DiGIR: **Di**stributed **G**eneric **I**nformation **R**etrieval

Manual Installation Guide for Apache and IIS

Table of Contents

Introduction

Distributed **G**eneric **I**nformation **R**etrieval (DiGIR) is a client/server protocol for retrieving information from distributed resources. It uses HTTP as the transport mechanism and XML for encoding messages sent between client and server. It is an open source project hosted on Source Forge (http://digir.sourceforge.com. DiGIR was originally conceived to be the replacement for the Z39.50 protocol used in the Species Analyst project, but is intended to work with any type of information, not just Natural History collections.

Installation

This section provides detailed information on installing a DiGIR Provider service. The steps in the process are:

1. Identify the machine on which the service will run
2. Ensure PHP is functioning correctly on your web server
3. Download and install the DiGIR provider code
4. Copy DIGIR_ROOT/www/localconfig_dist.php to DIGIR_ROOT/www/localconfig.php and edit to reflect your local installation choices
5. Use the DiGIR configurator to complete the process and add at least one resource: http://localhost/digir/admin/setup.php

After completing the process, the following URL's will be available:

Default page:	http://localhost/digir/admin/index.php
DiGIR Provider service:	http://localhost/digir/DiGIR.php
DiGIR Administration Interface:	http://localhost/digir/admin/setup.php

Requirements

This document assumes installation on a functional web server with an operational PHP environment. Detailed instructions for installing PHP are available at the PHP website. The following are required to operate a DiGIR Data Provider service:

Web Server

This can be any web server that supports PHP. Apache or Microsoft's IIS are both suitable candidates (my personal preference is Apache 2.x).

PHP

PHP is a cross platform web scripting language. Since the DiGIR data provider is written in PHP, you need a PHP interpreter for your system. DiGIR generally works best with the latest version of PHP. At the time of writing, the latest stable release version of PHP is "4.3.1" and DiGIR is known to work correctly with that version. Version 4.2.3 or later is required for all functionality of the DiGIR Data Provider.

Domain Name for Server

It is recommended that the machine running the DiGIR Provider service has a fully qualified domain name (FQDN). If you do not have a FQDN for your server then you can use one of the many Dynamic DNS services to register a name for your server regardless of whether you have a static or dynamic IP address. One free service that has proven to be particularly stable is DynDNS.

Note that if you only have a dynamically allocated IP address, you must use Dynamic DNS services to name your machine otherwise the location entered into the directory of DiGIR providers will become unusable once your IP address changes.

Setting Up the Provider Environment.

Is PHP Working?

Warning!! *Once this test has been completed, it is a good idea to delete the test.php file that you create since the information it provides can be quite helpful to someone that may not have the best interests of the web server in mind.*

If you are certain that PHP is working correctly then you can skip this step. In a location that is accessible by your web server (e.g. c:\inetpub\wwwroot or /var/www/html), create a short script called "test.php" using a text editor. The contents of the script should look like this:

```
<?php
phpinfo();
?>
```

Run the script by loading it with your web browser (e.g. http://localhost/test.php). You should see a page with lots of information about your PHP installation and the web server environment. Make sure that the version is at least 4.2.3. If the resulting page looks like the script you typed above, then there is a problem. This means that the web server is not correctly determining that PHP files should be processed with the PHP processing engine. Carefully review the steps you followed when setting up the PHP engine for your web server, and make sure that the script is operating correctly before proceeding with the installation of the DiGIR provider service.

Note

Improper configuration of php can open significant security holes in your system. You should carefully examine the information provided in http://www.php.net/manual/en/security.php (especially http://www.php.net/manual/en/security.cgi-bin.php if you are considering a cgi-bin installation).

If you are using a php enabled hosting service, or are running a server shared by multiple users, you should probably consider everything placed in your DiGIR configuration files to be visible to all users on the machine, as it takes a good deal of care to make configuration files accessible to the web server but not to other users on the system.

PHP Configuration Directives

The PHP interpreter engine uses a configuration file to set default operating parameters. This file, php.ini is a plain text file that is formatted like a windows "ini" file. There are a couple of adjustments, or rather settings to check in the php.ini file. On windows, php.ini is located in the $SystemRoot folder (c:\windows). On linux, it is located in /etc.

PHP File Paths

PHP understands forward (/) and backward (\) slashes to be file path separators. The only difference is that when the backslash is used in PHP code, it must be escaped by typing a double-backslash (e.g. c:\\digir\\phpprovider). To avoid confusion, it is best to get into the habit of just using a forward slash for path separators (e.g. c:/digir/phpprovider).

register_globals

This setting should be off. This setting does not alter the functionality of the DiGIR provider, but it does have a big impact on the security of your web server. See the discussion on php.net.

extension_dir

This entry indicates where php can find it's extensions. These are compiled libraries which are dynamically loaded by the PHP interpreter. For the purposes of DiGIR, it should be a full path to the folder that contains the extension php_mbstring.*. If you installed PHP to c:\php then the extension folder setting is most likely:

extension_dir = c:/php/extensions

cgi.force_redirect

If you are running the IIS web server, then this must be 0 or Off.

extension=php_mbstring

The default configuration setting has this library commented out. If you are running PHP as a CGI application, then it is safe to leave it this way. If you are running PHP as a module loaded by the web server, then you must uncomment this entry so that it is loaded by the PHP interpretor, otherwise it will not be available to the DiGIR provider (and will cause error messages in the responses).

extension=php_sockets

This library is required by the DiGIRm service. If DiGIRm is disabled then there is no need to enable this library which implements low level internet communication functions. The default configuration setting has this library commented out. If you are running PHP as a CGI application, then it is safe to leave it this way. If you are running PHP as a module loaded by the web server, then you must uncomment this entry so that it is loaded by the PHP interpreter, otherwise it will not be available to the DiGIR provider (and will cause error messages in the responses).

There are numerous other settings in php.ini. Please review the php manual if you are curious about how these affect the operation of your PHP installation.

DiGIR File and Folders

The following examples are for a Windows installation. Please adjust as necessary for a Linux installation.

The distribution of the DiGIR Provider service will unzip to the following hierarchy. For example, if you unzip the distribution in `C:\`, a folder `C:\DiGIRprov` wil be created along with all the subfolders described below. The term `DiGIR_ROOT` used later in this document will refer to the file system path to the DiGIR folder (on linux, this may be `/var/www/DiGIRprov` for example).

```
DiGIR/
   admin          Location of administrative tools
   cache          Temporary files will be written here
   config         Configuration files
   doc            Documentation
   lib/           Various PHP libraries
     adodb +        PHP ADODB database abstraction library
     pear +         PHP PEAR libraries
     xpath +        XPath interpreter library
   log            Log files written here
   www            The DiGIR Provider Services
```

The contents of these folders is described in more detail below. Also indicated is whether the folder should be browseable or writable. "Browseable" in this context means whether scripts or pages contained within that folder may be retrieved via the web server when the URL of a document or script is requested by a web browser. Even though it will probably not hurt, it is generally not advisable to permit generation of "directory indexes" (file lists) when a directory rather than file is requested.

Folder:	`admin`
Browseable:	Yes (controlled access)
Writable:	No
Description:	This folder is a place holder for administrative tools currently under development. It will need to be accessible from the web to

enable remote management of the DiGIR Provider service. **It's important to use a security mechanism for restricting access to this folder** (such as IP address restrictions or HTTP authentication).

Folder: `cache`

Browseable: No

Writable: Yes

Description: In order to improve performance, the DiGIR Provider service will write some information to disk. This folder should be writable by the process running the DiGIR Provider script (typically the user id of the web server). It should not be accessible by a web browser.

Folder: `config`

Browseable: No

Writable: Yes (if you want to use the DiGIR configurator)

Description: The config folder contains information about the configuration of the DiGIR Provider service. Some of the configuration files may contain sensitive information such as passwords for connecting to databases, and so this folder should be protected as necessary. The contents must be readable by the web server process, writable if you want to use the DiGIR configurator, but should not be browseable.

Folder: `doc`

Browseable: No (optional)

Writable: No

Description: Contains this document plus additional release notes and other documentation.

Folder: `lib`

Browseable: No

Writable: No

Description: Contains PHP libraries necessary for the operation on the DiGIR Provider service. The DiGIR Provider distribution should contain all the libraries required to operate, and will modify the `include_path` of your PHP installation during operation to override the settings in `php.ini`. Three sub-folders contain the PHP ADODB, a subset of PEAR, and XPath libraries. Each of these folders may have many sub-folders.

Folder: `log`

Browseable: No

Writable: Yes

Description: The DiGIR Provider service will record every transaction by appending information to a log file contained in this folder.

Folder: `www`

Browseable: Yes

Writable: No

Description: This folder contains the scripts that actually run the DiGIR Provider service. It should be accessible by web browsers, but should not be writable. Files contained in this folder are described in Appendix A.

Web Server Configuration

This section provides an overview of how to configure the web server (Apache or IIS) to work with the DiGIR Provider service. In both cases, the installation is described for Windows systems. The Apache instructions will be quite similar for installation on an Linux box.

After following the example web server configurations below, you should be able open a browser to http://localhost/digir/index.php. There will be several error messages displayed- this is normal at this stage of installation and is easily corrected by following the steps described below.

Apache

The primary configuration file for Apache is httpd.conf located in APACHE_INSTALL_DIR\conf which on windows might be c:\program files\Apache Group\Apache2\conf or /etc/httpd/conf on Linux.

The file is plain text with pseudo xml sections for configuring properties and functionality of the web server. You must read the manual to be certain of any changes you make to this file, otherwise you may inadvertently create a security hole that will be quickly breached.

An example configuration section for Apache 2 running mod_php with the DiGIR Provider service unpacked in the folder C:\var\ is shown below. Please check that the configuration settings are appropriate for your system if you are just copying the configuration information.

```
#PHP Configuration section - this portion enables the PHP interpreter
#The DiGIR administrative interface.  The settings allow access
#only from the localhost, meaning that you can only access the admin
# folder from the machine that is running the web server.  The admin
# folder is physically located at c:/var/DiGIRprov/admin, and is
# accessible through the web browser at the address: http://localhost/digir/admin
Alias /digir/admin C:/var/DiGIRprov/admin
<Directory C:/var/DiGIRprov/admin>
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
    DirectoryIndex index.html, index.php
</Directory>

#Alias for the DiGIR Provider service.  The settings are for a
#provider that was installed to c:/var/DiGIRprov, with the default
```

```
#folder name (www) for the location of the DiGIR php scripts.
Alias /digir C:/var/DiGIRprov/www
<Directory C:/var/DiGIRprov/www>
    AllowOverride None
    Order allow,deny
    Allow from all
    DirectoryIndex index.html, index.php
</Directory>
```

IIS

This installation guide assumes IIS 4.x or higher is installaed on the server. Adminstration of the web server (both IIS and PWS) is accomplished through a graphic interface rather than the editing of text files, as in Apache. There are two options available for configuring DiGIR with IIS.

New Web Site

The creation of a new web site in IIS isolates the DiGIR Provider files from all other activity on the web server. Howver, it requires that the user be able to create a host header name on the root domain of the server. Some DNS registrar services (e.g. EasyDNS) permit the creation of host header names, or aliases, directly by a user.

Activate the Internet Service Manager for IIS, and at the root entry, add a new Web Site. Follow the directions of the IIS Wizard using the following properties:

- Description=IIS internal name of the web site (e.g. Digir)
- Host Header=the host header address created for the DNS (e.g. digir.*hostname.ca*)
- Local Path=physical path where the DiGIR entry point is located. These files are normally found in the <*drive*>\digirprov\www directory
- Anonymous Access=Allowed
- Access Permissions=Read, Execute (Including Script)

Once the wizard is finished, select the new Web Site, and open its properties. Make the following adjustments:

- Directory Security=Enable Anonymous Access; user is the Internet Guest Account (e.g. IUSR_*localhost)*
- Application Name=Digir
- Default Document=enabled
- Default Document Name=Index.php

Next, create a new virtual directory within the DiGIR web site with the following properties:

- Local Path=physical path where the DiGIR admin files are located. These files are normally found in the <*drive*>\digirprov\admin directory

- Directory Security=Disable Anonymous Access; set access only to the user responsible for administrating the site (e.g. Administrator)

The DiGIR provider should now be available at the address:

http://Header.localhost

Virtual Directory

If a DNS Header can not be set up for the DNS of the server, the DiGIR provider can be set up within an existing web site on IIS. Activate the Internet Service Manager for IIS, and select the active web site in which DiGIR will be enabled. Create a new virtual web directory called "Digir". Navigate to the new directories properties, and set them accordingly:

- Local Path=physical path where the DiGIR entry point is located. These files are normally found in the <*drive*>\digirprov\www directory
- Access Permissions=Read, Execute (Including Script)
- Directory Security=Enable Anonymous Access; user is the Internet Guest Account (e.g. IUSR_*localhost)*
- Application Name=Digir
- Default Document=enabled
- Default Document Name=Index.php

Create a second virtual directory, as a sub-directory of Digir, called Admin. This virtual directory needs to have the same properties as the Digir directory, *except*:

- Local Path=physical path where the DiGIR admin files are located. These files are normally found in the <*drive*>\digirprov\admin directory
- Directory Security=Disable Anonymous Access; set access only to the user responsible for administrating the site (e.g. Administrator)

For the DiGIR provider to function correctly, Access Permissions to the physical files located on the hard drive must also be assigned to the Internet Guest Account. Open Windows Explorer, and navigate to the root directory in which the provider files are located (e.g. <*drive*>\digirprov). Right-clikc, and go to the Properties of the directory. Under the Security tab, add the Internet Guest Account, and give the account Read, Write, and Change permissions.

DiGIR Provider Configuration

Overview

Configuration of DiGIR involves editing of operating parameters for the service (by modifying localconfig.php), setting the description of the DiGIR Provider service (providerMeta.xml), and adding resources (connections to databases) by creating a database configuration file and adding an entry in the resource list file resources.xml. An aoutomated tool for editing these files is provided, the DiGIR Configurator, which is available inside the administrative tools directory:

http://localhost/digir/admin/setup.php

Operating Parameters (localconfig.php)

DiGIR operational defaults are controlled by defining constants in the file localconfig.php which overrides the default values defined in DiGIR_globals. It should not be necessary to edit any other file in the DIGIR_ROOT/www folder in order to have a functional default DiGIR provider installation.

The default settings are helpful for debugging and configuring an installation, however they are not optimal for a fully configured provider. The additional information these settings provide can also reduce the security of your installation. The following default values should be changed once your installation is functioning satisfactorily.

Using a web browser, open the url to the location where you installed the DiGIR provider service. For the example installation in this document, this would be:

http://localhost/digir/index.php

You should see an error message something like the following:
Fatal Error due to a configuration problem. The file:
localconfig.php
must exist in the same folder as DiGIR.php
If this is a new installation, copy
www/localconfig_dist.php
to
www/localconfig.php

To resolve this problem, copy the file localconfig_dist.php to localconfig.php.

Security Related Settings

DIGIR_ALLOWDEBUG
Name: DIGIR_ALLOWDEBUG
Default: TRUE
Example: define('DIGIR_ALLOWDEBUG',FALSE);

The provider will generate debug information that is inserted inline with the xml output when a parameter debug=1 is added to the URL when calling the DiGIR Provider service. The information it provides can be quite helpful for tracking down configuration problems, but can also expose the connection string and structure of your database. For this reason, you should set this option to FALSE unless it is required.

DIGIR_ALLOW_CONFIG_DUMP
Name: DIGIR_ALLOW_CONFIG_DUMP
Default: TRUE
Example: define('DIGIR_ALLOW_CONFIG_DUMP',FALSE);

Setting DIGIR_ALLOW_CONFIG_DUMP to TRUE will cause all the defines in this file to be dumped to the browser when this script is called directly. This can be useful during the installation process when checking the actual values of the config settings.

Warning

Set DIGIR_ALLOW_CONFIG_DUMP to FALSE or delete this entry when you have finished configuring the provider service. Whilst not a direct security threat, the information it provides could be helpful to someone with destructive intentions.

Performance Related Settings

DIGIR_MAX_RUNTIME
Name: DIGIR_MAX_RUNTIME
Default: 120
Example: define('DIGIR_MAX_RUNTIME',300);

Sets the maximum runtime of the script in seconds. The default value allows the script to run for two minutes, which should be ample except for very large result sets or poorly designed databases (e.g. no indexing).

DIGIR_USE_CACHE
Name: DIGIR_USE_CACHE
Default: FALSE
Example: define('DIGIR_USE_CACHE',TRUE);

Turns on or off the use of response caching for hte DiGIR Provider service. It is recommended that caching is turned off until the installation is completed and determined to be operating correctly. Setting this entry to TRUE will significantly improve throughput, especially for the response of Metadata operations.

DIGIR_METADATA_CACHE_LIFE_SECS
Name: DIGIR_METADATA_CACHE_LIFE_SECS
Default: 86400
Example: define('DIGIR_METADATA_CACHE_LIFE_SECS',86400);

Number of seconds that cached metadata will be used before forcing an update. Default is one day (86400 seconds).

DIGIR_FORMAT_CACHE_LIFE_SECS
Name: DIGIR_FORMAT_CACHE_LIFE_SECS
Default: 86400
Example: define('DIGIR_FORMAT_CACHE_LIFE_SECS',86400);

Number of seconds a downloaded record format definition is cached before checking for a new version.

DIGIR_RESULTSET_CACHE_LIFE_SECS
Name: DIGIR_RESULTSET_CACHE_LIFE_SECS
Default: 0
Example: define('DIGIR_RESULTSET_CACHE_LIFE_SECS',0);

Indicates the time in seconds that a search response will remain in the cache. A unique request is identified by the combination of the parameters used to invoke the function call getContent(). Default duration is zero, which turns off resultset caching.

DIGIR_INVENTORY_CACHE_LIFE_SECS
Name: DIGIR_INVENTORY_CACHE_LIFE_SECS
Default: 3600
Example: define('DIGIR_INVENTORY_CACHE_LIFE_SECS',3600);

Indicates the time in seconds that an inventory response will remain in the cache. A unique request is identified by the combination of the parameters used to invoke the function scanColumn(). Default duration is one hour.

DIGIR_LOG_LEVEL
Name: DIGIR_LOG_LEVEL
Default: 7 (PEAR_LOG_DEBUG)
Example: define('DIGIRLOG_LEVEL',6);

The value of DIGIRLOG_LEVEL sets the verbosity of the log files. A normal operating value of 6 is recommended. The default value of 7 will write additional information to the log that may be of help when tracking down problems with a new (or altered) installation.

Note
For the PHP savvy, the DiGIR Provider service uses the standard PEAR Log facility which may be configured to use a variety of logging options, including logging to a database. The DiGIR Provider code should work independently of the type of log storage you choose.

DIGIR_STATUSINTERVAL
Name: DIGIR_STATUSINTERVAL
Default: 3600
Example: define('DIGIR_STATUSINTERVAL',600);

For each request serviced by the DiGIR Provider, a record of the type of request and the time stamp of the request is kept in an array that is cached to disk. Records older than the current time minus DIGIR_STATUSINTERVAL seconds are deleted from the array. The total number of each type of request is computed from the array and is appended as a diagnostic to each response generated by the DiGIR Provider. If the provider is running under a very high load (hundreds - thousands or requests per hour), then you should set this value smaller than the default of one hour.

Other Commonly Modified Settings

DIGIR_CONFIG_DIR
Name: DIGIR_CONFIG_DIR
Default: ../config/
Example: define('DIGIR_CONFIG_DIR','c:/my_secure_folder/DiGIR/config/');

This entry identifies the path to the folder which contains all the configuration files. Note the use of forward slashes for path separators.

DIGIR_CACHE_DIRECTORY
Name: DIGIR_CACHE_DIRECTORY
Default: ../cache/
Example: define('DIGIR_CACHE_DIRECTORY','c:/temp/DIGIRcache');

This is the location of the cache files. It is ok to delete content from this folder any time since DiGIR will simply rebuild the files as necessary, and deleting the files is the best way to force DiGIR to reload cached information (such as after editing provider metadata). Note that the size of the cache can be fairly large depending on the popularity of your DiGIR Provider service. On some systems, it is probably better to locate the cache in your regular temp folder as shown in the example.

Important: The path defined in this entry must exist and be writable by the web server service.

DIGIR_LOG_NAME
Name: DiGIR_LOG_NAME
Default: log.txt
Example: define('DIGIR_LOG_NAME`,strftime('digir_%d%m%Y.txt'));

Specifies the name of the log file for the DiGIR Provider service. Note that this specifies the file name only. To set name of the folder that contains the log files, use DIGIR_LOG_PATH. The example shows how to create a log that will use a new file for every day.

DIGIR_LOG_PATH
Name: DIGIR
Default: ../log/
Example: define('DIGIR_LOG_PATH','c:/logs/digir');

Sets the name of the folder that will contain the log files for the DiGIR Provider service. This folder must exist, and be writable by the process running the PHP script.

DIGIR_STATUSPICKLE
Name: DIGIR_STATUSPICKLE
Default: DIGIR_LOG_PATH/DiGIRStatus.txt
Example: define('DIGIR_STATUS_PICKLE','c:/temp/digir/status.txt');

The name of the file that is used to cache DiGIR Provider service status information between requests. This file must be in a location that is writable by the process running the PHP scripts.

Resource Database Configuration

One of the greatest difficulties in attempting to share natural history collections in a distributed environment is the simple fact that there is no established standard for database content, schema, structure.

Nevertheless, almost all collection and observation databases share similarities which may be exploited to perform search and retrieval of common information. The Darwin Core attempts to provide a set of guidelines for addressing this commonality regardless of the underlying mechanism for storing the record content.

Darwin Core

The Darwin Core profile provides a list of suggested access points and recommendations for their usage for searching natural history specimen and observation databases. It provides suggestions for stringifying queries such that they are protocol independent, and provides guidance as to the content, structure and format of records retrieved from an information server supporting the Darwin Core. Currently, the Darwin Core schema is in its second version, and contains 48 elements, or access points.

Darwin Core Elements

Name	Required	Type	Description
Date Last Modified	Y	DateTime	ISO 8601 compliant stamp indicating the date and time in UTC(GMT) when the record was last modified. Example: the instant "November 5, 1994, 8:15:30 am, US Eastern Standard Time" would be represented as "1994-11-05T13:15:30Z" (see W3C Note on Date and Time Formats). (What to do when this date-time is unknown? Use Date-Time first "published"?
Institution Code	Y	Text	A "standard" code identifier that identifies the institution to which the collection belongs. No global registry exists for assigning institutional codes. Use the code that is "standard" in your discipline.
Collection Code	Y	Text	A unique alphanumeric value which identifies the collection within the institution
Catalog Number	Y	Text / Numeric	A unique alphanumeric value which identifies an individual record within the collection. It is recommended that this value provides a key by which the actual specimen can be identified. If the specimen has several items such as various types of preparation, this value should identify the individual component of the specimen
Scientific Name	Y	Text	The full name of lowest level taxon the Cataloged Item can be identified as a member of; includes genus name, specific epithet, and subspecific epithet (zool.) or infraspecific rank abbreviation, and infraspecific epithet (bot.) Use name of suprageneric taxon (e.g., family name) if Cataloged Item cannot be identified to genus, species, or infraspecific taxon.
Basis of	N	Text	An abbreviation indicating whether the record

record			represents an observation (O), living organism (L), specimen (S), germplasm/seed (G), etc.
Kingdom	N	Text	The kingdom to which the organism belongs
Phylum	N	Text	The phylum (or division) to which the organism belongs
Class	N	Text	The class name of the organism
Order	N	Text	The order name of the organism
Family	N	Text	The family name of the organism
Genus	N	Text	The genus name of the organism
Species	N	Text	The specific epithet of the organism
Subspecies	N	Text	The sub-specific epithet of the organism
Scientific Name Author	N	Text	The author of a scientific name. Author string as applied to the accepted name. Can be more than one author (concatenated string). Should be formatted according to the conventions of the applicable taxonomic discipline.
Identified By	N	Text	The name(s) of the person(s) who applied the currently accepted Scientific Name to the Cataloged Item.
Year Identified	N	Numeric	The year portion of the date when the Collection Item was identified; as four digits [-9999..9999], e.g., 1906, 2002.
Month Identified	N	Numeric	The month portion of the date when the Collection Item was identified; as two digits [01..12].
Day Identified	N	Numeric	The day portion of the date when the Collection Item was identified; as two digits [01..31].
Type Status	N	Text	Indicates the kind of nomenclatural type that a specimen represents. (This is incomplete because type status actually describes the relationship between a name and a specimen [or ternary relationship between a specimen, name, and publication].) In particular, the type status may not apply to the name listed in the scientific name, i.e., current identification. In rare cases, a single specimen may be the type of more than one name.
Collector Number	N	Text	An identifying "number" (really a string) applied to specimens (in some disciplines) at the time of collection. Establishes a links different parts/preparations of a single specimen and between field notes and the specimen.
Field Number	N	Text	A "number" (really a string) created at collection time to identify all material that resulted from a collecting event.
Collector	N	Text	The name(s) of the collector(s) responsible for collection the specimen or taking the observation
Year Collected	N	Numeric	The year (expressed as an integer) in which the specimen was collected. The full year should be expressed (e.g. 1972 must be expressed as "1972" not "72"). Must always be a four digit integer [-9999..9999]
Month Collected	N	Numeric	The month of year the specimen was collected from the field. Possible values range from 01...12 inclusive
Day Collected	N	Numeric	The day of the month the specimen was collected from the field. Possible value ranges from 01..31 inclusive

Julian Day	N	Numeric	The ordinal day of the year; i.e., the number of days since January 1 of the same year. (January 1 is Julian Day 1.)
Time of Day	N	Numeric	The time of day a specimen was collected expressed as decimal hours from midnight local time (e.g. 12.0 = mid day, 13.5 = 1:30pm)
Continent Ocean	N	Text	The continent or ocean from which a specimen was collected.
Country	N	Text	The country or major political unit from which the specimen was collected. ISO 3166-1 values should be used. Full country names are currently in use. A future recommendation is to use ISO3166-1 two letter codes or the full name when searching
State Province	N	Text	The state, province or region (i.e. next political region smaller than Country) from which the specimen was collected. There is some suggestion to use the values described in ISO 3166-2, however these values are in a continual state of flux and it appears unlikely that an appropriate mechanism (by ISO) will be in place to manage these changes. Hence it is recommended that where possible, the full, unabbreviated name should be used for storing information. The server should optionally handle abbreviations as an access point. Note: this is a recurring theme (country and state) abbreviation. Check the existence of an attribute type to deal with abbreviations from the bib-1 profile
County	N	Text	The county (or shire, or next political region smaller than State / Province) from which the specimen was collected
Locality	N	Text	The locality description (place name plus optionally a displacement from the place name) from which the specimen was collected. Where a displacement from a location is provided, it should be in un-projected units of measurement
Longitude	N	Numeric	The longitude of the location from which the specimen was collected. This value should be expressed in decimal degrees with a datum such as WGS-84
Latitude	N	Numeric	The latitude of the location from which the specimen was collected. This value should be expressed in decimal degrees with a datum such as WGS-84
Coordinate Precision	N	Numeric	An estimate of how tightly the collecting locality was specified; expressed as a distance, in meters, that corresponds to a radius around the latitude-longitude coordinates. Use NULL where precision is unknown, cannot be estimated, or is not applicable.
Bounding Box	N	Bounding Box	This access point provides a mechanism for performing searches using a bounding box. A Bounding Box element is not typically present in the database, but rather is derived from the Latitude and Longitude columns by the data provider
Minimum Elevation	N	Numeric	The minimum distance in meters above (positive) or below sea level of the collecting locality.
Maximum Elevation	N	Numeric	The maximum distance in meters above (positive) or below sea level of the collecting locality.
Minimum Depth	N	Numeric	The minimum distance in meters below the surface of the water at which the collection was made; all

			material collected was at least this deep. Positive below the surface, negative above (e.g. collecting above sea level in tidal areas).
Maximum Depth	N	Numeric	The maximum distance in meters below the surface of the water at which the collection was made; all material collected was at most this deep. Positive below the surface, negative above (e.g. collecting above sea level in tidal areas).
Sex	N	Text	The sex of a specimen. The domain should be a controlled set of terms (codes) based on community consensus. Proposed values: M=Male; F=Female; H=Hermaphrodite; I=Indeterminate (examined but could not be determined; U=Unknown (not examined); T=Transitional (between sexes; useful for sequential hermaphrodites)
Preparation Type	N	Text	The type of preparation (skin. slide, etc). Probably best to add this as a record element rather than access point. Should be a list of preparations for a single collection record.
Individual Count	N	Numeric	The number of individuals present in the lot or container. Not an estimate of abundance or density at the collecting locality.
Previous Catalog Number	N	Text	The previous (fully qualified) catalog number of the Cataloged Item if the item earlier identified by another Catalog Number, either in the current catalog or another Institution / catalog. A fully qualified Catalog Number is preceded by Institution Code and Collection Code, with a space separating the each sub element. Referencing a previous Catalog Number does not imply that a record for the referenced item is or is not present in the corresponding catalog, or even that the referenced catalog still exists. This access point is intended to provide a way to retrieve this record by previously used identifier, which may used in the literature. In future versions of this schema this attribute should be set-valued.
Relationship Type	N	Text	A named or coded valued that identifies the kind relationship between this Collection Item and the referenced Collection Item. Named values include: "parasite of", "epiphyte on", "progeny of", etc. In future versions of this schema this attribute should be set-valued.
Related Catalog Item	N	Text	The fully qualified identifier of a related Catalog Item (a reference to another specimen); Institution Code, Collection Code, and Catalog Number of the related Cataloged Item, where a space separates the three sub-elements.
Notes	N	Text	Free text notes attached to the specimen record

Darwin Core Mapping

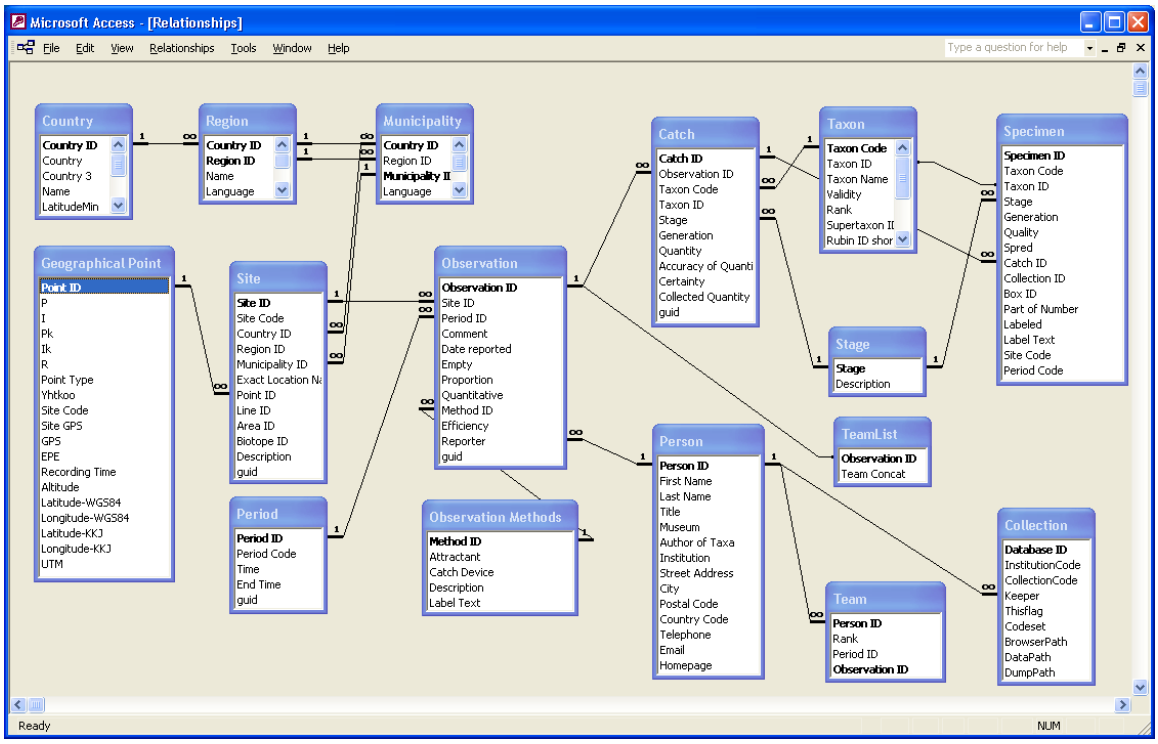
Three methods may be employed for the delivery of Darwin Core elements to a data provider. The first relies on joins between tables in the database to occur at the DiGIR provider level. Because this can be generated by the configurator, it may initially seem to be the easier of the methods. However it results in the slowest search response times for the provider.

The second method is to generate a Query/View within the database itself, which maps fields from the database schema to relevant Darwin Core fields. By employing this method, joins between tables are maintained at the database level, and search times improve due to the greater efficiency of the database server.

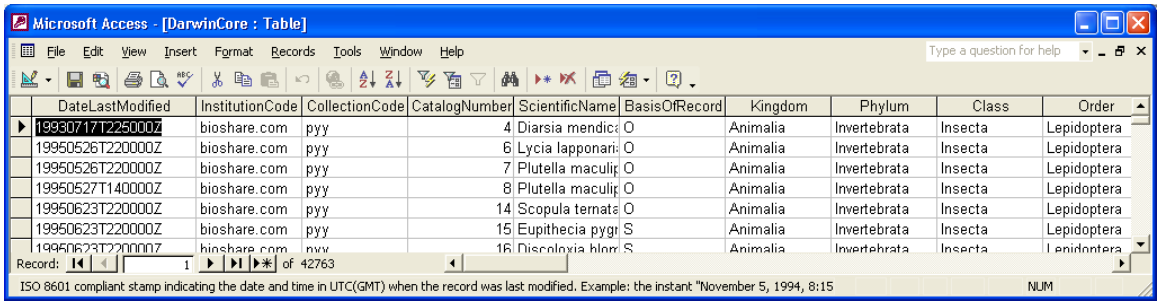
A final method is to extract relevant Darwin Core fields from the database schema into a static table. While this may result in the fastest search times by the provider, it can also result in information being served which is out of date, if the static table is not routinely updated to reflect changes in the database.

Example Database Schema (Biotella.mdb)

Biotella is one of many available observation and specimen database tools (http://www.bioshare.net/biotella). It is an "Open source" Microsoft Access Visual Basic application which can be used as a data source for a DiGIR Provider.



Biotella.mdb Schema



Extracted Darwin Core Table in BioTella

Using the Configurator

Ensure that a copy of the file digir.css is copied from the /DiGIR_ROOT /www directory to the Digir_Root directory. This file contains formatting information for the configurator web pages. To access the configurator, open the web browser and navigate to the address indicated above. If the web server has been configured correctly, the user will be required to log in using credentials for the account assigned to \diGIR_ROOT\Admin.

Provider Metadata

The main screen of the configurator provides summary information about the DiGIR provider installation. In the panel on the left, metadata for the provider can be accessed by clicking on the lick under the heading 'provider'. Mandatory fields are identified by color, and changes can be saved by clicking the 'Save' button located at the bottom of the page.

NOTE – changes made to these files will not be reflected, unless the Apache provider service is stopped and re-started, or the IIS web server is stopped and re-started. Alternatively, files in the DiGIR_ROOT\cache directory can be manually deleted.

Provider Metadata Fields

1. Name – name of the DiGIR Provider installation, used to identify the provider to search portals
2. Host Name – name of the host computer on which the provider is lcoated
3. Host Code – unique code for the host
4. Host Related Information – a URL linking to related information for the host
5. Contact Information – contact information for the host
6. Abstract - a short text description of the host

Resource Configuration

New resources for the DiGIR provider can be added by clicking on the 'Add Resource' button located in the panel on the left. Existing resources can be edited by clicking on links under the 'resources' heading. See *Resource Configuration* for more detailed information.

Resource configuration affects at least two configuration files. The file resources.xml contains a list of resource names and the name of the file that contains the associated configuration information. This resource configuration file contains metadata describing the resource, database connection information and concept to database column matching.

The resource configuration files are XML files that define:

1. Metadata about the resource
2. Database connection information
3. Database schema representation
4. Mapping between database columns and conceptual schema elements

Resource Metadata Fields

1. *Code* – uniquely identifying code for the resource
2. *Name* – name of the resource. The name is what will be used by DiGIR portals to uniquely identify the resource.
3. *Related Information* – a URL linking to related information for the resource
4. *Contact Information* – contact information for the resource
5. *Abstract* – a short text description of the resource
6. *Keywords* – keywords used by DiGIR portals to search for resources
7. *Citation* – an example of how data from the resource should be cited
8. *Use Restrictions* – a description of any use limitations associated with data from the resource
9. *Record Limit* – maximum number of records which can be returned by the provider
10. *Record Identifier*–unique code identifying records from the resource
11. *Record Basis* – Observation, collection, etc
12. *MinQueryTermLength* – minimum number of characters which must be supplied, for a successful search to be completed
13. *MaxSearchResponseRecord s*– maximum number of records which can be returned by a single search
14. MaxInventroryResponseRecords – maximum number of records which can be returned by a single Inventory query

Resource Database Connection

Once the resource metadata has been filled in, the configurator will request information required for connecting to the resource database. Depending on the type of database which is used, a different connection string will be required to provide a programmatic link to the DiGIR provider. Several example connection strings for databases supported by the DiGIR provider are identified below.

MySQL

Provider=MySQLProv;Data Source=mydb;User Id=UserName;Password=asdasd

Microsoft Access

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=<*drive*>\somepath\mydb.mdb;User Id=admin;Password=

ODBC

DSN=myDsn;Uid=username;Pwd=

Microsoft SQL Server (OLE DB)

Standard Security, Local Server:

Provider=sqloledb;Data Source=*somesource*;Initial Catalog=*dbname*;User Id=;Password=

Connect via an IP address:
"Provider=sqloledb;Data Source=190.190.200.100,1433;Network Library=DBMSSOCN;Initial Catalog=*dbname*;User ID=;Password=;"
(DBMSSOCN=TCP/IP instead of Named Pipes, at the end of the Data Source is the port to use (1433 is the default))

Resource Schema and Mappings

Once the connection information for the resource has been defined, the user will be asked to select the Root Table and Key field from the database. If a Query/View/Table has been previously established in the database (as discussed earlier), select this from the drop-down list. The next step can then be skipped (defining table joins). Mapping the Darwin Core fields to the root table is the last step.

Mappings can be established. In the table presented to the user, select the Root Table from the drop-down box under 'Table'. If Darwin Core fields have been properly aliased in the database, all available fields should be filled in automatically by the cofigurator. If fields are not filled in, double-check their spelling, or ensure that they were included in the Query/View/Table. Only those fields indicated (first 5) are mandatory for a resource to be established.

At this stage, the user can also determine whether a Darwin Core field is Searchable and/or Returnable. Searchable fields are used by DiGIR Portals to search for information stored within the resource. Data in fields which are set as Returnable will be included in any results returned from a successful search. An example of where this could be useful is in the case of Endangered Species Data. Setting spatial location fields to Searchable will allow a portal to search for records within a defined area. However, setting the same field as non-returnable keeps detailed spatial location data from being retrieved.

DiGIR Provider Testing

The DiGIR Provider service is a service application that has no implicit user interface, but rather is intended for machine to machine communications. As such, human interaction with the operation of a DiGIR Provider service is generally limited to monitoring the status of the provider through review of the dynamic status information, the log files, and any diagnostic information that may be reported by users of the service.

Basic Configuration

Testing the provider can be done using several files included in the installation. Basic configuration errors are reported by opening a browser, and navigating to:

http://localhost\digir\digir.php

Errors in configuration settings will be reported to the user as simple text messages, and include suggested fixes for the problem.

Provider Response

Testing of the search responses for Record Searches, Inventory Searches, and Metadata Searches can be done by navigating to:

http://localhost/digir/test/eg_search.php

http://localhost/digir/test/eg_inventory.php

http://localhost/\digir/test/eg_meta.php

Each of these pages provides a basic interface for submitting xml requests to the DiGIR provider. By editing the supplied XML to reflect the DiGIR Provider location and the name of the resource to be searched (where necessary), simple XML responses from the provider can be elicited.