

Entering Agreements in the DAITSS System

Table of Contents

1	Preface	1
2	Establishing Agreements	2
3	DADI and Agreements	2
3.1	Inserting data into DADI	3
3.1.1	CONTACT Table	4
3.1.2	ACCOUNT Table	4
3.1.3	PROJECT Table	5
3.1.4	ACCOUNT PROJECT Table	5
3.1.5	SUB ACCOUNT Table	6
3.1.6	OUTPUT REQUEST Table	6
3.1.7	MEDIA TYPE Table	6
3.1.8	ARCHIVE LOGIC Table	7

1 Preface

This document is intended for DAITSS administrators and describes the steps required to enter user agreement information in the Digital Archive Database Instance (DADI). At the time of writing, DAITSS uses a MySQL database which can be accessed traditionally through the command line, or alternatively through a PHP-based administrative interface. Most of the examples contained in this document use the PHP interface to illustrate individual steps, but all steps can be accomplished using the basic MySQL command-line interface. The method of execution of a step is somewhat irrelevant, what is more important is the nature and order of the steps.

2 Establishing Agreements

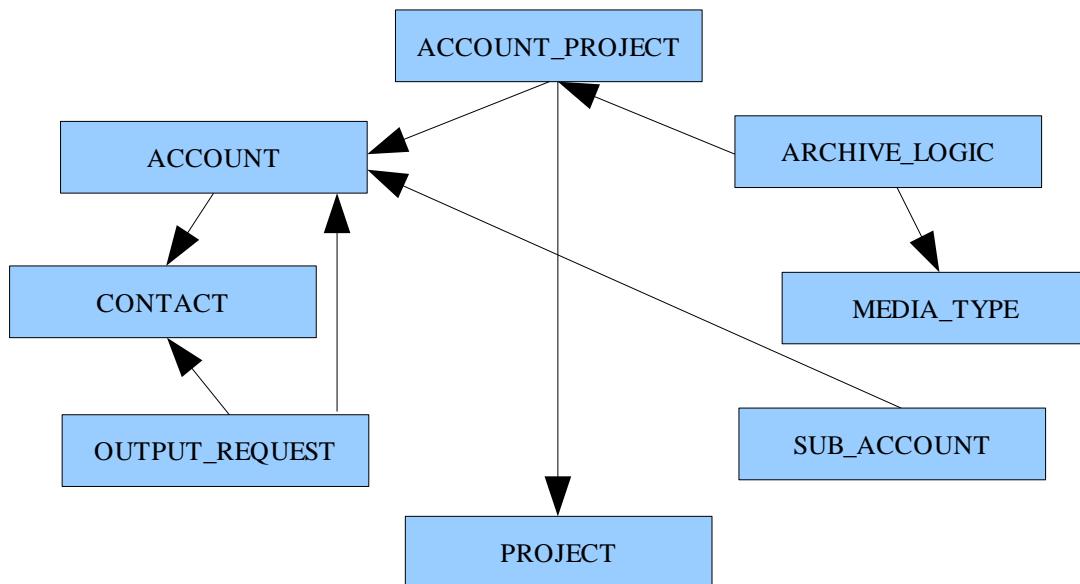
Before any data can be ingested for a particular account, agreements must be established with the account holder. Agreements are established on an account basis and describe information regarding associated projects and preservation logic which

specifies how materials submitted by an account holder are to be treated according to project and media type (format). Agreements also specify contact information including the rights of contacts to request output from DAITSS in the form of reports, dissemination, and withdrawal. Agreements must be established using the official form contained in the document FCLALibraryAgreement_FDA.doc (“FCLA -- LIBRARY AGREEMENT”) which is primarily a legal document binding an account holder to the archive. Of equal importance are the two appendices which are to be completed by the account holder. Appendix A is used to provide contact information while Appendix B is used to provide preservation level information. More information regarding agreements can be found in the document entitled “Setting Up a Library to Contribute to the FCLA Digital Archive” (Submitting Materials to the FCLA Digital Archive.doc).

3 DADI and Agreements

The Digital Archive Database Instance (DADI) is the database back-end for the DAITSS software. For a Submission Information Package (SIP) to be ingested, formal agreements for the related account and project (as specified in “FCLA -- LIBRARY AGREEMENT” and its appendices) must first be entered into DADI. This entails data entry into 8 separate tables. Those tables are represented in Graph 1 below. Note the directional arrows which represent foreign key relationship constraints. The table from which an arrow emanates is the table containing the foreign key, while the destination table represents the target of that foreign key constraint. This forms a precedence graph that illustrates the order in which tables must be updated to guarantee that foreign key constraints are met. The target of a foreign key constraint must be updated before the table containing the foreign key declaration.

While there are many possible precedence sequences that can be constructed from Graph 1, an intuitive sequence that matches all precedence requirements is: CONTACT – ACCOUNT – PROJECT – ACCOUNT_PROJECT – SUB_ACCOUNT – OUTPUT_REQUEST – MEDIA_TYPE – ARCHIVE_LOGIC. It is possible that some tables will not need additional rows to be inserted, for example the MEDIA_TYPE table may already include all mime-types specified in an agreement. It is imperative, however, that every table be examined to ensure that its contents are sufficient.



Graph 1: DADI tables involved in agreements

3.1 Inserting Data into DADI

As mentioned previously, there is more than one precedence sequence that can be derived from the precedence graph (Graph 1). The following sections list the suggested sequence and include any notes that may be pertinent for data entry into a specific table.

3.1.1 CONTACT Table

- Information to be entered in the CONTACT table is provided by an account holder completely in Appendix A of the “FCLA -- LIBRARY AGREEMENT”.
- The CONTACT table includes 5 fields for a physical address (ADDR_L1-5) of which only the first 2 are required.
- When entering a new row, it is not necessary to enter a value for the ID

field. This is an auto-incrementing field and will be supplied by the database.

3.1.2 ACCOUNT Table

- The CODE attribute is the primary key for the ACCOUNT table and must therefore be unique. In general, a 3-4 character code is sufficient. The code should be representative of the institution holding the account.
 - NAME should be the full name of the organization holding the account.
 - ADMIN_CONTACT specifies the contact responsible for administration and billing. The entry must be an existing key in the CONTACT table. Since this key is numeric and supplied by the database, it will be necessary to cross reference the correct key value in the CONTACT table. The same holds true for the TECH_CONTACT field. The easiest way to do this using the php administrative interface is to right-click on the browse button for the CONTACT table and select “Open link in new window”. Using this technique, you can read information from the CONTACT table and enter it in the ACCOUNT table.
 - REPORT_EMAIL should contain an email address to which all reports should be sent. The address is given in Appendix A of the FDA Agreement. It does not necessarily have to be the email address of a recognized contact. For example, email can be sent to an address with the sole purpose of collecting reports, thereby lessening the burden on a personal address and reducing the chance that a report would be inadvertently trashed or junked.
- ➔ **IMPORTANT NOTE:** In development and test databases, the email address entered here should be a FDA-controlled address (usually a developer's address)

3.1.3 PROJECT Table

- Project codes for a given account are provided in the FDA Agreement document in the “PRESERVATION LEVEL TABLE” contained in Appendix B. All projects identified by the participating organization must be entered in the PROJECT table. Project codes must be unique and can be shared among accounts. There is no need to provide any account information in the code itself. Associations will be made between accounts and projects in the ACCOUNT_PROJECT table later.

3.1.4 ACCOUNT_PROJECT Table

- In this table, associations between all project codes and the accounts to which they pertain are created.
- The ID field is an auto-incrementing primary key provided by the database, there is no need to specify a value for this field during data entry.
- Entering information for the ACCOUNT and PROJECT fields is relatively straightforward, since both values are coded meaningfully.

3.1.5 SUB_ACCOUNT Table

- Any sub-accounts associated with an account are entered in the SUB_ACCOUNT table. Sub-accounts are provided in the FDA agreement and while they do not have an internal meaning to DAITSS, any sub-account information specified in a descriptor will be validated. If a package is received that contains a sub-account, that sub-account value will be matched against the associated account in the SUB_ACCOUNT table. If there is no such match, the package will be rejected despite the fact the sub-account codes are not used in any other archive-specific logic.

3.1.6 OUTPUT_REQUEST Table

- The OUTPUT_REQUEST table links permissions to contacts for materials belonging to a particular account. At the time of writing, there are 3 separate requests identified in the OUTPUT_REQUEST table. These are whether a contact may successfully request a report, whether they may request a dissemination of archive entities, and whether they may request a withdrawal of archive entities. These values are stored in separate fields as simple boolean values (actually in the case of mySql, currently there is no built-in boolean data type and so an enumeration of “TRUE” and “FALSE” strings is used). Again, information about request permissions is included in the FDA Agreement document. Primary administrative and technical contacts are always afforded full request permissions. Recall that since contacts are given auto-incrementing id values in the CONTACT table, it is necessary to cross-reference those values for entry into the OUTPUT_REQUEST table.

3.1.7 MEDIA_TYPE Table

- The MEDIA_TYPE table contains all mime-media types supported by the archive (in other words, mime-types for which DataFile subclasses have been implemented). This table is updated whenever new mime-types become supported and so it should always contain entries for the mime-types appearing in the FDA Agreement document. Despite this, it is still a good idea to double-check the presence of pertinent mime-types to clear up any misunderstandings and to short-circuit any problems down the road.

3.1.8 ARCHIVE_LOGIC Table

- This is the table that governs preservation level determination. Here an ACCOUNT_PROJECT/mime-type combination is mapped to the desired preservation level as provided in the FDA Agreement Appendix B. It is very important to remember that XML must be given a preservation level

of “FULL” for all accounts and projects. This ensures that descriptors are preserved for the lifetime of an intellectual entity. Being an intra-occurrence constraint, this can only be enforced a few ways, through software, triggers, or careful data entry. Since tables are updated manually and not through software, and since triggers are not yet supported in mySql, it is necessary to be very careful when coding agreements in the ARCHIVE_LOGIC table, making sure that XML always has full preservation.

- Typical agreements explicitly specify preservation levels for a handful of formats and use the phrase “all other formats” as a catch-all for all unspecified formats. In the ARCHIVE_LOGIC table, this translates to a media type of “default”. If a default preservation level is not defined in the agreement, a new row must be added to the table with a MEDIA_TYPE of “default” and a preservation level of “BIT”.
- As with previous tables that have auto-incrementing primary keys, ID values should not be specified during data entry. Also, since the ACCOUNT_PROJECT foreign key references such a primary key, that value will have to be cross-referenced in a manner described for entering ADMIN_CONTACT keys in the the ACCOUNT table.