
DataFax 4.0 Implementation Details

Martin Renters
Clinical DataFax Systems

Design Goals

- Try to stay database independent (don't use any fancy features)
- Allow ODBC capable tools access to data (use Excel/Access/Crystal Reports, etc. to view data)
- Let database enforce permissions (needed for ODBC access)

Changes

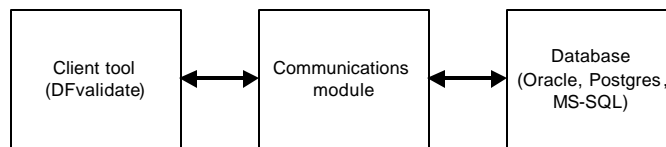
- Most user tools will be similar to existing 3.5 tools
- No more secondary records – this information is in audit trail
- Error records – images still kept in database but without keys

Limits

- Many limits have been increased
 - 65,000+ centers, plates
 - 65,000+ workflow levels
 - 65,000+ status levels
- Images stored in native resolution

Architecture

- Three components: client tool, communications module and database backend



Architecture Advantages

- Allows compression for slow links (like DFlite does)
- Allows encryption for Internet-wide connections
- No huge client DLLs need to be installed
- Client code becomes smaller (doesn't need database drivers for all databases)

Conventions

- All DataFax owned tables start with 'DF'
- All DataFax variables start with 'DF'


Functional Areas

- Setup Information (plates, modules, variables, etc.)
- Coding Information (status, workflow, etc.)
- Centers/Members Information
- Study Data
- Images
- QC Data
- Audit Data
- Workflow Data

Setup Information

- Plates are now made up of one or more Data Modules
- Modules contain the variables and may be referenced on multiple plates
- A plate may contain multiple instances of the same module (for repeating data)
- Each module is associated with an SQL table

Setup Information (example)



DataFax #007 Plate 016

Patient No. Patient Initials

F M L

Instructions:

- Ask patient to return used study medication blister packs.
- Put an **X** in the location of each **remaining** pill.
- If patient forgot to return blister packs, ask patient to complete the record from memory.

WEEK 1

month day year

DAY 1 DATE

Reporting method: Direct Observation Patient Report

Mark this box if all pills were taken and proceed to week 2

	Morning	Evening
DAY 1:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 2:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 3:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 4:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 5:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 6:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 7:	<input type="checkbox"/>	<input type="checkbox"/>

WEEK 2

month day year

DAY 1 DATE

Reporting method: Direct Observation Patient Report

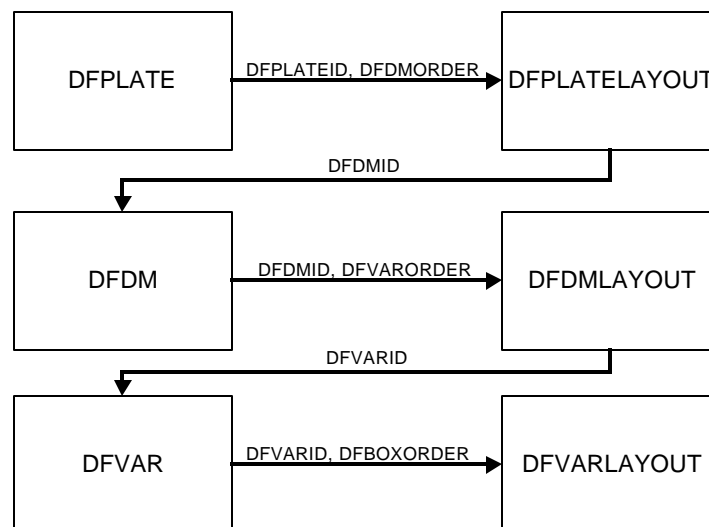
Mark this box if all pills were taken and proceed to week 3

	Morning	Evening
DAY 1:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 2:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 3:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 4:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 5:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 6:	<input type="checkbox"/>	<input type="checkbox"/>
DAY 7:	<input type="checkbox"/>	<input type="checkbox"/>

Setup Information (con't)

- Each module and variable is given a unique identifier
- Setup related tables are:
 - DFPLATE, DFPLATELAYOUT
 - DFDM, DFDMLAYOUT
 - DFVAR, DFVARLAYOUT

Setup Information (Linkage)



Setup Information (Plates)

- **DFPLATE** – definitions of plates
Plate number, Plate name, enter/exit EC, high/low resolution image ids, etc.
- **DFPLATELAYOUT** – layout of data modules on plates
Module ID, repetition number, name, X/Y position on page

Setup Information (Modules)

- **DFDM** - Data Module definitions
Name, Description, enter/exit EC, SQL table name
- **DFDMLAYOUT** – layout of variables in a module
Variable ID, order, X/Y position relative to module position

Setup Information (Variables)

- **DFVAR – variable definitions**
Name, type, style, required, legal value ranges, format, enter/exit EC, source type, note (for use, history, meaning)
- **DFVARLAYOUT – layout of boxes for variables**
X/Y positions of boxes relative to variable, size of boxes

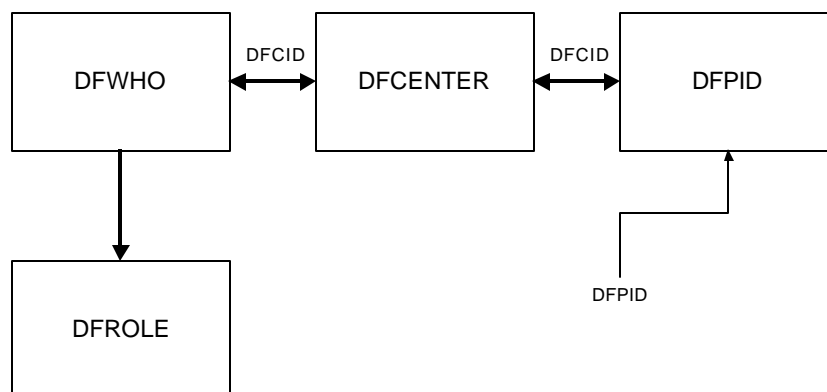
Coding Tables

- Contains various bits of information such as label associated with status, workflow (validation) levels, QC types, reason codes, etc.

Centers/Members Information

- Replaces DFcenters.db
- Table for information about each center (name, timezone)
- Table for information about people at the center (names, telephone, address, email, role)
- Table mapping patient IDs to center (one to one)

Centers (Linkage)



Study Data (Keys)

- A keys table (DFKEY) contains:
 - Record serial number
 - Patient ID
 - Visit, Visit repetition
 - Plate, Plate repetition
 - Status, reason codes
 - Data source id (image id)
 - Creation, modification timestamps
 - User who last modified the record

Study Data (Data Tables)

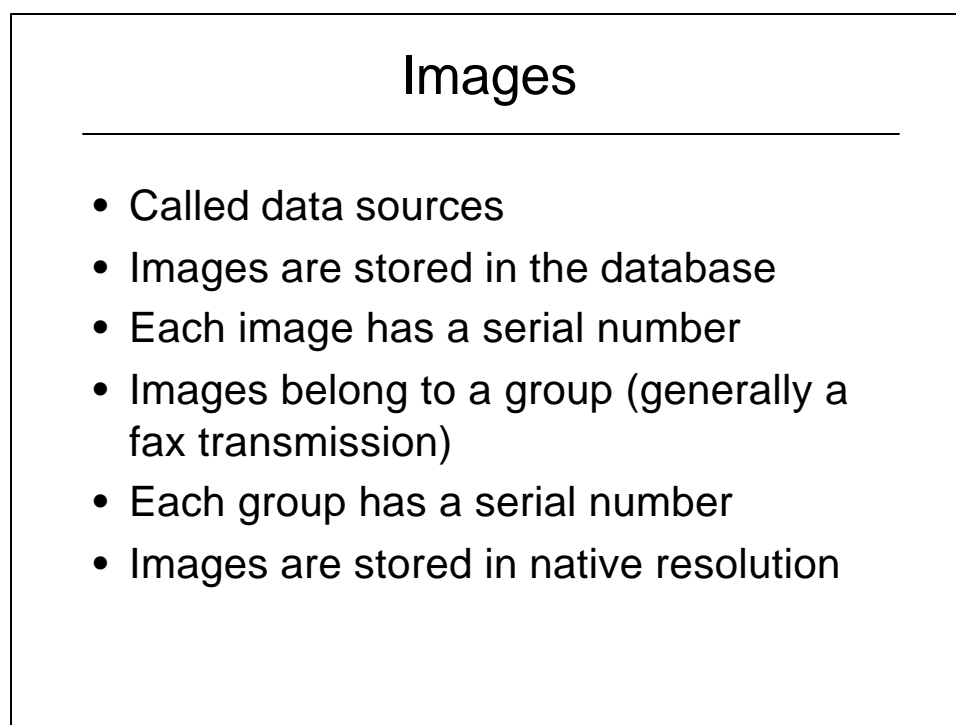
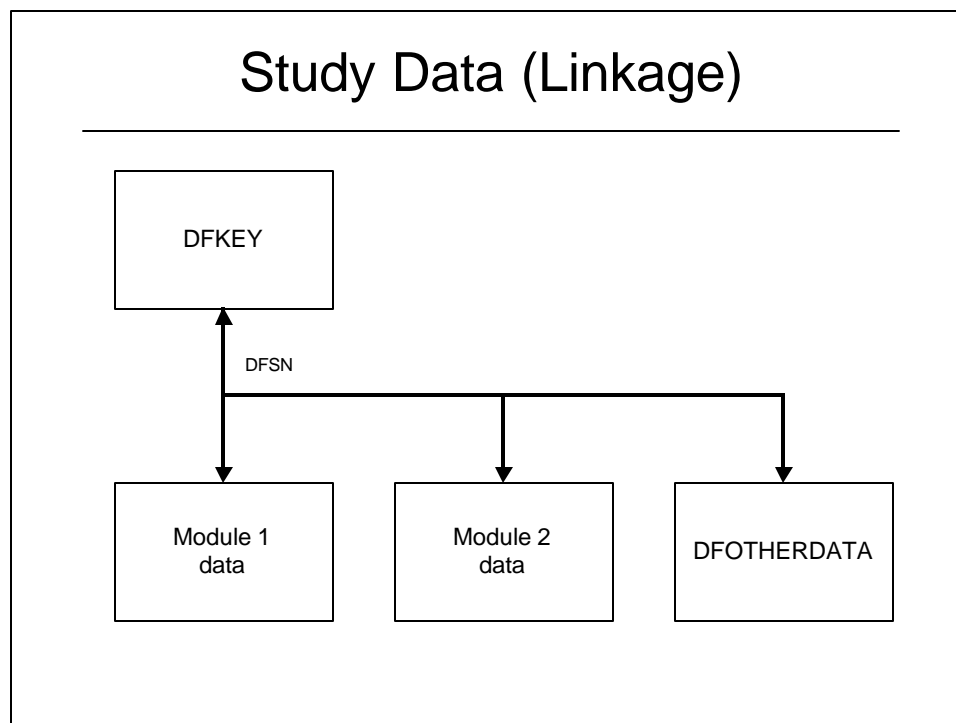
- Data fields are stored in modules
- Each module is an SQL table
- Each module has fields:
 - Record serial number (used to link with keys)
 - Module repetition number (for modules that repeat on a plate)
 - User defined data
- It is possible that the data on a page is spread out over multiple SQL tables

Study Data (Other info)

- SQL has one type of missing data – the NULL value – DataFax wants more types
- Partial Dates (2001/02/00) cannot be stored in SQL date fields
- While we were at it, how about a reason as to why a field has a particular value and investigator comments

Study Data (Other data con't)

- **DFOTHERDATA**
 - Record serial number
 - Module ID, repetition number
 - Variable ID
 - Missing Value code
 - Partial Date
 - Reason
 - Investigator comments

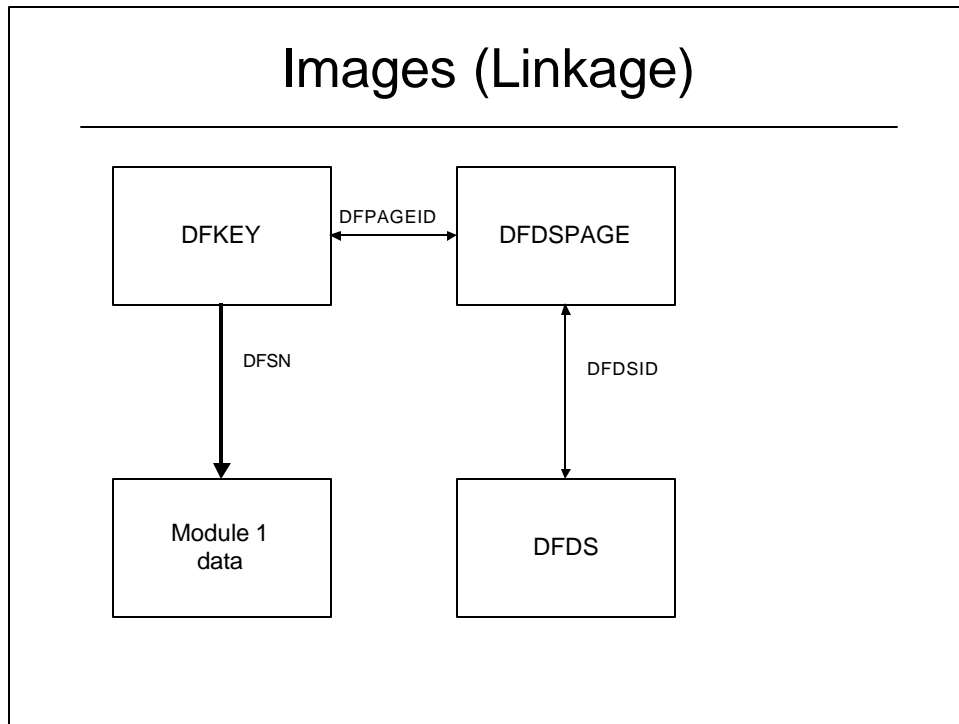


Images (DFDS)

- Data source table
- Hold information about a transmission
 - Data Source serial number
 - Type of data source (fax, email, raw data)
 - Arrival timestamp
 - Submitter
 - Number of pages

Images (DFDSPAGE)

- Holds individual page information
 - Page serial number
 - Data Source group serial number
 - Image status (new/error/active)
 - Image data (TIFF)



QC Data

- A QC may be attached to more than one field (related fields)
- A Field may have more than one QC on it
- QC reports table links QCs to the reports they are in
 - Allows us to see what has happened to QCs in a particular report

QC Data (DFQC)

- QC serial number
- Status ID
- Problem type ID (used once per field)
- Query
- Note
- flags (Q&A/faxback, Internal/External)
- username and timestamps (creation, modification, resolution, sending)

QC Data (DFQCREF)

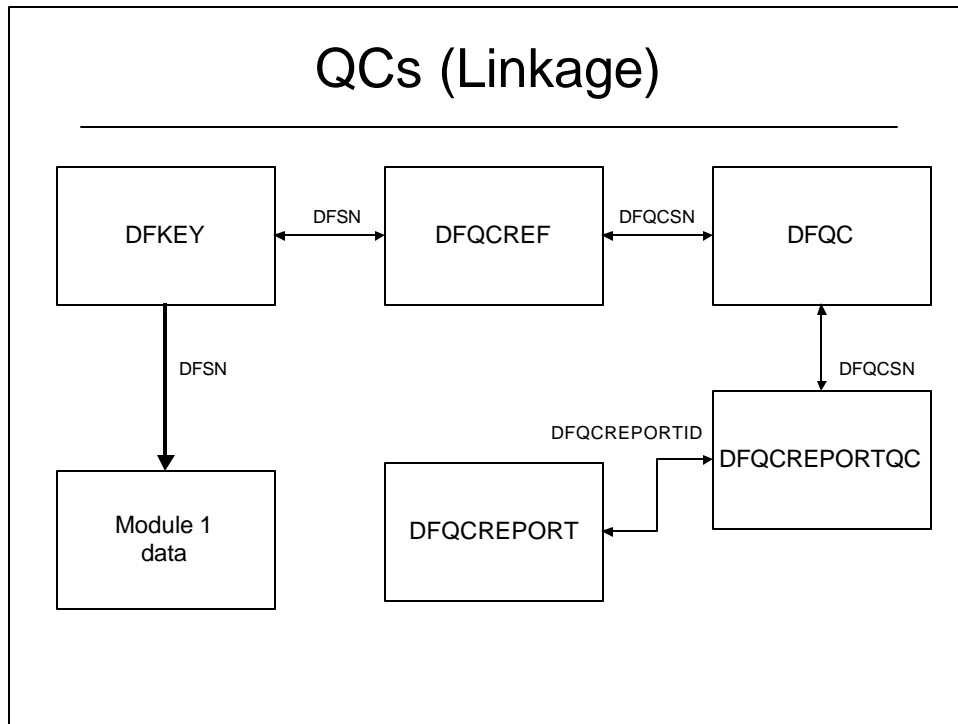
- links QCs to variables on data records
 - Record serial number
 - Data Module ID, repetition
 - Variable ID
 - QC serial number
 - Owner flag

QC Data (DFQCREPORT)

- Holds information about QC reports
 - Report ID
 - Creation date
 - Center number
 - Destination address
 - Date sent
 - Who sent report

QC Data (DFQCREPORTQC)

- Maps QCs into reports
 - Report ID
 - QC serial number



Audit Data

- Audit records are automatically generated by the underlying database system via triggers
- This ensures that modifications made outside of DataFax (via ODBC perhaps) live by the same rules as DataFax tools
- Indexed by record serial number for quick retrieval!

Audit Data (con't)

- **DFAUDITREC**

Record serial number (same as DFKEY)

Module ID, repetition

Variable ID

Type of change (Insert, Update, Delete)

Old, New data value

User who made change

Timestamp