



FFI-MT, FFI-Lite and FFI Which is right for you?

There are three different FFI desktop applications and that causes confusion for potential users deciding which application to use. This handout will help select which FFI application is right for a particular user or group of users.

The Basics

The user interface, data management and reporting capabilities in all FFI applications are the same. The only difference is FFI-MT and FFI allow simultaneous user access to a FFI database. FFI data can be transferred/converted between the different FFI applications as a monitoring program changes size and scope. Individual administrative units, project units, macro plots and sample events can also be moved (imported and exported) between the different FFI applications. Selecting one of the FFI applications now does not limit moving the data to another FFI application later or limit the ability to share data with other FFI users.

Government users should check to see which FFI application is approved for use in their agency. FFI-MT users must also be able to install SQL Server or SQL Server Express.

Application	Monitoring Program/Database Size	Database Type	Allows Database Access for Multiple Simultaneous Users	Requires Installation and Management of SQL Server/Express
FFI-MT ¹	Medium to Large	SQL Server or SQL Server Express	Yes	Yes
FFI-Lite ²	Small to Medium	SQL Compact Edition	No	No
FFI ³	Medium to Large	SQL Server or SQL Server Express	Yes	Yes

¹NPS is required to access FFI-MT on the EROS Remote Application.

²As of this writing FFI-Lite is approved for installation on computers managed by FS, NPS, BLM, FWS and BIA.

³Support ended with version 1.05.13.81. Replaced by FFI-MT.

More information

FFI-MT (Many Tables) is recommended for new users who need simultaneous multiple user access to their FFI data. FFI-MT stores the data for each method in their own table. So, all tree data are in one table, all density data are in one table and so forth. This database design is not size efficient but, because all method data are stored together, creating views, queries and reports is faster than in FFI, especially, as the database size increases. Data are stored in a SQL Server or SQL Server Express database (SQL Server Express is a free version of SQL Server with a database size limit of 10GB). The user must install and maintain the instance of SQL Server or SQL Server Express. The SQL instance may be

installed on a network server, allowing server-client setup where only FFI is installed on the client computer(s). FFI-MT has been tested with SQL Server Express versions 2012-2019.

FFI-Lite was first developed as a “light-weight” FFI application for field data collection, but now has all the functionality of FFI-MT and FFI. It can still be used for field data collection and data can be imported into FFI, FFI-MT or FFI-Lite. FFI-Lite can be used as a stand-alone application for small- to medium-sized programs. FFI-Lite uses the same general database schema as FFI-MT, but data are stored in a SQL-CE database rather than SQL Server or SQL Server Express. SQL-CE is native in Windows, so it doesn’t require a separate software installation by IT staff. FFI-Lite begins to run slow when database size increases (~500 plots and sample visits, combined).

FFI is the original application and has been replaced by FFI-MT. The application stores attribute data in one large table that includes the attribute value, a Globally Unique Identifier (GUID) to identify the data field the data belongs to, and GUIDs for the administrative unit, macro plot and sample event. This data schema is size efficient because the attribute data are all in the same table, however attribute values for a specific record can be scattered throughout the table. For example, a tree’s DBH and height may be 1000s of rows apart in the data table even though, intuitively, they should be close together because they are near each other on the data entry screens. The benefit of this database design is reduced database size. The drawback is creating a view, query or report requires searching each row in the data table, which can be slow when the database gets very large.