

DataFetch Model DF3D Installation Advice for IT Staff

Introduction

Installation of the DataFetch DF3D-based solutions requires a few considerations which the institution's IT staff needs to attend to. These considerations should be easy to comply with, but in the event of difficulties or questions, please contact DataFetch for direct technical assistance.

Program Installation

The DataFetch Unilogger program, or the DataFetch Event Log Recorder program (ELR2), are supplied as self-installing executables; launching them will install the programs, and create desktop icons on the computer. The programs also include an 'uninstall' option, accessible via the 'Start/All Programs/DataFetch/University Logger' path, or the 'Start/All Programs/DataFetch/ELR2' path, or via the Control Panel 'Add/Remove Programs' Icon.

Local Operation

As delivered, both DataFetch programs are set up for operation completely local to the laptop or PC it is running on. This means that the directory paths are defaulted to the same directory as where the program itself is installed (C:\Program Files\DataFetch\University Logger, or C:\Program Files\DataFetch\ELR2), and there is a sample student database (studentdb.xls) installed into the same paths. By using the supplied sample barcode sheets, the programs can be immediately tested, and can be used for training and familiarization purposes.

Administrator/User differences

Upon installation, both programs install a shortcut to the program on the desktop. In some cases, the program might be installed under an 'administrator' login, in which case, the desktop icon must be copied to the 'user' desktop in order for the program to be easily accessible to a user.

Both programs store their options and preferences in the system Registry, under the following key and its subordinates: HKEY_LOCAL_MACHINE\SOFTWARE\DATAFETCH. The programs, to a large extent, share options and preferences, so configuring one of the programs will also configure common options for the other program. If the programs were installed under 'administrator' rights, and a user wishes to use it, it will

be necessary to set the permissions for read/write access to this key, for the user. This is easily accomplished by opening the registry, using 'regedit' (click on 'Start', 'Run', enter 'regedit' in the text box, and click on 'OK'. After finding the 'DataFetch' key, right click on it, and select 'permissions...', and set permission for both read and write for the user account.

Network Installation

While the programs may be operated entirely local to the PC or laptop it is installed on, most institutions will want or need the critical log files and student database file to be stored on a networked server. By selecting 'Options' in the menu, the administrator can set up paths to network storage for all of the critical files.

Some files, such as the student database file, as well as the outside guest restriction file, should be accessible to all potential users of the program. For example, if Unilogger is installed in five separate residence halls, all five installations should have their file paths to the student database file, as well as the outside guest restricted file, pointing to a single network location

In the case of the Unilogger program, the daily log files, as well as the current master log file, will be unique to each installation; therefore, those paths should be set for individual directories on the network server.

While serving common files via a networked server would seem to be the best choice, the speed of the network may slow program operation. This is especially true of the student database file, access to which is important for both programs. Users are advised to consider using the DataFetch-supplied encryption utility, and storing the student database file locally on each workstation; encryption will keep the data secure.

In the interests of rapid access, the student database file is 'opened' by the applications when the program is started, and remains 'opened' until the program is closed. This means that any updates to the student database file should be deferred until the program is not being used. Both the Unilogger, as well as the ELR2 application, contain an option which can be used to schedule a shutdown in order to permit access to the file for update by the network server.

The studentdb.xls or studentdb.xlsx File

The program requires that the university/college supply a database of all students covered by the security system.

The file may be in one of two forms. For smaller institutions (fewer than 10,000 students), the file may be in Excel 2003 or earlier form. This form performs lookup functions by a linear (i.e., top to bottom) search algorithm, which, on most laptops or PC's, is fast enough to not present any annoying delay when scanning student ID's.

For larger institutions (up to 130,000 ID numbers and beyond), the .xlsx form using Excel 2007 or later is recommended (and required, if the database is larger than 65,536 students). When using this form, the lookup employs a successive approximation search technique that dramatically speeds up access to the desired student. However, it does require that the database file be sorted, in ascending order, by the student ID number represented by the scan, swipe, or RFID access to the card.

In both cases, the format of the spreadsheet is extremely simple; it consists of one row for each student, with six columns (and an optional seventh column):

Column 1 contains the primary student ID number

Column 2 contains an optional secondary student ID number (see note)

Column 3 contains the student name

Column 4 contains the name of the student's residence hall

Column 5 contains the room number

Column 6 contains the student's telephone number

Column 7 contains a text note with any restrictions the student may be under. In Unilogger version 3.0 and later, or ELR version 2.0 or later, the student restriction can be handled by a separate file, so this field is optional.

In Unilogger version 2.3 or earlier, or ELR Versions earlier than version 2.00, The row just after the last student should have a '0' in the first column; this indicates the end of the database. Later versions of these programs don't require the final '0' in the first column.

There are two additional rules which should be followed:

- **The entire spreadsheet** must be formatted as 'text'; this is because some student ID numbers may contain alpha characters. The program works on a 'string matching' basis. Regardless of formatting, no cell may contain a formula.
- **There must be no embedded blank lines**, no header lines, and no student ID numbers which are the number '0'.

Note: Some institutions use identity cards which carry a barcode, but the barcode represents a different number than the actual student ID number. If using the barcode in installations like this, the secondary ID number should be the number associated

with the barcode. As long as the barcode number, and the student ID number, has a one-to-one correspondence, the program will locate the student in the database.

An example of a student database useable for test purposes is shown below. Note that there are no embedded blank lines, no header lines, and the last row has a '0' in the first column. While it can't be seen in the graphic, the entire spreadsheet has been formatted as text.

The studentdb.xls may have up to 65,535 students.

The studentdb.xlsx File

For faster look-up and/or larger student databases, the .xlsx form of the file is preferred. The file is the same as described above, **except** that the file must be sorted according to whichever column represents the scanned or swiped student ID number. The sort is easily accomplished using Excel; simply highlight the column which represents the key, and select a data sort from the Excel menu; Excel will ask if the sort should be extended across the entire spreadsheet, which should be answered affirmatively.

	A	B	C	D	E	F	G
1	147239	1D000001	John Doe	Liberty Hall	340	781-555-1234	
2	147240	1D000002	Ted Smith	Jacobs Hall	567	781-555-1234	Restricted from hosting outside guests until 3-31-2011
3	147241	1D000003	Alex Wagner	Langdon Hall	110	781-555-1234	
4	147242	1D000004	Lisa Trachtman	Stetson Hall	231	781-555-1234	
5	147243	1D000005	Michael Roman	Liberty Hall	340	781-555-1234	
6	147244	1D000006	Judy Kerschner	Stetson Hall	23	781-555-1234	
7	147245	1D000007	Michael Kerschner	Langdon Hall	472	781-555-1234	
8	147246	1D000008	Lisa Bernstein	Jacobs Hall	567	781-555-1234	
9	147247	1D000009	Ricky Bernstein	Liberty Hall	45	781-555-1234	
10	147248	1D000010	Joshua Bernstein	Liberty Hall	200	781-555-1234	
11	147249	1D000011	Jayden Bernstein	Stetson Hall	140	781-555-1234	
12	147250	1D000012	Elaine Guthertz	Jacobs Hall	245	781-555-1234	
13	147251	1D000013	Ted Guthertz	Jacobs Hall	567	781-555-1234	
14	147252	1D000014	Jeffrey Guthertz	Stetson Hall	346	781-555-1234	Restricted from being admitted as a guest until further notice
15	147253	1D000015	David Guthertz	Stetson Hall	82	781-555-1234	
16	147254	1D000016	Shirley Bernstein	Liberty Hall	90	781-555-1234	
17	147255	1D000017	Jane Bernstein	Langdon Hall	82	781-555-1234	
18	147256	1D000018	Sally Bogg	Stetson Hall	56	781-555-1234	
19	147257	1D000019	Steven Bogg	Liberty Hall	378	781-555-1234	
20	147258	1D000020	Neal Levine	Liberty Hall	84	781-555-1234	
21	147259	1D000021	Neil Silverstone	Jacobs Hall	34	781-555-1234	
22	147260	1D000022	Alan Silverstone	Langdon Hall	580	781-555-1234	
23	147261	1D000023	Renee Silverstone	Liberty Hall	34	781-555-1234	
24	147262	1D000024	Lee Silverstone	Liberty Hall	20	781-555-1234	
25	147263	1D000025	Mark Silverstone	Liberty Hall	127	781-555-1234	
26	AADGLDB	1D000026	Greg House	Stetson Hall	34	781-555-1234	
27	B413571	1D000027	Lisa Cuddy	Jacobs Hall	678	781-555-1234	
28	B600649	1D000028	Thomas Wilson	Liberty Hall	23	781-341-3611	
29							
30							
31							

(example of the studentdb.xls file)

The sample student database supplied and installed when the program is installed may be referred to as a template for the actual student database which must be supplied by the institution.

DataFetch Encryption Utility

The latest versions of the DataFetch application programs contain an encryption facility, which can be used to keep the student database file secure, even when stored on remote workstations. The encryption algorithm, which is proprietary to DataFetch, is accomplished on a 'cell by cell' basis, rather than encrypting the entire file. The encryption algorithm is more than sufficiently secure to prevent the contents from being decrypted by all but the most gifted hackers. Universities may take additional measures to keep the data secure, if they wish, as long as the DataFetch applications have unencumbered access to the files.

