPrefs[45]	mandatory field rules for incoming email	
\$ProjPrefs[46]	closing of all subtasks process	1
\$ProjPrefs[47]	Knowledge-Paks Online	
\$ProjPrefs[48]	ticket submission tracking	
\$ProjPrefs[49]	WYSIWYG enabled for this project. ON/OFF	
\$ProjPrefs[50]	edit locking timeout period	30
\$ProjPrefs[51]	present customers with a choice of projects when they first log in	
\$ProjPrefs[52]	rows in description textarea	10
\$ProjPrefs[53]	columns in description textarea	85
\$ProjPrefs[54]	Advanced Live eSupport with aveComm	
\$ProjPrefs[55]	enabling use of global issues	1
\$ProjPrefs[56]	auto-select status for email reply to Closed ticket	Open
\$ProjPrefs[57]	dropdown/multi widths: 0 = same width (defined in stylesheet), 1 = as long as needed	
\$ProjPrefs[58]	disallow time tracking pause (0=allow pause, 1=no pause)	
\$ProjPrefs[59]	single-frame interface for customers	
\$ProjPrefs[60]	accept new incoming mail submissions from customers	yes

Field	Description	Default
\$ProjPrefs[61]	accept new incoming mail submissions from agents	yes
\$ProjPrefs[62]	if value present, it is a survey project (contains a list of projects, separated by comma, that it is accepting surveys from)	
\$ProjPrefs[63]	number of hours in a work day	24
\$ProjPrefs[64]	display of time calculations in reports days, hours, minutes	DHM
\$ProjPrefs[65]	status of source issue on move	
\$ProjPrefs[66]	auto check availability	
\$ProjPrefs[67]	RightAnswers search field	
\$ProjPrefs[68]	auto-changing subtask statuses	
\$ProjPrefs[69]	is a survey project	
\$ProjPrefs[70]	auto-select status after agent responds to an issue through email (except request/closed)	
\$ProjPrefs[71]	enable time tracking comments	
\$ProjPrefs[72]	when to send emails to CCs, similar to \$ProjPrefs[9] and \$ProjPrefs[35]	
\$ProjPrefs[73]	??? (apparently unused)	
\$ProjPrefs[74]	time tracking entered as hours and minutes (0) or as an end time (1)	
\$ProjPrefs[75]	field permissions for incoming email (agents)	allAgentRoles
\$ProjPrefs[76]	field permissions for incoming email (customers)	allRoles
\$ProjPrefs[77]	cross-project copy subtasks yes/no	
\$ProjPrefs[78]	append GMT offset to email	
\$ProjPrefs[79]	show "Powered by FootPrints" in single-frame interface	
\$ProjPrefs[80]	default width of assignee picker	170
\$ProjPrefs[81]	hours in "day" for SLAs	24
\$ProjPrefs[82]	change priority of subtasks when master changes	

Table 9. \$CMI/etc/MRpasswd and \$CMI/etc/MRpasswd_users/*/*/* fields

Field number	Description
0	username
1	encrypted password
2	user type in default project
3	real name, or hint for forgotten password question
4	default project
5	email address, or answer for forgotten password question
6	time zone
7	popup help (Portera - not used by footprints?)
8	pager address
9	wireless address
10	send notification mails to primary email address
11	send notification mails to pager address
12	send notification mails to wireless address
13	default language
14	screen resolution
15	concurrent flag
16	is this an auto-added user? name of auto-added profile or 1 for default profile
17	show error alerts flag (system administrators only)
18	date format (american/european)
19	spell check preferences
20	allow multiple windows for creating and editing tickets
21	toolbar static or dynamic
22	WYSIWYG enabled
23	508 Compliance
24	agent edit last description
25	font Size
26	hide project totals/globals tables on homepage
27	display all descriptions in text area
28	authentication method for login by user (PRIMARY or SECONDARY)
29	checkbox method ('applet' or 'no-applet' -- blank is the same as applet')

Field number	Description
30	confirm when using Quick Issue template from create/edit (blank or 'off')
31	open a new window to generate attachments (blank or 'off')

Table 10. \$CMI/etc/MRusers/* fields

Field	Description
(first field of each line in the file)	(project number - removed when the line is read)
\$UserPrefs[0]	user type
\$UserPrefs[1]	number shown at once in list on home page
\$UserPrefs[2]	direction (primary,secondary) 0=descending, 1=ascending
\$UserPrefs[3]	order by (primary,secondary)
\$UserPrefs[4]	which field (OBSOLETE)
\$UserPrefs[5]	display (OBSOLETE)
\$UserPrefs[6]	operator (OBSOLETE)
\$UserPrefs[7]	object (status or priority) (OBSOLETE)
\$UserPrefs[8]	date options (OBSOLETE)
\$UserPrefs[9]	generations/details (OBSOLETE)
\$UserPrefs[10]	frames/non-frames
\$UserPrefs[11]	password verification on/off
\$UserPrefs[12]	default time-tracking pay-rate for this project (other rates are in \$CMI/etc/rates/username)
\$UserPrefs[13]	(A)ssignments or (P)refs or (S)aved search. there are many more. See \$CMI/cgi/MRchange_preferences.pl
\$UserPrefs[14]	homepage autorefresh interval
\$UserPrefs[15]	saved search name for \$UserPrefs[13]
\$UserPrefs[16]	default decision-field value (NO LONGER USED AS OF V6.0)
\$UserPrefs[17]	home page field layout string
\$UserPrefs[18]	internal user chat prefereces
\$UserPrefs[19]	user's role
\$UserPrefs[20]	prefill tickets with agent's contact info
\$UserPrefs[21]	comma separated list of fields to NOT include on homepage drop down
\$UserPrefs[22]	default drop down view of agent availability popup screen
\$UserPrefs[23]	default start time for agent availailability popup screen
\$UserPrefs[24]	UserID of user's supervisor

Table 11. \$CMI/db/MASTER*/MR/* directory contents

File or directory name	Description
ArchiveProject	if the project is a project that archives old issues for another project, this file contains the project ID of the "source" project
ArchiveTasks.yaml	contains configuration for regularly scheduled (recurring) tasks to archive or purge issues from the project
Autoassign	contains a list of auto assignment criteria for issues of the form FIELD:CHOICE:ASSIGNEE (if FIELD = CHOICE, assign ASSIGNEE)
Calendar	contains work week schedule and exceptions
CalendarOptions.txt	contains calendar configuration options
Colors	fragment of a Perl script that sets variables defining the project's color settings
CreateOrder	lists order of dialogs on the issue create/edit pages
Dependencies	(obsolete) version 5.5 dependency file, limited to a single decision field
Dependencies.enum	(obsolete) version 5.5 dependency choice fields file, limited to a single decision field
Dependencies.idx	Index of all dependencies
Dependencies_ <DECISIONNAME>	dependency file for dependency <i>DECISIONNAME</i>
Dependencies_ <DECISIONNAME>.enum	dependent choice fields file for dependency <i>DECISIONNAME</i>
DependenciesOptions_<DECISIONNAME>	contains miscellaneous dependency options
DependencyGroup_ <GROUPNAME> .dat	contains dropdown dependency <i>GROUPNAME</i>
Dependency_popup_create_<ROLENAME> .storable Dependency_popup_edit_ <ROLENAME> .storable	temporary cache files to speed up popup dependencies
DependencyGroups.idx	index file for dropdown dependencies
Descriptionorder	if this file exists, descriptions will be shown in ascending order (oldest first) instead of the default descending order
Detailorder	contains the order of sections in the ticket details
DumpsFromPurges	directory containing issues dumped to files during a purge
Escalation	contains information about each escalation for the project
Escalation.preSLA	backup copy of Escalation file created when setting up service level agreements
Escalation.email	default escalation email template used if a custom template does not exist
EscalationEmails	directory containing escalation custom email templates
EscalationEmailsIndex.yaml	contains information about different email configurations
ExcludeFor_ABFIELDS_	contains fields to exclude for the address book
ExcludeFor_SOLVED_	contains list of project fields not to include in solutions

File or directory name	Description
ExternalTableLookup_ <NAME> .yaml	contains configuration for Dynamic SQL Database Link
Group <TYPE> _Custom_Message.txt	contains project-specific pop-up welcome messages for users of group <i>TYPE</i> (RO, RS, etc.)
KBApproval	list of people who can approve knowledge base submissions for a project
KBSearch.txt	project-specific knowledge base search queries
Life.cycle	contains status in order from first to last for life of an issue
LogoSettings.txt	contains filename and options for project-specific logos
MASTER.dmp	created by running <code>\$CMI/cgi/mrDump.pl</code> manually
MR.db	(used in rebuilding database indexes?)
MRHomepageProjectTotals.txt	contains project total box configuration data
MRInternalFields.txt	contains project-specific names for internal fields
MRLOG	log of database transactions (Unix/Linux)
MRLOG_ <DATE>	log of database transactions (Microsoft Windows)
MRMasterPMs.txt	???, but only used if <code>\$USE_PM_SYSTEM</code> is set in <code>MRlocalDefs</code>
MRNUMBER	mrID of last issue created
MRmailList	list of addresses to which mail should always be sent
MRteams	file containing team definitions and memberships
PriWords	file containing words which correspond to each priority
RecordNameSettings.txt	contains project-specific names for issues (singular and plural)
SFIColorSettings.txt	colors for the project's eService Customer Portal
SLAdescription_ <ID>	contains the description text for the service level agreement denoted by <i>ID</i>
SLAsetup.yaml	service level agreement configuration for the project
Schema	contains definitions of project fields for the project
SchemaFormat.txt	formatting information for each project field
SchemaOrder	order project fields are displayed
SurveysSent	directory containing a file for each email address that a survey has been sent to listing times surveys were sent
TIME.dmp	created by running <code>\$CMI/cgi/mrDump.pl</code> manually
Titles	list of dropdown choices for the project's Title field
VoteFile	keeps track of votes if <code>\$VOTING</code> is enabled
_SURVEY_response.html	saved response to a survey
allchprojPathFile.txt	path to AllChange software

File or directory name	Description
alldescriptions_enabled.txt	flag file indicating that the MASTER <N> .mrALLEDSCRIPTIONS database column was populated during the upgrade to FootPrints 6.5
autoCopyConditions.txt	list of cloning conditions for auto-copying issues (currently not accessible via web)
autorun_reports.txt	list of reports to be auto run and their schedules
create_date	date project was created
cust1FrameHpSettings.txt	YAML file containing the configuration for the single-frame eService Customer Portal home page for customers
disableQuickCount.txt	if this file exists, the number of total matching issues is omitted from the homepage or search results page in order to improve performance
euHomepageFieldLayout	home page field layout for end users
fieldLengths	contains minimum and maximum lengths for project fields that have them
fulltext_enabled.txt	if this file exists, FootPrints assumes that the database tables for the project have fulltext indices created on their multi-line text fields (MySQL and MS-SQL only)
loadDump.log	error messages from loading an mrDump generated file into the database
mailTemplate_ <TEMPLATETYPE> <FORMATTYPE>	custom email template file - <i>TEMPLATETYPE</i> can be 'html' or 'text'; <i>FORMATTYPE</i> can be 'internalassign' (Assignees), 'externalassign' (End user), 'externaloverride' (cc:), 'internal_wireless_pager'
mailfile.txt	stores what to include in each type of email (for when custom email templates are not being used)
mrESCALATEDBY_added	flag file to indicate that MASTER <N> .mrESCALATEDBY database column was added the during upgrade to FootPrints 6.0c
mrRATEDESC_added	flag file to indicate that MASTER <N> TIMETRACKING.mrRATEDESC database column was added during the upgrade to FootPrints 6.6
no_accept.txt	project-specific list of partial email addresses from which incoming email should be rejected
no_send.txt	project-specific list of partial email addresses to which no outgoing email should be sent
only_accept.txt	project-specific list of partial email addresses; if the file contains any addresses, email incoming email will be rejected unless it matches one of the addresses in the file
only_send.txt	project-specific list of partial email addresses; if the file contains any addresses, outgoing email will only be sent if it matches one of the addresses in the file
projInfoSettings	information about the project splash screen / project info popup
projectGlobals.txt	cached list of global tickets in project (to speed up global ticket scoreboard on home page)
quickcreatefile.txt	contains templates for quick issues
quickDesc.txt	OBSOLETE (used for project-wide quick descriptions when this functionality was still controlled by MRlocalDefs; different file format than quickDescriptions.txt below)
quickDescriptions.txt	contains project-wide (non-user-specific) quick descriptions for this project
roleProperties_ <ROLENAME> .txt	each file contains role information for the project-specific role <i>ROLENAME</i>
rolesList.txt	list of all roles in the project

File or directory name	Description
savedMetrics	directory containing a file for each username (or “_SHARED_”) that contains the definitions for saved metric reports
templateinfofile.txt	list of (mail?) templates that have been uploaded to the project
temp_table_created.txt	flag file indicating that the MASTER <N> _TEMP database table has been created for use in change management
timestampFields	YAML file containing specifying which fields FootPrints should append timestamps to
unassigned_enabled.txt	flag file to indicate that MASTER <N> .mrUNASSIGNED database column was added during the upgrade to FootPrints 7.0

Table 12. \$CMI/db/ABMASTER*/MR/* directory contents

File name	Description
ABMASTER.dmp	created by running \$CMI/cgi/mrDump.pl manually
LDAPFieldNameMapping	associates LDAP attributes with FootPrints address book fields
MR.db	??? (maybe used in rebuilding database indexes?)
MRABRecordNameSettings.txt	contains address-book specific names for contacts
MRLOG	log of database transactions (Unix/Linux)
MRLOG_ <DATE_>	log of database transactions (Microsoft Windows)
MRNUMBER	number of the last contact created
ProjAndKeys.fp	maps project numbers to address book primary keys
Schema	contains definitions of address book fields for the address book
Schema.fp	schema file for FootPrints address books
SchemaFormat.txt	formatting information for each address book field
SchemaOrder	order in which address book fields are displayed
SearchAB	definitions of all searches for this address book
SearchListAB	list of all searches for this address book
abmasterMasterTableCreated	flag file to indicate that the ABMASTER <N> _MASTER database table exists
fieldLengths	contains minimum and maximum lengths for address book fields that have them
ldap.cfg	LDAP server configuration information
ldaplogin.txt	LDAP server authentication information
masterContactFields.txt	YAML file containing the master contact fields configuration
timestampFields	YAML file containing specifying which fields FootPrints should append timestamps to

Table 13. Schema file format

Field number	Description
0	project or address book field name
1	type (char, enum, date, http, or name of a field of type enum)
2	permissions (see tables below)
3	not used (always zero?)
4	more type information: multi: multiline char field or multi-choice field time: date-time field (field type = date) datetime: date-time link field (field type = http) check: checkbox field tracking: submissstion tracking field

Note

If advanced field permissions are being used, then roles are used for field permissions instead of the project field permissions values in the Schema file.

Table 14. Project field permissions in Schema file

Permissions value	Description
0	Internal / Optional
1	Internal / Mandatory
2	Public Read-only / Optional
3	Public Read-only / Mandatory
4	Public Write-once / Optional
5	Public Write-once / Mandatory
6	Public Read-Write / Optional
7	Public Read-Write / Mandatory
88	Public Read-Write-Edit / Optional
89	Public Read-Write-Edit / Mandatory

Note

If advanced field permissions are being used, then roles are used for field permissions instead of the address book field permissions values in the Schema file.

Table 15. Address book field permissions in Schema file

Permissions value	Description
0	Public / Optional
1	Public / Mandatory
6	Internal / Optional
7	Internal / Mandatory

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Examples

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Examples

Example 1. Minimalistic example of code to produce a web page by directly querying FootPrints' database

```
#!/usr/bin/perl
use DBI;

$FOOTPRINTS_SERVER = 'help.lsa.umich.edu';
$FOOTPRINTS_PROJECT_NUMBER = 16;
$DATABASE_DSN= "DBI:mysql:Footprints:$FOOTPRINTS_SERVER";
$DATABASE_USERNAME = 'lsait_unix_web';
$DATABASE_PASSWORD = 'XXXXXXXXX';

print << __END_OF_HEADER__
Cache-control: private, must-revalidate, no-cache
Expires: Mon, 16 Apr 1973 02:10:00 GMT
Pragma: no cache
Content-type: text/html

<html><head><title>FootPrints requests</title></head><body>
__END_OF_HEADER__
;

print "<h3>Public tickets in project ${FOOTPRINTS_PROJECT_NUMBER} on
${FOOTPRINTS_SERVER}:</h3><br />\n\n";

$dbh = DBI->connect( $DATABASE_DSN, $DATABASE_USERNAME, $DATABASE_PASSWORD );

$stmt = $dbh->prepare( "SELECT mrID, mrSTATUS, mrSUBMITTER from
Footprints.MASTER${FOOTPRINTS_PROJECT_NUMBER} WHERE mrSTATUS != \"Closed\" AND mrSTATUS !=
\"_DELETED\" ORDER BY mrID DESC;" );
$stmt->execute();

print "<table cellpadding=\"1\" cellspacing=\"4\">\n";
print "<tr><th>Ticket number</th><th>Status</th><th>Submitter</th><tr>\n";

while ( my @row = $stmt->fetchrow_array() )
{
    $row[1] =~ s/_/ /g; # Unescape spaces in mrSTATUS
    $row[1] = 'New' if $row[1] eq '_REQUEST_';
    print "<tr>";
    foreach my $data (@row) { print "<td>$data</td>"; }
    print "</tr>\n";
}

print "</table><br /></body></html>\n";
$dbh->disconnect();
```

Example 2. A typical FootPrints CGI

```
#!/usr/footprints_perl/bin/perl --
#--
#-- This is a special version of the file $CMI/cgi/MREditUser_Page.pl from
#-- FootPrints 7.0c. It has been adapted for use in the "FootPrints Under
#-- The Hood" presentation. Some comments have been added to point out
#-- various things about the typical content of FootPrints CGIs.
#--
#-- All comments that have been added for the purposes of this presentation
#-- start with "#--" instead of just "#".
#--

#
# Copyright 1996-2004 UniPress Software Inc.                #COPYRIGHT LINE#
#

package FP;

#
# MREditUser_Page.pl - select a user to edit
#

# chron:    20 June 1997 (begun)
#           20 June 1997 (last modified)
# copy:     Copyright 1996-97 UniPress Software, Inc.

require "MRlib.pl";          #-- Sets global variables, defines AUTOLOAD

$REQUIRED_PAGE_ACCESS = 3;  #-- Must be a project administrator to view this
                             #-- page. This is enforced by InitUserPrefs()

## Import Text Strings - this has to come after we require MRlib.pl
&importLanguageFile('MREditUser_Page.txt');
## End Import Text Strings #####

#####
#### LOCAL SUBROUTINES ####
#####

#####
#### END SUBROUTINES SECTION ####
#####

#####
#### MAIN #### CODE #### BEGIN #### HERE ####
#####

&ReadParse();              #-- Reads query string and POST data, puts the
                             #-- results into %in

$username=$in{'USER'};
$ProjectID = $in{'PROJECTID'};
$userkey = $in{'MRP'};
```

```

$errorcode=&InitMRUserEnvironment($username);
$errorcode = &SetCMMASTER($ProjectID);
&InitUserPrefs($username,$ProjectID);
&InitRealUserName($ProjectID);
&InitUserEmail($ProjectID);
$exitcode = &InitProjPrefs($ProjectID);
$ProjectName = $ENV{'PROJECT_NAME'};
@UserFieldList = &GetUserFields($ProjectID);
$CMMASTER = $ENV{'CMMASTER'};
&CheckLogin();

# get list of project users from passwd file,
# put in an option string for <select>

# Initialize.
&ReadPasswordHash();

foreach my $key (@PasswordInfoOrderedKeys)
{
    $userid = &ReadPasswordHash($key, 0);
    $N = $RealName{$userid};

    next if !&CheckUser($userid, $ProjectID);
    #rob- changed from ne "yes" just in case we someday
    #change the return value
    next if !&IsInternalUser($userid, $ProjectID);
    $optionstring .= "\n <OPTION VALUE=$userid> $N";
    $optionstring .= " \"$userid\"";
}

&StartIt();    #-- outputs the start of the web page

&StartDialog([${STR::MREditUser_Page_PageHeader_0}]);

print &MakeHelpButton(&RH_ShowHelpByString("Edit_Agents"));
if ($in{'ERROR'} ne ""){
    print "\n <BR><B> ${STR::MREditUser_Page_1}
        ${STR::MREditUser_Page_2}
        ${STR::MREditUser_Page_3} </B><P>";
}

print " $headfontOn ${STR::MREditUser_Page_4} <P>

    ${STR::MREditUser_Page_5}
    <UL>
    <LI> ${STR::MREditUser_Page_6}
    <LI> ${STR::MREditUser_Page_7}
    ${STR::MREditUser_Page_8} $headfontOff
    </UL>";

#-- Form to gather data for the CGI that actually adds/changes the user:

print " <FORM METHOD=POST NAME=DELUSERS ACTION=/MRcgi/MReddelUsers.pl>";

@NameList = "";
@ProjList = "";
$counter1 = 0;
$counter2 = 0;

```

```

print "\n <TABLE><TR><TH>$fontOn  ${STR::MREditUser_Page_9} $fontOff </TH>
      <TD>$fontOn  <SELECT NAME=UID>
      $optionstring
      </SELECT>$fontOff </TD></TABLE>";

print  " <INPUT TYPE=CHECKBOX NAME=DELETE>
      ${STR::MREditUser_Page_10}  ${STR::MREditUser_Page_11}
<B>${STR::MREditUser_Page_12} </B>";

print "<BR>";

#-- Most FootPrints forms contain hidden fields to pass necessary data
#-- to the target CGI (which otherwise would have no way of getting it)

print "\n <INPUT TYPE=HIDDEN NAME=PROJECTID VALUE=$ProjectID>";
print "\n <INPUT TYPE=HIDDEN NAME=USER VALUE=$username>";
print "\n <INPUT TYPE=HIDDEN NAME=MRP VALUE=$userkey>";

print "<BR>$headfontOn ${STR::MREditUser_Page_16} $headfontOff <BR><BR>";
print &MakeButton("go-but2.gif", "javascript: document.DELUSERS.submit()", $STR::MRlib_Go,
$STR::MRlib_Go, , , "ALIGN=MIDDLE");
print "\n </FORM>";

&EndDialog();

&EndIt();          #-- finishes up the web page
exit($exitcode);

#####
##### MAIN ##### CODE ##### END ##### HERE #####
#####

```

Example 3. External extension to FootPrints that emulates a web server.

```
#!/usr/bin/perl -w
#
# This is an example of how to extend FootPrints functionality
# by emulating a web server.
#
# This script essentially creates a very basic FootPrints project
# and links it to a new LDAP address book. It's a paired-down
# version of a script used at the University of Michigan. The
# following functionality has been REMOVED in order to make the
# example simpler:
#   - Gathering information about the new project via a form
#   - Validate the information about the new project
#   - Check the user's authorization
#   - Create project fields
#   - Adding the user creating the project to the project as
#     a project administrator
#   - Set up incoming/outgoing email for the project
#   - Log information about what's being done
#   - Generate a web page to show the user creating the project
#     the status and results
#
# WARNING: this script is not runnable the way it is and it has not
# been tested. This script exists only to provide a simplified
# example of how to write an external extension to FootPrints by
# emulating a web server. If you are interested in a working
# version of this script, please contact markmont@umich.edu
#

use warnings;
use URI::Escape;

$FOOTPRINTS_HOME      = '/usr/local/footprints';
$MRPROJECTS           = "$FOOTPRINTS_HOME/etc/MRprojects";

$LDAPSERVER           = 'ldap.itd.umich.edu';

# Set this to be the username of a FootPrints system administrator:
# (This is the user that will create the new project on behalf of
# the user requesting the new project)
$FOOTPRINTS_SYSADMIN = 'sysadminrobot';

sub
create_userkey
# Based on code from footprints/cgi/MRhomepage.pl
# This is necessary for making FootPrints think that
# the user actually logged in via MRentrancePage.pl,
# MRlogin.pl, and MRhomepage.pl when in fact all we
# did was call bin/mrLogin.
{

    srand (time|$$);

    # first decide on the length of the MRP
    my $length = int(rand(4) + 7); # 7 - 11 characters long
```

```

# these are the possible characters
my @chars = ('0'..'9', 'a'..'z', 'A'..'Z');

# make an array of some random characters
my @key = map { $chars[int(rand($#chars))]} (1 .. $length);

$key[0] = '0'; # Must start with '0' for agents

# encode the length of the key into the key: use the character whose distance from "A"
is equal to the length of the string
$key[2] = chr(65 + $length);

return join( '', @key );
}

```

```

sub query_string_param
# Creates a part of a query string
{
    my( $var, $val ) = @_;
    return uri_escape( $var ) . '=' . uri_escape( $val ) . '&';
}

```

```

sub
check_output
#
# Check to see if the program ran correctly:
# - There must be an META tag to refresh to a new URL
# - The phrases "error" and "stack trace" must not appear in the output.
#
{
    my( $output_ref, $program ) = @_;

    print "\nOutput from $program:\n\n";
    my( $output_error ) = 0;
    my( $output_meta ) = 0;
    foreach my $output_line (@{$output_ref})
    {
        $output_error = 1
            if $output_line =~ /\berror\b/i || $output_line =~ /stack\s*trace/i;
        $output_meta = 1 if $output_line =~ /<META\s+HTTP-EQUIV="REFRESH"/i;
        print $output_line;
    }
    die( "An error occurred when running $program" )
        if ( $output_error || !$output_meta );
}

```

```

#####
#
# MAIN PROGRAM:
#

```

```

# The name of the project to create. Presumably in real life this would
# come from an HTML form submitted by the user requesting the new project,

```

```

# and we'd have checked it to be sure that it's not too long, doesn't
# contain illegal characters, and that it's not the same as the name of
# a project that already exists.
$projectname = 'A test project';

# Some FootPrints programs and code expect us to be in this directory:
chdir( "$FOOTPRINTS_HOME/cgi" );

($tm_sec, $tm_min, $tm_hour, $tm_mday, $tm_mon, $tm_year, $tm_wday,
 $tm_yday, $tm_isdst) = localtime();
$admin_userkey = create_userkey();

#
# Create the query string to pass into cgi/MRnew_proj.pl
# This is easier than dealing with POST data (which is how this
# information would usually get there), and FootPrints doesn't care.
#
# Note that we're saying that we're in project 1 (you have to create
# a new project from some other, already-existing, project).
#
# We don't give the user a choice about address book configuration,
# that's all hard-coded.
#

$qqs = '';
$qqs .= query_string_param( 'AB', 1 );
$qqs .= query_string_param( 'ABMASTER', '' );
$qqs .= query_string_param( 'CONVERT', '' );
$qqs .= query_string_param( 'FROMWHERE', '' );
$qqs .= query_string_param( 'LDAPBDN', 'ou=People,dc=umich,dc=edu' );
$qqs .= query_string_param( 'LDAPNAME', $LDAPSERVER );
$qqs .= query_string_param( 'LDAPPORT', 389 );
$qqs .= query_string_param( 'LDAPTEST', 'Pachla' );
$qqs .= query_string_param( 'MRP', $admin_userkey );
$qqs .= query_string_param( 'NEWDN', '' );
$qqs .= query_string_param( 'NEWLDAPNAME', 'UofM Online Directory' );
$qqs .= query_string_param( 'NEWPASSWD', '' );
$qqs .= query_string_param( 'NEWPROJNAME', $projectname );
$qqs .= query_string_param( 'PROJECTID', 1 );
$qqs .= query_string_param( 'SAVE_CHANGES', 1 );
$qqs .= query_string_param( 'USER', $FOOTPRINTS_SYSADMIN );

#
# Set up the environment to make FootPrints think that it's being
# invoked by a web server. Note that the query string created above
# is placed into the environment.
#

$ENV{'CONTENT_LENGTH'} = "0";
$ENV{'CONTENT_TYPE'} = "application/x-www-form-urlencoded";
$ENV{'QUERY_STRING'} = $qqs;
$ENV{'REMOTE_USER'} = $FOOTPRINTS_SYSADMIN;
$ENV{'REQUEST_METHOD'} = "GET";

#
# Set some additional environment variables to make FootPrints happy.
# These are necessary for bin/mrLogin and bin/mrLogout below. We could

```

```

# log in by faking web requests to the FootPrints CGIs (MRlogin.pl,
# MRhomepage.pl), but it's easier to do it this way.
#

$ENV{'PATH'} = "$FOOTPRINTS_HOME/bin:./opt/bin:/sbin:/usr/bin:/usr/sbin";
$ENV{'CMI'} = "$FOOTPRINTS_HOME/";
$ENV{'CMMASTER'} = "$FOOTPRINTS_HOME/db/MASTER1";
$ENV{'PROJECT_NAME'} = "LSA FootPrints Test";
$ENV{'MRweb'} = "1";
$ENV{'mrp'} = $admin_userkey;
$ENV{'mrt'} = sprintf( "%04d-%02d-%02d %02d:%02d:%02d", $tm_year + 1900,
    $tm_mon + 1, $tm_mday, $tm_hour, $tm_min, $tm_sec );
$ENV{'mru'} = $FOOTPRINTS_SYSADMIN;

#
# Log into FootPrints
#

system("$FOOTPRINTS_HOME/bin/mrLogin 1");

#
# Actually create the new project using the query string we created
# above:
#

my @output = ` $FOOTPRINTS_HOME/cgi/MRnew_proj.pl 2>&1 `;
check_output( \@output, "$FOOTPRINTS_HOME/cgi/MRnew_proj.pl" );

#
# Find the project ID of the new project:
#

$ProjectID = 0;
open( PROJFILE, "<${MRPROJECTS}" )
    or die( "Cannot open $MRPROJECTS: $" );
while ( $line = <PROJFILE> )
{
    chomp( $line );
    @fields = split( /:/, $line );
    if ( $fields[2] eq $projectname )
    {
        $ProjectID = $fields[0];
        last;
    }
}
close( PROJFILE );
die( "New project not found in $MRPROJECTS" ) unless $ProjectID;

#
# Switch to the new project:
#

system("$FOOTPRINTS_HOME/bin/mrLogout");
$ENV{'CMMASTER'} = "$FOOTPRINTS_HOME/db/MASTER${ProjectID}";
system( "$FOOTPRINTS_HOME/bin/mrLogin $ProjectID" );

```

```

#
# Set up LDAP field mappings for the address book:
# Again, we don't give the user a choice about address book configuration.
#

$qs = '';
$qs .= query_string_param( 'DISPLAYED_NAME', 'Email Address' );
$qs .= query_string_param( 'FIELDS', 'Uniqname:uid:char::DELIM::Full
Name:displayname:char::DELIM::Department:ou:char::DELIM::Room/Address:postaladdress:multiC::DELIM::T
Number:telephonenumber:char::DELIM::Email Address:mail:mail::DELIM::' );
$qs .= query_string_param( 'FIELD_TYPE', 'mail' );
$qs .= query_string_param( 'FP_FIELDS', 'uid:char' );
$qs .= query_string_param( 'FROMWHERE', '' );
$qs .= query_string_param( 'LDAP_ATTRIBS', 'Email Address:mail' );
$qs .= query_string_param( 'MRP', $admin_userkey );
$qs .= query_string_param( 'PRIMARYKEY', 'Uniqname' );
$qs .= query_string_param( 'PROJECTID', $ProjectID );
$qs .= query_string_param( 'SAVE_CHANGES', 1 );
$qs .= query_string_param( 'USER', $FOOTPRINTS_SYSADMIN );

$ENV{'QUERY_STRING'} = $qs;

@output = ` $FOOTPRINTS_HOME/cgi/MRLDAPFieldSetup.pl 2>&1`;
check_output( \@output, "$FOOTPRINTS_HOME/cgi/MRLDAPFieldSetup.pl" );

#
# Clean up and log out:
#

system( "$FOOTPRINTS_HOME/bin/mrLogout" );

# end of script

```

Example 4. External extension to FootPrints that re-uses existing FootPrints subroutines.

```
#!/usr/footprints_perl/bin/perl
#
# Example showing how to re-use FootPrints subroutines in your own code
# to create external extensions to FootPrints.
#
# This script was created just for this presentation by removing most
# of the functionality from the new-project CGI that the University
# of Michigan uses (new-project allows people who are not FootPrints
# system administrators to create new FootPrints projects on their own).
# If you'd like the real new-project CGI, please contact markmont@umich.edu.
#

BEGIN
{
  # Tell Perl where to find MRlib.pl:
  push( @INC, '/usr/local/footprints/cgi' );
}

# MRlib.pl defines AUTOLOAD, which will automatically find
# and load all of the other subroutines we'll re-use from FootPrints

package FP;
require "MRlib.pl";

# We can now use PrintStyle(), TableTop(), StartDialog(), and
# whatever other FootPrints subroutines we need, saving us a
# lot of work for stuff we'd otherwise need to implement ourselves.

use CGI;

$FOOTPRINTS_HOME = '/usr/local/footprints';
$SYSADMIN_EMAIL = 'lsa-sst-unix@umich.edu';

$q = new CGI;

print
  $q->header( 'text/html' ),
  $q->start_html( -title => 'FootPrints - Create a New Project',
    -bgcolor => "#fafaf2" ),
  );

PrintStyle();

print
  $q->font(
    { -face => 'Verdana, Arial, Helvetica, sans-serif',
      -color => "#003399" },
  $q->br(),
  $q->h1( 'Create a new FootPrints project' )
  );
```

```

$servername = $ENV{'SERVER_NAME'};
$username   = $ENV{'REMOTE_USER'};
$host      = hostname();

#
# Default values for form fields:
#

$readdocs      = 2;
$joinedmailgroup = 2;
$name          = '';
$description   = '';
$public        = 1;
$login_url     = '';
$logout_url    = 'http://www.lsa.umich.edu/lsait/';
$email_local   = '';
$email_public  = '';
$email_admin   = "$username@umich.edu";
$spam_action   = 1;
$spam_score    = 90;
$email_spam    = "$username@umich.edu";

TableTop( round => 1, dowhat => 'print' );
print <<__END_OF_HTML__
  <p>
    Use this form to request new FootPrints projects.
    All of the fields below are required.
    The information you enter below will be used to create
    the project and will also be sent via email to the
    LSAIT FootPrints project team.
  </p>
__END_OF_HTML__
;
TableBottom( dowhat => 'print' ); print "<br />\n";

print "<form method=\"post\" name=\"newprojectinfo\" enctype=\"multipart/form-data\"
action=\"/cgi-bin/new-project\">\n";

if ( $readdocs == 1 )
{
  $readdocs_yes = 'checked';
  $readdocs_no  = '';
}
else
{
  $readdocs_yes = '';
  $readdocs_no  = 'checked';
}

if ( $joinedmailgroup == 1 )
{
  $joinedmailgroup_yes = 'checked';
  $joinedmailgroup_no  = '';
}
else
{
  $joinedmailgroup_yes = '';

```


an agent or project administrator for this project be added.)

\$joinedmailgroup_yes/>Yes

 \$joinedmailgroup_no/>No

It is also strongly recommended (but not required) that you join the
 [footprints-project-administrators](https://directory.umich.edu/ldapweb-bin/url?ldap:///cn=footprints-project-administrators,ou=Us)
 mailing list.
 [Click here to join](http://listserver.itd.umich.edu/cgi-bin/lyris.pl?join=footprints-project-administrators). This is a campus-wide list for all FootPrints project administrators at the University of Michigan to ask questions, share tips, make suggestions, and discuss FootPrints.

Project Name:

This project name will appear in every email and on every web page.

Choose a project name that will make sense both to the people who use the project as well as to others in the College. For example, "Maintenance requests" is probably a bad project name since it's not clear from just the project name what department(s) should use the project.

Description:
\$description

Enter a short description of what the project is for and who should use it. This description will appear in the list of available projects and should help users decide whether this project applies to them.

Project listing:

 \$list_for_everyone/>Everyone should see the project listing.

 \$list_for_agents/>Only agents should see the project listing.

This option controls who will see information about the project on the main page of the FootPrints web server. Normally, you will want to let everyone see your project information so that they will know that your project exists, who should use it, and for what purposes your project should be used. In order to keep projects that users would not be interested in (such as test projects) from cluttering up the listings, though, you also have the option of

for your project's incoming email.

Anyone will be able to submit new requests by sending them to the email address above. Advanced users can also send email containing FootPrints commands to this address; FootPrints will execute the commands and send a reply message containing the commands' output.

NOTE: It is *strongly recommended* that you *do not advertise* the address above. Instead, have this address as a member of one or more [UofM Online Directory](https://directory.umich.edu/) groups. This way, mail sent by users during FootPrints server maintenance periods will be queued for later delivery instead of being bounced, and you can make a change to the group instead of re-training all of your users if the project is ever moved to a different FootPrints server.

Published Project Email Address:

Enter the email address you will ask your users to send mail to when submitting requests to the FootPrints project. It is recommended that this be the email address for a [UofM Online Directory](https://directory.umich.edu/) group that contains the server project email address (above) as a member. (If you choose to use a UMOD group, you will need to create or modify it yourself after you have finished creating the project).

Last Resort Email Address:

The Last Resort Email Address is displayed in FootPrints error messages. For example, if there is a problem with a new request that a user is attempting to submit via email (they lack the necessary permissions, the request contains illegal content, or there is some other problem), FootPrints will ask the user to contact the address above for help. FootPrints will also use this address to notify you of any outgoing email that could not be delivered.

It is recommended that the Last Resort Email Address be a [UofM Online Directory](https://directory.umich.edu/) group containing all of the Agents and Project Administrators for this project. (Suggestion: you can have this be the same group as the one you added to the lsa-footprints-admins mail group above.)

```

<p>
<font color="red"><b>WARNING:</b></font>&nbsp;
<b>Email sent to the Last Resort Email Address must
<em>never</em>&nbsp; wind up at any FootPrints Project
Email Address.</b> Make sure that the address you
provide does not go to any group(s) which in turn
send email to a FootPrints project, otherwise an email
loop will be created.
</p>
<hr />
<p>
<b>Junk email handling:</b><br />
<table border="0">
  <tr>
    <td valign="top">
      <input type="radio" name="spam_action" value=1 $spam_passthrough/>
    </td>
    <td>Do not filter incoming email</td>
  </tr>
  <tr>
    <td valign="top">
      <input type="radio" name="spam_action" value=2 $spam_redirect/>
    </td>
    <td>
      Redirect any incoming email message for which Brightmail assigns
      a score of <input type="text" name="spam_score"
      value="$spam_score" size=3 maxlength=3 />
      or more to the following email address:<br/>
      <input type="text" name="email_spam" value="$email_spam"
      size=60 maxlength=127 />
    </td>
  </tr>
</table>
</p>
<p>
The "Key Required" project option will keep most junk email out of your
project. If you plan on leaving this project option enabled, then it
may not be necessary to enable junk email filtering too.
<i>Note that enabling junk email filtering carries the risk that some
legitimate email may be incorrectly identified as junk and redirected.</i>
</p>
<p>
<b>Note:</b>&nbsp;
Currently, Brightmail is not providing a numerical junk email score for
each message. Useful values for the score above are 90 (things Brightmail
identifies as "junk") and 76 (things Brightmail identifies as "suspected
junk").
</p>
<br />
__END_OF_HTML__
;
EndDialog(); print "<br />\n";

print "</form>\n";

TableTop( round => 1, dowhat => 'print' );
print "<p>Click the button below to send the project creation request.\n";
print "Please click the button <em>only once</em> and be patient as it may take several
minutes to create the new project.</p>\n";

```

```
print MakeButton("go-but2.gif", "javascript: document.newprojectinfo.submit()", "Go",  
"Go", "", "#003399", "");  
TableBottom( dowhat => 'print' ); print "<br />\n";  
  
print $q->end_html;  
exit( 0 );  
  
# end of script
```

Example 5. External extension to FootPrints that modifies a FootPrints configuration file directly

```
#!/usr/bin/perl
#
# Example showing how to modify files in $CMI/etc or $CMI/db directly
# from an external extension to FootPrints.
#
# This example looks for a project with a particular name in
# $CMI/etc/MRprojects. When it finds it, it notes the project number
# for possible later use, it sets the project administrator email
# address, and it turns on the project menu for users in the project.
#

$FOOTPRINTS_HOME    = '/usr/local/footprints';
$MRprojects         = "$FOOTPRINTS_HOME/etc/MRprojects";

# Which project to look for and what to change it's admin email address
# to. These values are hardcoded for the sake of this example, but
# would presumably be obtained through an HTML form, checked to be sure
# they are not too long or contain illegal characters, etc.
$projectname        = "A test project";
$project_admin_email = "test-project@umich.edu" #

$ProjectID = 0;

# Original MRproject file:
open( PROJFILE, "<${MRprojects}" )
  or die( "Could not open $MRprojects: $!" );

# New MRprojects file that we're creating:
open( NEWPROJFILE, ">${MRprojects}.new" )
  or die( "Could not create ${MRprojects}.new: $!" );

#
# Copy each line of the original file to the new file.
# If we find the line for the project we're modifying,
# make the necessary changes to that line before writing
# it out.
#
while ( $line = <PROJFILE> )
{
    chomp( $line );
    my( @fields ) = split( /:/, $line );

    # Is this the project we are looking for?
    if ( $fields[2] eq $projectname )
    {
        # Remember the project ID for later (shows that we found the project
        # we were looking for and made the change).
        $ProjectID = $fields[0];

        # Set the project administrator email address to the new value
        $fields[6] = $project_admin_email;
    }
}
```

```

# Turn on the project menu for the project
$fields[51] = 1;

# For loop to prevent a "use of uninitialized value" warning in the join
for my $i (0..50) { $fields[$i] = '' if ! defined( $fields[$i] ); }
$line = join( ':', @fields );

}

# Write the possible modified line to the new file
print NEWPROJFILE "$line\n";

}

close( NEWPROJFILE );
close( PROJFILE );

#
# If we didn't find the project, clean up and exit without changing
# the original MRprojects file.
#
if ( ! $ProjectID )
{
    unlink( "${MRprojects}.new" );
    die( "New project not found in $MRprojects" );
}

#
# We found the project and made the change, replace the original MRprojects
# file with the new one we created.
#

rename( "${MRprojects}", "${MRprojects}.bak" )
    or die( "Could not rename old ${MRprojects}: $!" );

if ( ! rename( "${MRprojects}.new", "${MRprojects}" ) )
{
    $error = $!;
    # Put the original file back:
    rename( "${MRprojects}.bak", "${MRprojects}" );
    die( "Could not rename new ${MRprojects}: $error" );
}

# end of script

```

Example 6. Adding page breaks to wrapped reports: User interface changes

```
diff -urdN
footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/PrintFieldLayoutJavaScript.pl
footprints/cgi/SUBS/MRsearch_page/PrintFieldLayoutJavaScript.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/PrintFieldLayoutJavaScript.pl
2005-07-15 11:26:10.000000000 -0400
+++ footprints/cgi/SUBS/MRsearch_page/PrintFieldLayoutJavaScript.pl      2005-09-19
21:46:50.505585000 -0400
@@ -134,6 +134,12 @@
     obj.options[obj.options.length] = new
Option("\$STR::MRsearch_page_footprintsFieldLineBreak\");
    }

+   function addPageBreak()
+   {
+       obj = document.searchpage.choiceArea;
+       obj.options[obj.options.length] = new Option("Page Break");
+   }
+
    function resetForm()
    {
        initializeDisplayLists();
diff -urdN
footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/PrintFieldLayoutSection.pl
footprints/cgi/SUBS/MRsearch_page/PrintFieldLayoutSection.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/PrintFieldLayoutSection.pl
2005-07-15 11:26:10.000000000 -0400
+++ footprints/cgi/SUBS/MRsearch_page/PrintFieldLayoutSection.pl      2005-09-19
21:49:01.736367000 -0400
@@ -102,7 +102,7 @@

    print "$fieldPicker_addButton <BR>";

-   # "add line break" button for wrapped style reports only
+   # "add line/page break" buttons for wrapped style reports only
    if ($FP::in{'RETURN'} eq 'Wrap')
    {
        my $fieldPicker_linebreakButton = &FP::MakeButton("linebreak.gif", "javascript:
addLineBreak(); ", "\$STR::MRsearch_page_addLineBreak", "", "", "",
"TABINDEX=$FP::tabCount");
@@ -110,6 +110,12 @@
        print "$fieldPicker_linebreakButton<BR>";

        $FP::tabCount++;

+       print &FP::MakeButton("news_icon.gif", "javascript: addPageBreak(); ",
+           "Page Break", "", "", "", "TABINDEX=$FP::tabCount") . " <BR>\n";
+
        $FP::tabCount++;
+
    }

    print "$fieldPicker_upArrowButton";
```

Example 7. Adding page breaks to wrapped reports: Getting the front-end script to pass page break information to the back-end script

```
diff -urdN footprints-005-report-page-breaks/cgi/SUBS/InitFieldsAllowed.pl
footprints/cgi/SUBS/InitFieldsAllowed.pl
--- footprints-005-report-page-breaks/cgi/SUBS/InitFieldsAllowed.pl      2005-06-29
15:03:35.000000000 -0400
+++ footprints/cgi/SUBS/InitFieldsAllowed.pl      2005-09-19 21:28:34.158793000 -0400
@@ -37,6 +37,7 @@
     # FP Fields

     %FieldsAllowed = ( 'Line Break' => 1,
+
+         'Page Break' => 1,
+         'mr' => 1,
+         'status' => 1,
+         'entry date' => 1,

diff -urdN
footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/FieldsDisplayedSection.pl
footprints/cgi/SUBS/MRsearch_page/FieldsDisplayedSection.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/FieldsDisplayedSection.pl
2005-08-16 11:16:55.000000000 -0400
+++ footprints/cgi/SUBS/MRsearch_page/FieldsDisplayedSection.pl 2005-09-19
21:46:01.545286000 -0400
@@ -134,7 +134,8 @@
     if ($val =~ /^USETHISFIELD(\d*)USETHISINDEX(.*?)$/)
     {
         $L++;
-         my $field = ($2 eq 'Line Break') ? "Line Break$L" : $2;
+         my $field = ($2 eq 'Line Break' || $2 eq 'Page Break') ?
+         "$2$L" : $2;
+         $order{$field} = $1;
     }
@@ -280,8 +281,10 @@
     {
         my $field1 = $field;
         $field = 'Line Break' if $field =~ /Line Break/;
+         $field = 'Page Break' if $field =~ /Page Break/;

-         next if ($field ne 'Line Break' && !$FP::FieldsAllowed{$field1});
+         next if ($field ne 'Line Break' && $field ne 'Page Break' &&
+         !$FP::FieldsAllowed{$field1});
         next if !$order{$field1};

         # Don't allow dateTime link fields to be included
@@ -294,6 +297,7 @@
        $printfield ||= $GoodABFields{$field};
        $printfield ||= &FP::UnFix_Fields($field);
        $printfield = $STR::MRsearch_page_footprintsFieldLineBreak if $field eq 'Line
Break';
+        $printfield = 'Page Break' if $field eq 'Page Break';

        push @usefieldlist, $printfield;
        $usefieldhash{$field} = 1;

diff -urdN
footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/TranslateFieldsChosen.pl
footprints/cgi/SUBS/MRsearch_page/TranslateFieldsChosen.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRsearch_page/TranslateFieldsChosen.pl
2005-07-15 11:26:11.000000000 -0400
```

```
+++ footprints/cgi/SUBS/MRsearch_page/TranslateFieldsChosen.pl 2005-09-19
21:51:23.947222000 -0400
@@ -41,6 +41,7 @@
```

```
    my %currentLanguageFpFields = (
        $STR::MRsearch_page_footprintsFieldLineBreak =>
'INTERNAL_LINE_BREAK',
+        'Page Break' => 'INTERNAL_PAGE_BREAK',
        $STR::MRsearch_page_footprintsFieldNumber =>
'INTERNAL_MR',
        $STR::MRsearch_page_footprintsFieldPriority =>
'INTERNAL_PRIORITY',
        $STR::MRsearch_page_footprintsFieldDateSubmitted =>
'INTERNAL_SUBMIT_DATE',
@@ -67,6 +68,7 @@
```

```
    my %internalFpFields = (
        'INTERNAL_LINE_BREAK' => 'Line Break',
+        'INTERNAL_PAGE_BREAK' => 'Page Break',
        'INTERNAL_MR' => 'mr',
        'INTERNAL_PRIORITY' => 'priority',
        'INTERNAL_SUBMIT_DATE' => 'entry date',
```

Example 8. Adding page breaks to wrapped reports: Adding page break functionality to the back-end script

```
diff -urdN footprints-005-report-page-breaks/cgi/SUBS/MRdirectSearch/FormatIt.pl
footprints/cgi/SUBS/MRdirectSearch/FormatIt.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRdirectSearch/FormatIt.pl      2005-08-19
11:17:25.000000000 -0400
+++ footprints/cgi/SUBS/MRdirectSearch/FormatIt.pl      2005-09-19 21:36:55.532014000
-0400
@@ -202,6 +202,7 @@

    my @splitRows = ();
    my %lineseps = ();
+   my %pageseps = ();
    my $numCols = scalar keys %printTheseFields;

    if ( $FP::in =~ /USEFIELD/ && $FP::in !~ /USETHISFIELD/ )
@@ -231,10 +232,11 @@
        my ($key, $val) = split(/=/,$in_copy[$i],2);
        if ( $key eq 'choiceArea' )
        {
-           if ( $val =~ /^USETHISFIELD(\d*)USETHISINDEXLine Break$/ )
+           if ( $val =~ /^USETHISFIELD(\d*)USETHISINDEX(Line|Page) Break$/ )
            {
                $j++;
-               $lineseps{$1-$j} = $1;
+               $lineseps{$1-$j} = $1; # page break implies line break, too
+               $pageseps{$1-$j} = 1 if $2 eq 'Page';
            }
            elsif ( $val =~ /^USETHISFIELD(\d*)USETHISINDEX(.*?)$/ )
            {
@@ -244,11 +246,15 @@
        }
    }
    my $last = 0;
+   my $rowNumber = 0;
+   my %newPageAfterRow = ();
    for my $key (sort {$a <=> $b} keys %lineseps)
    {
        last if ($key > $numCols);
        push @splitRows, ($key-$last);
+       $newPageAfterRow{$rowNumber} = 1 if $pageseps{$key};
        $last = $key;
+       $rowNumber++;
    }
    if ($numCols > $last)
    {
@@ -399,6 +405,7 @@
        print $sortHeaderRow if ($sortHeaderRow && !$inMsglSection &&
!$msglHeaderPrinted);

        my $multipleRows = 1 if (scalar @rowSplits > 1);
+       my $rowNumber = 0;
        while(@rowSplits)
        {
            my $subRowSize = shift @rowSplits;
@@ -427,7 +434,12 @@
        }
    }
}
```

```
print "</TR></TABLE></TD></TR></TABLE>";
+
+   print "<br style=\"page-break-before: always\">\n"
+     if $newPageAfterRow{$rowNumber};
+
+   $cellNumber = $lastCell + 1;
+   $rowNumber++;
  }
}
```

Example 9. Adding page breaks to wrapped reports: changes to the back-end script for it to get the page-break information from the front end script

```
diff -urdN
footprints-005-report-page-breaks/cgi/SUBS/MRdirectSearch/TranslateFieldsChosen.pl
footprints/cgi/SUBS/MRdirectSearch/TranslateFieldsChosen.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRdirectSearch/TranslateFieldsChosen.pl
2005-06-29 15:03:50.000000000 -0400
+++ footprints/cgi/SUBS/MRdirectSearch/TranslateFieldsChosen.pl 2005-09-19
21:39:14.342845000 -0400
@@ -40,6 +40,7 @@

    %currentLanguageFpFields = (
                                $STR::MRdirectSearch_footprintsFieldLineBreak =>
'INTERNAL_LINE_BREAK',
+                                'Page Break' => 'INTERNAL_PAGE_BREAK',
                                $STR::MRdirectSearch_footprintsFieldNumber =>
'INTERNAL_MR',
                                $STR::MRdirectSearch_footprintsFieldPriority =>
'INTERNAL_PRIORITY',
                                $STR::MRdirectSearch_footprintsFieldDateSubmitted =>
'INTERNAL_SUBMIT_DATE',
@@ -69,6 +70,7 @@

    %internalFpFields = (
                                'INTERNAL_LINE_BREAK' => 'Line Break',
+                                'INTERNAL_PAGE_BREAK' => 'Page Break',
                                'INTERNAL_MR' => 'mr',
                                'INTERNAL_PRIORITY' => 'priority',
                                'INTERNAL_SUBMIT_DATE' => 'entry date',
diff -urdN footprints-005-report-page-breaks/cgi/SUBS/MRdirectSearch/getSearchData.pl
footprints/cgi/SUBS/MRdirectSearch/getSearchData.pl
--- footprints-005-report-page-breaks/cgi/SUBS/MRdirectSearch/getSearchData.pl 2005-08-22
15:34:21.000000000 -0400
+++ footprints/cgi/SUBS/MRdirectSearch/getSearchData.pl 2005-09-19 21:41:12.323563000
-0400
@@ -55,6 +55,7 @@
{
    # version 4+ format
    %twoWordFields = ('Line Break' => 1,
+                    'Page Break' => 1,
                    'entry date' => 1,
                    'entry time' => 1,
                    'Project Name' => 1
@@ -62,7 +63,7 @@

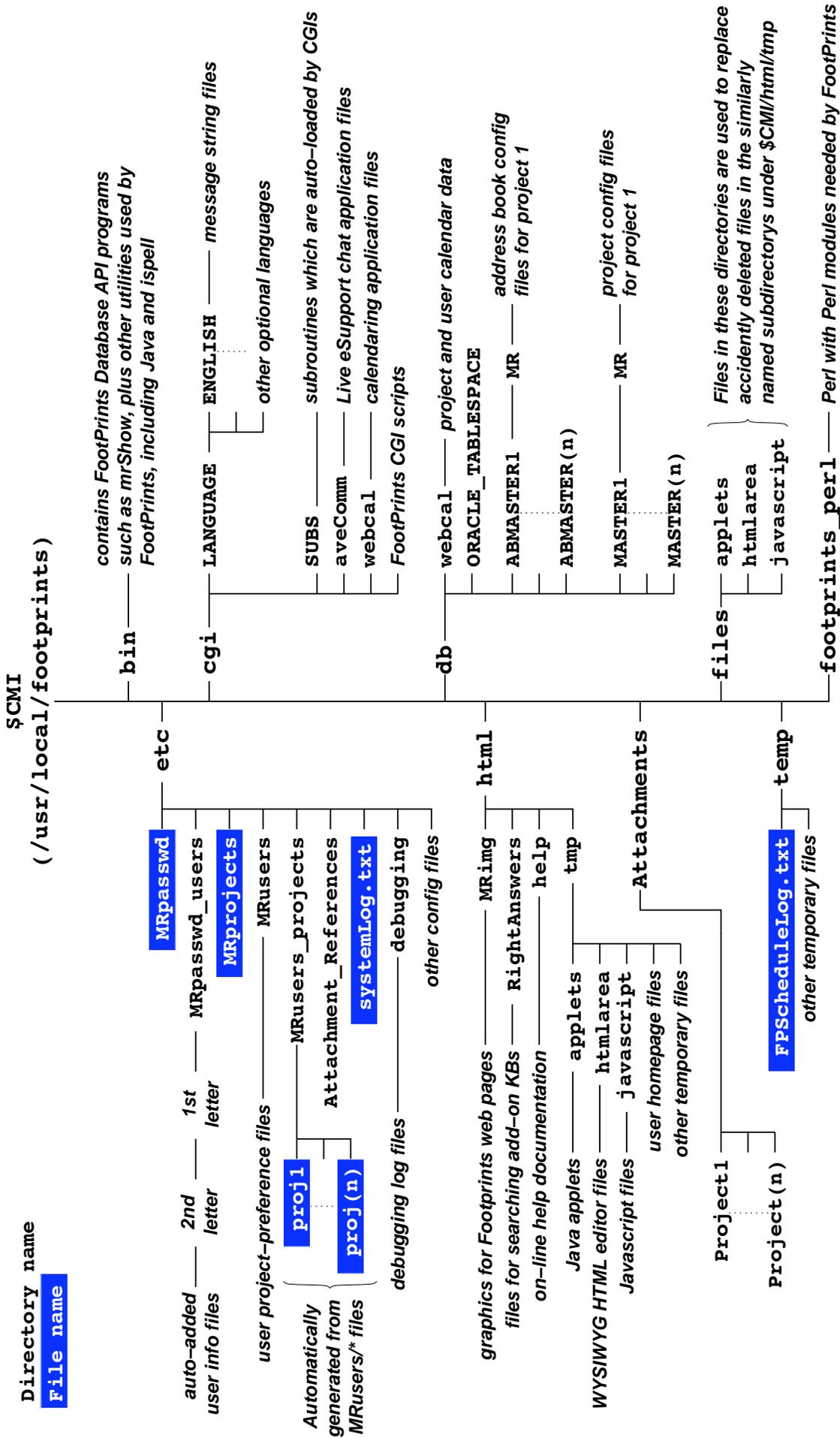
    foreach my $val (&FP::GetMultiples("choiceArea"))
    {
-        if ( $val =~ /^USETHISFIELD(\d*)USETHISINDEX(.*?)$/ && $val !~ /Line Break$/
+        if ( $val =~ /^USETHISFIELD(\d*)USETHISINDEX(.*?)$/ && $val !~ /Line Break$/
&& $val !~ /Page Break$/ )
        {
            # bugfix - use temp vars instead of $1, $2, because AUTOLOAD
            # changes $2 - JOE 6/5/01
```


**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Figures

**REPLACE THIS
SHEET WITH A
TAB THAT HAS
THE FOLLOWING
LABEL:**

Figures

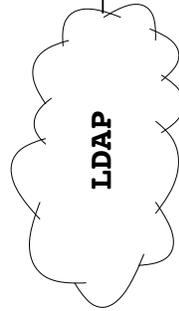


Symbols:

- exactly 1
- 0 or more
- | 0 or 1
- |/ 1 or more

ABMASTERY	
abID	primary key
abSUBMITTER	
abASSIGNEE	
abSUBMITDATE	
abUPDATEDATE	
abSTATUS	
custom AB field 1	
...	
custom AB field (n)	

The ABMASTERY table is only used if an address book is an internal FootPrints address book and is not dynamically linked to LDAP or another SQL database.



MASTERx	
mrID	primary key
mrREF_TO_MR	
mrREF_TO_MRX	
mrREF_TO_AB	
mrTITLE	
mrPRIORITY	key
mrSTATUS	key
mrDESCRIPTION	key
mrALLDSCRIPTIONS	key
mrASSIGNEES	
mrATTACHMENTS	
mrUPDATEDATE	
mrSUBMITTER	
mrPOPULARITY	
mrURGENT	key
mrESCALATEDBY	
mrUNASSIGNED	
custom project field 1	
...	
custom project field (n)	

MASTERx_HISTORY	
mrID	key
mrGENERATION	
mrHISTORY	

MASTERx_DESCRIPTIONS	
mrID	key
mrGENERATION	
mrDESCRIPTION	

MASTERx_TIMETRACKING	
mrID	key
mrGENERATION	
mrTIMESPENT	
mrRATE	
mrTIMEDATE	
mrTIMEUSER	
mrRATEDESC	

MASTERx_ABDATA	
mrID	primary key
custom AB field 1	
...	
custom AB field (n)	

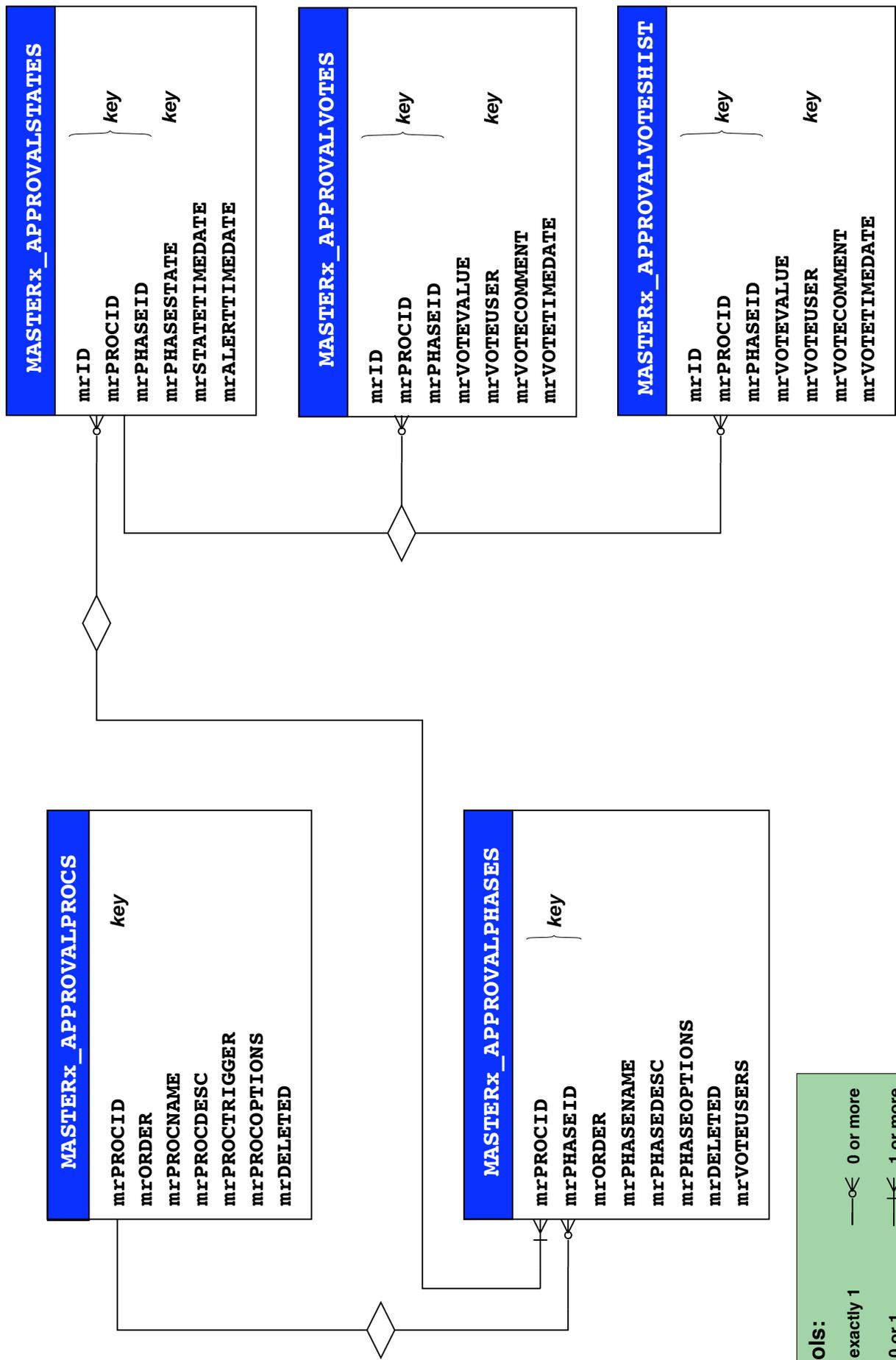
The mrGENERATION field in each table is independent of the mrGENERATION field in other tables. For a given mrID, MASTERx_HISTORY might have five records with mrGENERATION numbers of 0 through 4, while for the same mrID MASTERx_DESCRIPTIONS might have only one record with an mrGENERATION number of 1. The mrGENERATION number merely differentiates multiple records with the same mrID within a single table.

ABMASTERx_MASTER
<i>custom AB field 1</i>
⋮
<i>custom AB field (n)</i>

Master Contact Record table

MASTERx_TEMP
mrID
mrUSER
mrPID

Temporary table used for JOINS



Symbols:

- |○ exactly 1
- |○ 0 or more
- |○ 0 or 1
- |< 1 or more

