DATAFax 2014.1 Study Setup

DataFax study setup involves several steps, some of which can be easily overlooked in the rush to meet start-up deadlines. Taking an organized approach to setup saves time and plays an important role in the study’s success.

Following is an outline of topics covered in this training. Additional information can be found in the DFsetup User Guide by going to the Help menu.

Preparing for Study Setup

• What is Required? (page 2)
• Study Configuration and User Permissions: DFsystem (page 3)

Study Setup using DFsetup

• Logging into DFsetup (page 4)
• Global Settings (page 7)
• Importing CRFs (page 11)
• Plate Definition (page 13)
• Importing Definitions (page 15)
• DataFax Data Types (page 16)
• Styles (page 20)
• Saving Your Work: Critical Errors (page 22)
• Defining Data Fields: Field Properties (page 29)
• Defining Data Fields: Type-specific Properties (page 30)
• DataFax Meta-Words (page 34)
• Centers Database (page 35)
• Participant Visit Scheduling: DataFax Visit Maps (page 37)
• Conditional Maps (page 46)
• DataFax Page Map (page 51)
• Missing Value Codes (page 53)
• Edit Checks Dialog (page 54)
• Lookup Tables (page 56)
• Modifying Plates after Study Startup (page 58)
1.0 What is Required?

Following is a description of requirements that must be met before beginning DataFax study setup.

1.1 Bar-coded Case Report Forms (CRFs)

A complete set of bar-coded CRFs is needed to create the background images for the database before setup work can begin.

The barcode must include the following:

- DataFax study number
- Plate number
- Visit or sequence number (optional; may be located in the barcode or in the first data field on the page)

*Note: For electronic data capture [EDC] studies, barcodes are not necessary. These pages will be assigned a plate number using the Import CRFs dialog (see Importing CRFs (page 11)).*

1.2 Visit Map

The DataFax visit map identifies (1) a schedule of visits for the study and (2) the CRF pages to be collected at each visit. The visit map is used to generate missing page and overdue visit queries.

Information used to create the visit map can often be found in the visit scheduling section of the study protocol.

The visit map must at least be visualized (e.g., as part of a Data Management Plan) before the CRFs are final, as this will help to determine the visit code structure on the CRFs.

1.3 Variable Definitions

Variable definitions include information such as field names and descriptions, coding, formatting, and legal ranges, among other things [see Defining Data Fields: Field Properties (page 29)]. This information should be at least somewhat specified before using DFsetup. One way to do this is by writing plate specifications as part of a Data Management Plan.
2.0 Study Configuration and User Permissions: DFsystem

Each new study should be created by the DataFax Administrator in DFsystem early in the study-planning phase. This ensures that a unique DataFax study number has been reserved and cannot be assigned to any other study.

User permissions and roles are also assigned through DFsystem. You must create at least one study role with permission to use DFsetup before being allowed to do anything in the setup tool.
3.0 Logging into DFsetup

1. Double click the **DFsetup** icon on your desktop. This opens the DFsetup login screen.

2. Enter your login name and password, then click Login. You’ll see a list of studies to which you have been given access.

3. Double click the appropriate study. You will then be asked to choose an access mode.

3.1 Access Modes

**DFsetup** provides five different access modes (your study role may not include permission to use all of the access modes). Try to select the lowest mode needed to accomplish the task at hand, to avoid blocking other users any more than necessary.

Following is a description of each access mode:

<table>
<thead>
<tr>
<th>Developer Access</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use:</strong></td>
</tr>
<tr>
<td>• Create a development-production link between two study databases.</td>
</tr>
<tr>
<td>• Publish setup changes from the development study to the production study.</td>
</tr>
<tr>
<td>• Review differences between the development and production study setups.</td>
</tr>
<tr>
<td>• Revert the development setup back to the current production version.</td>
</tr>
<tr>
<td>• Break the link between development and production study databases.</td>
</tr>
</tbody>
</table>
Permissions:
- Can be used by DataFax and Study Administrators only.
- User must have “Setup - Plates” permission.

Restrictions:
- Only one user can enter Developer mode at a time.
- Mode will be unavailable if any user is in Developer, Exclusive, or Normal access mode.
- No iDataFax, DFimport.rpc, or DFbatch sessions can be active.

<table>
<thead>
<tr>
<th>Exclusive Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
</tr>
<tr>
<td>- Add new plates and data fields to study setup.</td>
</tr>
<tr>
<td>- Revise data field layout on existing plates by adding, inserting, deleting, or reordering data fields.</td>
</tr>
</tbody>
</table>

Permissions:
- User must have “Setup - Plates” permission.

Restrictions:
- Only one user can enter Exclusive mode at a time.
- Mode will be unavailable if any user is in Developer, Exclusive, or Normal access mode.
- No iDataFax, DFimport.rpc, or DFbatch sessions can be active.

<table>
<thead>
<tr>
<th>Normal Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
</tr>
<tr>
<td>- Create and modify styles.</td>
</tr>
<tr>
<td>- Modify existing field properties.</td>
</tr>
<tr>
<td>- Add and test edit checks.</td>
</tr>
</tbody>
</table>

Permissions:
- User must have “Setup - Plates” permission.

Restrictions:
- Only one user can enter Normal mode at a time.
- Mode will be unavailable if any user is in Developer, Exclusive, or Normal access mode.
- Normal mode may be used while there are active iDataFax sessions. Any saved setup changes only become active for new iDataFax logins.

<table>
<thead>
<tr>
<th>Configuration Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
</tr>
<tr>
<td>- Create and modify the configuration files (centers, visit map, etc.) available under the DFsetup View menu.</td>
</tr>
</tbody>
</table>
Permissions:
- Separate permission is granted for each configuration file in the study roles.

Restrictions:
- Mode can be used by more than one user at a time, but each configuration file can be opened by only one user at a time.
- Mode can be used while another user is in Exclusive or Normal access mode.
- Mode will be unavailable if any user is in Developer access mode.

**View Only**

Use:
- Open and view all study setup components but not make changes.

Permissions:
- User must have “Setup - View” permission.

Restrictions:
- None.
4.0 Global Settings

Global settings can be changed at any time, but should be one of the first things you update in DFsetup when defining a new study.

Different global settings can be specified for each study. To change global settings, go to the Study menu and select Global Settings.

Following is a description of the different Global Settings tabs:

4.1 Global

- Study launch year: Used by some generic DataFax reports and can be used in study-specific reports.

- Reason for change: Users may be required to enter reasons for data change at the field level (per field), for all fields (always), or (never) [for more information on reason for change, see Reason Level (page 26)].

- New participant binders: Choose how many empty binders will appear for each site.

- Start New Patient Dialog: Allows the user to create a new participant binder from Data view. If no new participant binders are displayed and the Start New Patient dialog is disabled, the only way a new participant can be started is in Fax view.

- Run edit checks in view mode: Allows edit checks to run even in view mode (for example, display a message, run DFopen_patient_binder, or update a log to show who looked at a record).
4.2 Help

- This is an open space where users can define study-level help in a richtext format. In iDataFax, this content will appear by navigating to Help > Study Help.

4.3 Fields

- Field Descriptions: May be up to 40 characters long or limited to 25 characters (for older versions of SAS and to create more compact QC reports).
• Start year (for two-digit years) and imputation choices: Can be set at the study level here and also tweaked on a field-by-field basis.

• Auto-generation: Can be used to quickly define unique and generic variable names and help messages.

• Field Names: Marking this box indicates that field names must be unique across modules for the entire study. For example, a field can be named "BP systolic" in the Vital Signs module and then used however many times that module is applied throughout the study, but a field in the Physical Exam module cannot be named "BP systolic." This setting is case-sensitive, so there could be two different fields named "BP systolic" and "bpsystolic."

4.4 Guides

• Enable Snap: Data field widgets snap into place when dragged near the boxes defining a data field on the imported CRF page.

• Display CRF: Makes the imported CRF visible.

• Display Fields: Makes any data fields defined on a CRF page visible.

• Maximum Box Height: When creating a new data field by dragging out a widget from a lower corner to an upper corner, the height is always constrained to the recommended maximum (25 pixels, about 1/4 inch), but a different constraint can be specified for dragging in a downward direction.
4.5 Color

You can customize the colors in DFsetup.

- Fields: Widgets showing size and location of data fields.
- Foreground: Field number printed inside field widgets.
5.0 Importing CRFs

After adjusting global settings, the next step is to import the CRF background images.

1. In DFsetup, go to the Study menu and choose Import CRFs.

2. In the window that appears, click on the (...) button and locate the file that contains your CRFs. Select your entry by clicking on it. The file name will appear in the Selection window near the bottom of the Import dialog.

3. Check the box for “Do not import pages with invalid or missing barcodes.”
4. Click **Import** and the Import CRFs window will appear:

![Import CRFs window](image)

The left side of the window, under CRF Pages, shows the pages listed by plate number, with the following possible actions:

- **Create**: Page is new.
- **Discard**: Page has no barcode.
- **Replace**: Page has already been defined.

**Note**: For EDC studies, barcodes are not necessary. These pages will be labeled “Discard” on import but can be changed to “Create” (by clicking “Identify CRF”) and assigned individual plate numbers.

5. When the CRF Page specifications are complete, click the Save Pages button to import the CRFs.
6.0 Plate Definition

After the import step has been completed, each plate must be defined by going to the View menu and choosing Plates. The following window will appear:

![Plate Definition Window](image)

The following items should be defined for each plate:

<table>
<thead>
<tr>
<th>Plate Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usually in the format “Acronym: CRF Name” (e.g., “DEM: Demographics”). The label is used by DFsetup during plate navigation and plate import operations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence is in Barcode or First Data Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the barcode includes a sequence/visit number, the “Sequence is” option should be set to Barcode.</td>
</tr>
<tr>
<td>If the sequence/visit number is not bar-coded, it must appear as the first data field (field #6) and the plate definition should indicate First Data Field.</td>
</tr>
<tr>
<td><strong>Note:</strong> Incorrectly setting this option for a plate may result in incorrect numbering of the data fields on the plate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ICR Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>The intelligent character recognition (ICR) software will attempt to read and enter choice and check boxes, number fields, three-character months (in dates that use this format), plus and minus signs (in signed numeric fields), and visual analog scales. ICR will not process text fields.</td>
</tr>
<tr>
<td>In most cases ICR is helpful to data entry clerks; however, it can be turned off for any or all plates. There are three options:</td>
</tr>
<tr>
<td>• <strong>None:</strong> Do not perform ICR.</td>
</tr>
<tr>
<td>• <strong>Standard:</strong> Perform ICR on new faxed pages but not on refaxed pages.</td>
</tr>
<tr>
<td>• <strong>Merge:</strong> Perform ICR on new faxed pages. For refaxes, keep previously entered data and perform ICR for any blank fields (except for blank fields to which a reason has been applied).</td>
</tr>
</tbody>
</table>
### Plate Triggers Early Termination

Should be set to **Yes** for any plate that may trigger the termination of a subject's participation. Such plates might include a death report, an early termination form, or a lost to follow-up form.

**Note:** A scheduled Termination visit (visit type T) should not be defined as triggering early termination. A Termination visit is considered required by DataFax while a plate signaling early termination is not.

### Eligible for Signing when

Can be set to **Final** or **Final or Incomplete** for any plate that may have e-signature fields.

### Plate Arrival Trigger

Sometimes it is useful to have an event occur automatically whenever a particular plate arrives in the database. For example, a trigger might be used to send someone an email message when a Serious Adverse Event form is received.

Plate arrival triggers execute when:

- a bar-coded CRF page arrives, provided the barcode can be read;
- a page is identified and sent to the study using the iDataFax Fax router function; and
- data records are imported to the new record queue using DFimport.rpc.

**Note:** If a page cannot be identified on arrival and thus ends up in the unidentified fax router, the plate arrival trigger will be delayed until the page has been identified using iDataFax.

Edit check programs can also perform the same tasks (and more).
7.0 Importing Definitions

Before defining new data fields, it is a good idea to consider whether it is possible to reuse work from another study. If a plate already exists in another study, the plate, style, and module definitions can be imported from that study and modified as needed [for more on styles see Styles (page 20)].

To import definitions from another study:

1. Go to the Study menu and choose Import Definitions.
2. In the window that appears, select the study from which you wish to import. Click OK.
3. Select the desired plate(s).

- When a plate is selected, the styles and modules used on that plate are automatically selected as well.
- When a style or module is deselected, the plate that uses that style or module is automatically deselected as well, unless that style or module already exists in the current study.

4. Use “Import As” to import to a different plate number in the current study.
8.0 DataFax Data Types

Before beginning to define data fields, it is helpful to understand the types of variables that DataFax supports.

8.1 Number

Number fields can be used to store whole numbers, decimals, and signed numbers (i.e., +/-). Signed numbers are described separately below.

Display/Store Length: The display and store length represent the maximum number of digits allowed in the field itself. For decimals, display and store length should include the decimal point.

- For example, to capture a numeric value of 35.5, the display and store length would both equal 4.

Format: The format is defined with an “n” for each digit; the decimal point is also expressed in the field’s format.

- For example, to capture a numeric value of 35.5, the format would be defined as nn.n.

8.1.1 Signed Numbers

Display/Store Length: For signed numbers, display and store length should include the +/- sign.

Format: The format for signed numbers should include an “S” (required) or “s” (optional).

- Snnn represents a 3-digit number with a required sign (+/-) as the leading character.
- snnn represents a 3-digit signed number or a 4-digit unsigned number (the leading +/- is optional).

Examples:

- To capture a value of -15.0 (sign is required), the format would be defined as Snn.n, and the display and store length would both equal 4.
- To capture a value of -25 to 100 (value may be signed or unsigned), the format would be defined as snnn, and the display and store length would both equal 4.

8.2 Date

Date fields include the following components, defined in the format field:

- Day:

  - **DD or dd** – A two-digit number representing the day of the month. Upper case (DD) means the day of the month is required, while lower case (dd) means that a partial date, with unknown day of month, is allowed (zeros must be entered in iDataFax to indicate unknown).
**Note:** You can also allow or restrict partial dates through the use of the impute feature [see Impute (page 32)].

- **Month:**
  - **MM or mm** – A two-digit number representing the month. Upper case (MM) means the month is required, while lower case (mm) means that a partial date, with unknown month, is allowed (zeros must be entered in **iDataFax** to indicate unknown).
  - **MMM or mmm** – A three-letter abbreviation representing the month. Upper case (MMM) means the month is required, while lower case (mmm) means that a partial date, with unknown month, is allowed (zeros must be entered in **iDataFax** to indicate unknown). Allowable three-letter abbreviations include the following: JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC.

- **Year:**
  - **yy** – A two-digit number representing the year, using an implied range specified as a consecutive 100-year period.
  - **yyyy** – A four-digit number representing the year.

Care must be taken when specifying a combination of upper and lower case date components. For example, it is not logical to have a date for which the day is required but the month is optional, i.e., DD/mm/yy.

**DELIMITERS**

Separation characters (or delimiters) can also be included in the format, but they are not mandatory. When included, separation characters appear in the database, are displayed in **iDataFax**, and are printed on Quality Control reports, without having to be explicitly entered on the study CRF or typed in by a data entry clerk.

Any character can be used to separate the date components, including no delimiter at all. If delimiters are present in the date format, they must also be specified in the date’s legal range. The following are valid date formats with various delimiters:

- dd/mm/yy
- MMM DD, yyyy
- yyyy:mm.dd

### 8.3 String

String fields are used to capture alphabetic (letters) or alphanumeric (letters and numbers) text. Each string field is defined with a maximum display and store length. Setting the number of characters to be stored in the database as larger than the number displayed on the screen is quite common.

**DataFax** also allows some limited formatting of string fields. For example, all input can be forced to upper or lower case.
Note: The | (pipe) character cannot be used in text strings because this special character is used as the delimiter between data fields in the study database. Consequently, DataFax blocks any attempt to enter this character into string fields.

8.4 Choice

Choice fields are used to capture responses to “mark only one” questions. For example, a yes/no variable would be defined as a choice field (that is, only one response is allowed). Each possible response has a unique code and an optional text label (e.g., yes = 1, no = 2).

There is a built-in option for when no response is marked, called “No Box.” The default code for this option is 0 and the text label is blank, but any code and/or label can be assigned.

Choice fields are limited to a maximum of 20 response options.

8.5 Check

A check field is a single box that is either marked or left blank (e.g., a “not done” box at the top of a page). Check fields can also be used to capture data for multiple choice questions where there is an instruction to “mark all that apply.” In this case, each possible response is a single check variable. Edit checks can be used to ensure that at least one response is marked.

By default, the code 1 is used to represent a checked box, while the code 0 (zero) represents a blank box. As with choice fields, these codes can be changed.

8.6 VAS (Visual Analog Scale)

A VAS is a horizontal scale with fixed end points. Optional anchors may be added anywhere along the scale to mark other fixed points.

Precision and a minimum and maximum value can be specified for VAS. The default minimum and maximum values are 0 and 100.

Horizontal VAS are created by dragging out a rectangular box the same length as the VAS scale and positioning it on top of the scale line.

iDataFax also supports vertical VAS fields, but these cannot be read by ICR. Vertical VAS are created by dragging out a box at each end of the scale. The boxes must be wide enough to display a numeric value and must be positioned so that the top edge of the upper box corresponds to the top of the scale and the bottom edge of the lower box corresponds to the bottom of the scale.
8.7 Time

A field consisting of an hour and minutes, or an hour, minutes and seconds.

8.8 Hidden Fields

Hidden fields are not an additional data type, but it is helpful to explain them here. Hidden fields appear in the database but not on the CRF. They are created by dragging out a widget anywhere in the CRF window and defining it like any other data field. Such fields are sometimes used for coding or other central office purposes. Study roles (defined in DFsystem) are used to determine whether a user sees hidden fields in iDataFax. Fields can also be masked using this setting. A masked field will appear to all DataFax users as a shaded field that does not allow data entry.
9.0 Styles

Every data field has a style. The style specifies the data type (number, string, date, check, choice, or VAS) and other properties. These properties can be locked at the style level, so they cannot be modified at the field level, or they can be left unlocked, so that they can be modified at the field level.

The use of styles simplifies field definition and helps to ensure consistency across the study. Styles also make it faster and easier to edit field definitions. For example, if the blood pressure legal range changes, and the range has been specified at the style level, you can update the range for all blood pressure variables by editing the style.

Styles can be accessed in **DFsetup** by going to the View menu and selecting Styles. While styles may be modified, added, or deleted at any time during setup, it is helpful to define as many styles as you can prior to beginning setup. DataFax will warn you if you try to delete a style that is in use by an existing field.

9.1 DataFax Styles

DataFax has the following built-in styles:

- SimpleNumber
- SimpleDate
- SimpleDateYYYY
- SimpleString
- SimpleTime
- SimpleChoice
- SimpleCheck
- SimpleVAS
- VisitDate

The “Simple” styles correspond to the seven data types discussed in DataFax Data Types (page 16). The VisitDate style is discussed below.

You cannot delete any of the built-in styles; however, the date styles can be modified to correspond to the date format used in your study.
VisitDate Style

This style should be applied to the field (or fields) used for collecting the visit date at scheduled visits. DataFax quality control programs use VisitDate dates to determine whether scheduled visits have arrived or are overdue. Before applying this style, you may need to modify it to be consistent with the date format used in your study (i.e., the order of year, month, day).

Note: In iDataFax, the report DF_ICvisitdates can be used to check for consistency of VisitDate-style dates within a visit.

9.2 Creating User-Defined Styles

Study-specific styles can be created for field types that appear repeatedly in the CRFs. Ideally, styles should be defined before creating individual data fields, but they can also be created as the need arises. Just be sure to consider the required properties of all fields that will use the style so that you can decide which properties to lock in at the style level.

A style definition can be changed at any time. Properties that were previously defined at the field level can be over-ridden by specifying and locking a value in the style. When this is done, all fields that use the style will inherit the newly locked style property.

Styles can also be changed by unlocking a previously locked property. The fields that use the style will continue to hold the property’s previous value, but the user is now able to edit the unlocked property for some or all of these fields.

Note: Styles can be changed at any time, including after study data starts to arrive, but be careful not to change anything that would conflict with data you have already collected.

The following steps should be followed when defining your own styles:

1. Review all CRF plates to identify where the same or similar fields occur. Define styles to represent the constant attributes of these fields.

2. For each style, the style name and data type must be specified. All other style attributes are optional and should only be defined if they will remain constant across all fields that use the style.

3. In DFsetup, go to the View menu and select Styles to open the style definition dialog.

4. Click New to add a new style.

If you want to use the same styles for multiple studies, you may import all or a few of the styles from one study to another by going to the Study menu and choosing Import Definition.
10.0 Saving Your Work: Critical Errors

Once you begin to define data fields [as described in Defining Data Fields: Field Properties (page 29)], you should save your work frequently by going to the File menu and choosing Save. However, you must resolve any critical errors before you can save your work.

If there are critical errors, you will see the following window when you try to save:

If you click the OK button and close DFsetup, your work will be lost.

Once all critical errors are resolved, there may be non-critical errors in your setup. If there are non-critical errors, you will see the following window when you save:

Notice that there is a “Continue (Save)” button in the bottom left of the window. Choose this button and your work will be saved.

One way to avoid most critical errors is by defining the visit number and participant ID fields on all CRFs before defining the rest of the variables.
11.0 Defining Data Fields: Module definition

Now that styles are defined, you can begin defining data fields. This process starts in the Modules View, which can be accessed by navigating to View > Modules.

In the Modules window, create a new module by clicking “New” under Module options on the lower lefthand corner of the window. Name the module according to what kind of data fields will be created within it (e.g. VitalSigns, AElog). The name cannot exceed 15 characters and only letters, numbers, and underscores are allowed. Note that this means meta words such as $(plate) cannot be used when naming modules.

Once you have created a module, you can add fields to it by clicking “New” under Field Options while the module name is highlighted on the Module List. A pop-up window will immediately prompt you to select a field style and then name the data field.

After naming the field, you can begin to define this field's properties within the module. The alias will default to the field name you have just defined, but it can be altered if desired.
Field properties for each data field can be defined here within the Module View or at the plate level, when applying data fields to the CRF. As you place the data fields on the CRF, whatever properties you have defined for that field within the Module View will carry over, but can be changed at the CRF level once the field is placed. If you decide that some properties should not be altered as data fields are applied, you can lock properties at the module level by clicking the box under the lock symbol to the left of each property. If you wish to inherit a property from the settings defined in the data style selected, you can select the "inherit" box next to the appropriate property (the left-most check box to the left of each property). This prevents that property from being changed as the data field is applied on the field. Names are always locked at the module level and cannot be changed at the plate level.

11.1 Name

Each field is required to have a name. Names do not have to be unique across the study unless indicated in global settings by marking the box “Field names must be unique across all module definitions” (for more information on this setting see section 4.3, fields). These names are used in the edit check language and can also be used by DFSas when creating SAS data sets.

11.2 Alias

Aliases are similar to names but should be made unique across all plate definitions in the study. This is not enforced as a requirement in DFsetup, but a warning box will fire indicating that the defined alias is not unique. The alias can be up to 80 characters in length, cannot contain spaces, and can consist of letters, digits, underscores, and meta-words [see DataFax Meta-Words (page 34)]. Aliases are case sensitive, thus MED1, Med1 and med1 are all considered to be unique. DFSas can also be directed to use aliases when exporting data.
Most aliases are defined at the field level, not in the style and/or module level. However, you may wish to include the alias in the style or module for fields that appear only once on any given plate (e.g., participant ID and initials). You can do this by appending the meta-word for plate number, $\text{plate}$, to the alias. For example, \text{PID}$_0\text{plate}$ at the style level would result in alias \text{PID}001 on plate 1, \text{PID}002 on plate 2, etc.

### 11.3 Description

A description must be specified for each data field.

- Maximum length = 25 or 40 characters (as specified in Global Settings).
- Description appears on the quality control report, and should therefore match the text on the CRF as closely as possible. This allows the person responding to the QC report to quickly locate the item in question.
- Question/item numbering should be included in the description if numbering appears on the CRFs.
- If abbreviations are necessary, they should be common medical abbreviations.

### 11.4 Legal

Legal ranges should be specified whenever possible. While DataFax does not prevent the entry of values outside of the legal range, it does highlight the field as red if an illegal value is entered. If there are any illegal fields on a data record, the user will be unable to save the record with status “Final.”

**Note:**

- If a query or reason is attached to a data field, the query/reason color (blue, orange, or green) will take precedence over the illegal value color (red).
- A record with an illegal value can be saved as “Final” if there is a resolved query or reason on the field. For example, if a blood pressure value is outside of the legal range, the site would be queried to confirm or correct the value. If the site indicates that the value is correct, you would resolve the query. The value is still outside of the legal range, but the record can now be saved as “Final.”

### 11.5 Help Messages

An optional help message for each field can be displayed at the bottom of the iDataFax window.

- Help messages can include meta words. For example, the legal range can be displayed by using the help message “Legal range is: $\text{legal}$.”
- Help messages may consist of specific instructions for completing the data field (e.g., “If middle initial is unknown, enter subject’s initials as M-T”).
11.6 Need

There are three different settings for this property:

- **Optional**: Field may be completed or left blank. Data record can be marked “Final” if field is blank.
- **Required**: Field must be completed with a data value or a missing value code. Data record cannot be marked “Final” if field is blank (unless there is a resolved query or reason on the field).
- **Essential**: Field must be completed with a data value. Missing value code cannot be applied. Data record cannot be marked “Final” if field is blank.

Key fields (participant and visit numbers) are always essential, regardless of how the need property is specified.

11.7 Reason Level

The reason level (1-7) identifies the minimum validation level at which changing the data field requires a reason. For example, when the reason level is set to 3 a user can change the value without specifying a reason if the record is below level 3. A reason must be provided if the record is at level 3 or higher.

The reason property at the style or field level only applies if the global setting (see Global Settings) is set to “per field.” A global setting of “never” is equivalent to specifying “none” for all fields, and a global setting of “always” is equivalent to specifying level 1 for all fields.

11.8 Hidden

Data fields classified as hidden only appear in **iDataFax** if the user’s role (defined in **DFsystem**) allows him or her to see hidden fields. Permission to see hidden fields may differ by plate and visit. Data fields classified as masked will appear in **iDataFax** as shaded boxes that do not allow data entry.

Edit checks on hidden and masked fields that a user cannot see will not be triggered in **iDataFax**. While hidden fields are skipped and their edit checks ignored, they can be read and set by other edit checks both interactively and in batch.

11.9 Display, Store, and Adjust to Fit

The Display length determines the number of characters that will be displayed on one line of the data field in **iDataFax**. The Store length determines the number of characters that can be saved in the database.

Clicking the “Adjust to Fit” button changes the length of the data field to match the specified display length. This is particularly useful for text fields. The adjustment is made to the right edge of the field, with the restriction that field length cannot exceed the right edge of the data entry screen. If matching the display length would exceed this boundary, the field will increase to the right as far as possible and the display length will decrease to match the new length.
11.10 Skip

The skip property can be used to skip over one or more fields depending on the response to the current item. You must specify (1) the number of fields to be skipped over and (2) the data value or values that trigger the skip.

To indicate that more than one value can trigger the skip, enter a comma delimited list of values and/or ranges (e.g., 1-3,5,7).

The following meta-words can be used in skip specifications:

- **$(blank)**: Triggers the skip when the current field is blank.
  
  - Can be combined with other values, except $(legal). For example, if the current field is a choice field and you want to skip fields if none of the choices was selected (i.e., the field is blank) or if one of choices 1-3 was selected, the skip specification would be: $(blank),1-3.

- **!$(blank)**: Triggers the skip when the current field is not blank.

- **$(legal)**: Triggers the skip when the current field contains one of the values specified in the legal range property or a missing value code.
  
  - Cannot be combined with other values and only refers to the values specified in the legal range property. If the legal range is not specified the skip rule will never be met.

- **!$(legal)**: Triggers the skip when the current field does not contain one of the legal values or a missing value code.

The skip rule applies in **iDataFax** when moving forward out of the field using the tab or return keys. It does not apply when moving backward using shift-tab or shift-return, or when moving directly to another field using the mouse.

More complicated rules and/or different skip behavior can be implemented using edit checks.
11.11 Edit Checks

Edit checks are small, user-defined programs used to implement data consistency checks, lookup tables, computed values, and many other types of programmed behavior.

Edit check programs are written, checked, and published using the Edit Checks dialog (accessible by going to the View menu and choosing Edit Checks). The Edit Checks dialog is briefly described in Edit Checks Dialog (page 54). It is also discussed in detail in the DataFax User Guide.

Enter the name of one or more edit checks (and any parameters, if applicable) in one of the following four boxes depending on when you want it to be triggered:

- **Plate Enter:** When the plate first appears on screen. Any plate entry edit checks are executed, in field number order, before moving to the first field on the page.
- **Field Enter:** When the user moves into a data field. Edit checks are executed when moving into the field by keyboard or mouse.
- **Field Exit:** When the user leaves a data field. Edit checks are executed when moving out of the field by keyboard or mouse.
- **Plate Exit:** When the user saves the data record. Plate exit edit checks are executed, in field number order, before leaving the record.

**Note:**

- **Field enter and exit edit checks only fire if the user visits the field; they are not triggered on fields the user skips. This applies to manual skipping via the mouse and scroll bar, and to automated skipping via the Skip field property and edit check dfmoveto instructions.**
- **Edit checks attached to hidden fields are not executed in iDataFax if the user does not have permission to see the hidden fields.**

**Batch Mode**

Edit checks can also be executed in batch mode. Batch mode simulates a user tabbing through all data fields on all records and then saving the record. Edit checks on hidden fields always run in batch mode regardless of who is running the batch, and dfmoveto statements in edit checks are ignored during batch processing, thus all fields are traversed.
12.0 Defining Data Fields: Field Properties

Now that data field specifications have been defined within their respective modules, they can be applied onto the CRFs. From the CRF view, map the module to the appropriate plate by clicking "Add" under Module Options on the lower left corner of the screen while the appropriate CRF is highlighted. Then select the desired module from the list that appears.

Once the module is mapped to the CRF, individual fields can be created on the CRF itself. This is done by (1) clicking on the boxes that make up the field or (2) dragging a data entry widget over the desired location in the CRF window. Then, while the data entry widget is highlighted click on the appropriate field name under the module title on the left side of the screen. The data entry widget will be created with the specifications defined at the module level for that field. Once the variable has been created, it can be moved using the arrow keys or by dragging it with the mouse. Holding the shift key while using the arrow keys moves the data entry widget in large steps. To cancel the creation of a new data entry widget at this stage, click anywhere in the screen background.

Numbers are typically designed with a separate box for each digit. The individual boxes are necessary if you plan to fax the forms and want the ICR software to read them. A data entry widget is created by clicking each of the boxes from left to right.

**Note:** For number and date fields, the order is important and must always be left to right. For choice fields the order is equally important, even more so for choice fields that span several rows and columns.

After creating a data entry widget, you can then define any field-specific properties. For more information about Field Properties, see Section 11.0.
13.0 Defining Data Fields: Type-specific Properties

This section discusses field properties that are specific to certain data types. Date and VAS types are not discussed below as they were fully explained earlier in the manual [Date (page 16) and VAS (Visual Analog Scale) (page 18)].

13.1 Check and Choice Fields

Check fields consist of a single box while choice fields have two or more boxes, but otherwise they are the same; either the field has no response (no box checked) or a single box is checked. In both cases, a unique code and a text label are associated with each box and with the “no response” state. The code is stored in the study database while the label exists only in the study schema. The code must be an integer between 0-65535 (inclusive), and the text label has a maximum length of 32 characters.

**Check field coding:** A code and (optional) label are specified for each of the two possible states: (1) the box is checked or (2) the box is not checked.

**Choice field coding:** A code and label are specified for each choice box and for the case in which no box is checked. Box 1, Box 2, etc. correspond to the order in which the boxes were selected when creating the data field.

**Legal:**

- The meta-word $(choices)$ may be used in the legal specification for choice fields. This limits legal responses during data entry to the specified codes for that item, including the code for no choice. However, if need is specified as “Required” or “Essential” the field will be flagged as illegal if no choice is selected.

- It is possible to specify legal codes directly (e.g., 0-4 for the choice field and 0-1 for the check field in the examples above). But remember to include the no choice code as a legal choice for “Optional” fields, otherwise they too will be flagged as illegal when blank.

A few tips:

- Keep the case consistent for all labels.
- Throughout the study, “Yes/No” response codes should be consistent regardless of their order in the question.
- Use consistent codes to represent “blank” or “no choice” responses.
- For check fields, use a consistent code to represent “check off” and “check on.” The code chosen for “check off” should be the same as that chosen for “no choice” in the definition of a choice field.
13.2 Number Fields

13.2.1 Display and Store

For number fields the display and store length must be the same. You cannot store more digits than are displayed on screen. If a format is specified, the display and store length must equal the format length. DFsetup will display a warning and will not allow a field definition to be saved if a mismatch occurs.

13.2.2 Format

The number field format is comprised of:

- an “n” for each digit in the number
- a decimal or other delimiters
- fixed components (e.g., a leading digit)
- a leading “S” if a sign (+/-) is required
- a leading “s” if a sign (+/-) is optional

If a whole number or decimal format is specified (e.g., “nnn” or “nn.nn”), and the user does not enter all of the required digits, iDataFax will auto-complete the value on field exit by adding leading and/or trailing zeros as needed. For example, 22 is converted to 022 to match format nnn, and 2.2 is converted to 02.20 to match format nn.nn. This does not occur if no format is specified, or a format is specified that contains anything other than “n”s and a single decimal.

13.2.3 Single Rectangular Box

Most number fields are defined using a separate box for each digit, but it is also possible to define a number field consisting of a rectangular box containing pre-printed or hand-written digits. To define such fields click inside the box, select a numeric style, and set the store and display lengths equal to the number of digits the field will contain.

When users complete such fields on a paper CRF all of the specified digits must be included, otherwise the ICR will fail and leave the field blank. Users should be instructed to enter leading zeros, leave a clear gap between each digit, and print large without touching the top and bottom of the box.

For pre-printed numbers ICR accuracy is best for 18 pt Avant Garde-Book with a spread of 15-20% to clearly separate the individual digits. The rectangular box used for these fields should be 1/3 inch high and wide enough to hold all of the digits without crowding. Whether pre-printed or hand-written, digits should be positioned mid-way between the upper and lower edges of the box and clearly separated from each other.

These fields should not have a format statement. Delimiters and decimals are not allowed.
13.3 Dates

13.3.1 Range for Two-digit Years
For dates with a two-digit year, a year range must be specified. The range is represented by a consecutive 100-year period in which the two-digit year is interpreted as falling. The range can be specified in Global Settings and, if necessary, overwritten at the style and field levels.

Example: A legal range of 94/09/05 – 01/01/01, with an implied range of 1910-2009, is equivalent to 1994/09/05 – 2001/01/01.

Example: A date of 15/08/40 (format = dd/mm/yy) with an implied range of 1910-2009, is equivalent to 15/08/1940, not 15/08/2040.

13.3.2 Impute
This feature allows you to specify how partial dates are to be treated in legal range tests and data export. Options include:

- **Never**: No imputation is performed, partial dates are invalid.
- **Beginning**: Missing day is imputed as the first day of the month; missing day/month is imputed as the first day of the year.
- **Middle**: Missing day is imputed as the 15th of the month; missing day/month is imputed as July 1st.
- **End**: Missing day is imputed as the last day of the month; missing day/month is imputed as the last day of the year.

The formats chosen for a date field must correspond to the imputation chosen for that field and vice versa. For example, imputing to “beginning” would be required to have a format of dd/MM/yy (for first of month) or dd/mm/yy (for first of year).

Imputation can be specified in Global Settings and, if necessary, overwritten at the style, module, and field levels.

**Note:** If partial dates are allowed, the partial date must be entered in iDataFax with the incomplete component as “00” (or “000” for a 3-character month). Date rounding will only be displayed upon export of the data itself (e.g., when using DFexport.rpc or DFsas).
13.4 String Fields

13.4.1 Display and store
Because the width of computer screens is limited, the length of text fields displayed on screen is often set shorter than the maximum that can be entered and stored in the study database. Multiline text fields can be created by holding the Control key (or Command key on Macs) while dragging out a rectangular box for the field. However, the space available on many CRF pages is only sufficient for a one line text field. In such cases the height of the text field automatically expands into a multiline text box when the user enters the field, and returns to its display size when the user leaves the field.

13.4.2 Format
Specifying a format is rare but possible for string fields. In string fields, “^” is used as a place holder for characters to be entered by the user. All other parts of the format are considered fixed. For example, “Model #^^^” could be used for a string field that must start with “Model #” followed by up to 3 other characters.

If a format is specified, its length must match the display and store length, with one exception. If the format ends in a single “^” character, as in “Model: ^,” the display and store length are honored and text can be entered from the position of the “^” character to the end of the store length.

13.4.3 Mapping
Mapping can be used to automatically set all text to upper or lower case, regardless of how the user types.
14.0 DataFax Meta-Words

Meta-words are words reserved by DataFax for which substitution of a given value occurs. They are used to simplify the specification of field attributes, and are particularly useful when defining styles.

The following meta-words can be used in style and field property specifications as described below:

14.1 Meta-Words for Legal Range specifications

today: Inserts the current calendar date from the system clock of the DataFax server. This meta-word is useful when defining a legal range for date fields (e.g., 12/25/2007-today).

$(choices): Inserts the numeric codes for all possible response options (including no box checked) for check and choice fields.

$(ids): Inserts the concatenated list of participant ID ranges from all sites in the centers database. This meta-word is typically used to specify the legal values property of the participant ID field.

14.2 Meta-Words for Help specifications

$(legal): Inserts the specified legal range, useful in help messages (e.g., “Legal range is: $(legal)”).

14.3 Meta-Words for Field Name specifications

$(field): Substitutes the field number as displayed in the center of the field widget in the CRF window.

$(plate): Substitutes the CRF plate number in three-digit zero padded format. For example, $(plate) for CRF plate numbers 1, 55 and 101 will be 001, 055, 101.

$(rplate): Also substitutes the CRF plate number. It is the same as $(plate) except that the value is not zero padded.

These three meta-words can be used to generate automatic field names. For example, locking x$(plate)_$(field) in the generic and/or unique field name property of all styles would generate field names by plate and field number for all data fields (e.g., x22_14 for field 14 on plate 22).

14.4 Meta-Words for Skip specifications

$(blank): Indicates that no box was checked for check and choice fields, and indicates a null string for all other field types. This meta-word can be used to skip fields when the current field is blank. !$(blank) can be used to skip fields when the current field is not blank.

$(legal): Inserts the specified legal range. Can be used to skip fields if the current field contains one of these values. !$(legal) can be used to skip fields when the current field does not contain one of these values. This meta-word can not be combined with any other values when constructing a skip specification.
15.0 Centers Database

The study-specific Centers Database documents the location of each participating study site, the contact person, and the participant ID number ranges. DataFax quality control and other reports use the centers database to determine the center to which each subject belongs.

The centers database is accessed in DFsetup by going to the View menu and choosing Centers.

15.1 Configuration controls

All of the study configuration dialogs described in the following sections (Missing Value Codes, Page Map, etc.) have the same controls, which are described below.

- **View:** Used to select a study configuration dialog.
- **Arrows:** Move selected row up or down (not present in the centers dialog).
- **Copy icon:** Copy the selected row.
- **Paste icon:** Insert previously copied row above the current row.
- **Add:** Open a new blank row below the current row.
- **Delete:** Remove the current row.
- **Import:** Replace all rows with information from another study.
- **Import File:** Add or replace rows with information from a file.
- **Export:** Export information to a delimited text file.
- **Search:** Case insensitive string match on all fields.
- **Revert:** Reload last saved version.
- **Verify:** Check last saved version.
- **Save:** Save all rows to DataFax server.
- **Done:** Quit the dialog.

15.2 Required Elements

The following information is required in the centers database:

15.2.1 Center Number

A one- to four-digit number that uniquely identifies the center within the database. The legal range for center numbers is 1-2146.

15.2.2 Contact Person and Affiliation

Name of the primary contact to whom QC reports will be addressed. Affiliation appears on the quality control report following the contact person’s name.
15.2.3 Primary Fax

List of fax numbers and/or email addresses to which QC reports should be sent. To send reports to multiple recipients, separate each fax number or email address with a single space.

Fax: Each fax number must contain all of the digits required to dial the destination fax machine (i.e., long distance code, country code, area code, etc.). May be formatted to aid readability (e.g., (789)-123-4567) but cannot contain spaces.

Email: Each email address must start with the prefix `mailto:` followed by the complete email address, e.g. mailto:jack@somewhere.com.

15.2.4 Patients

Comma delimited list of all participant ID numbers and ranges belonging to each center. Each participant can belong to one and only one center.

15.2.5 Reply To

Email address of the person who will receive any replies to QC reports that were sent to the center by email. This is also the person who will receive email transmission failure notifications. If a QC report is sent via email and this field is blank, reply emails or transmission failures will be delivered to the DataFax problem mail recipient defined by the DataFax administrator in DFSystem.

15.2.6 Error Monitor

A special entry must be included in the centers database for an error monitor, who is typically someone at the study coordinating center. This special center is identified by the string ERROR MONITOR in the Patients field. Any data records with participant IDs that do not belong to one of the clinical sites are automatically assigned to the error monitor center. This might occur, for example, if the centers database is modified after participants have been entered.

Only one error monitor center can be defined for each study. By default, center number zero is used for this special center, but this can be changed to any valid center number.

15.3 Center dialog: optional elements

The following optional elements are not required by DataFax.

- **Address**: Postal address
- **Secondary Fax**: Other fax or email address for the contact person.
- **Phone**: Phone number of the contact person.
- **Investigator**: Name of the principal investigator at the center.
- **Inv. Phone**: Phone number of the principal investigator at the center.
### 16.0 Participant Visit Scheduling: DataFax Visit Maps

The visit map is a schedule of (1) required and optional visits for the study and (2) required and optional plates for each visit. It is used to generate missing page and overdue visit queries, and it also determines how participant binders are organized and displayed in iDataFax. Thus, data entry cannot begin until a visit map has been created.

The visit map is accessed in DFsetup by going to the View menu and choosing Visit Map. The order in which visits appear in the visit map table must correspond to the order in which they are performed during the trial. The location can be adjusted after the visit is defined by using the up and down arrows.

![Visit Map Example](image-url)
16.1 Basic Visit Map Elements

Each row of the visit map consists of the following basic elements. Some elements can be edited in the table itself, while others are edited in the “Visit or Cycle Definition” panel to the right of the visit map window.

**Visit or Cycle**: Determines whether the row defines an individual visit or a cycle. More information on cycles is included later in the manual.

**Type**: Select the appropriate visit type from the drop-down menu. The one letter acronym for the selected visit will appear in the first column of the visit map table.

A detailed description of the visit types can be found in the DFsetup User Guide. The following table provides a summary of each visit type:

<table>
<thead>
<tr>
<th>Visit Code</th>
<th>Visit Type</th>
<th>Scheduled</th>
<th>When Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Screening</td>
<td>No</td>
<td>If subject enters the trial (i.e., baseline visit arrives)</td>
</tr>
<tr>
<td>P</td>
<td>Pre-baseline</td>
<td>Yes</td>
<td>Before arrival of baseline visit</td>
</tr>
<tr>
<td>B</td>
<td>Baseline</td>
<td>Yes</td>
<td>Can be scheduled from a pre-baseline visit</td>
</tr>
<tr>
<td>S</td>
<td>Scheduled Follow-up</td>
<td>Yes</td>
<td>Scheduled from the baseline visit</td>
</tr>
<tr>
<td>O</td>
<td>Optional</td>
<td>No</td>
<td>Not required</td>
</tr>
<tr>
<td>T</td>
<td>Termination</td>
<td>Yes</td>
<td>Scheduled from the baseline visit</td>
</tr>
<tr>
<td>E</td>
<td>Early Termination</td>
<td>No</td>
<td>If early termination event occurs</td>
</tr>
<tr>
<td>A</td>
<td>Abort</td>
<td>No</td>
<td>If abort event occurs</td>
</tr>
<tr>
<td>r</td>
<td>Required</td>
<td>No</td>
<td>Before arrival of next visit</td>
</tr>
<tr>
<td>R</td>
<td>Required</td>
<td>Yes</td>
<td>On termination if scheduled pre-termination</td>
</tr>
<tr>
<td>F</td>
<td>Final Visit</td>
<td>Yes</td>
<td>Used to terminate multi-cycle visit maps only</td>
</tr>
<tr>
<td>W</td>
<td>Termination Window</td>
<td>Yes</td>
<td>Used in studies ending on a specified date</td>
</tr>
</tbody>
</table>
Number: The visit (or sequence) number is entered in the second column of the table.

Note: Within DataFax, every record has a sequence number. Typically we refer to the sequence number as a visit number when it is used to identify those CRFs that belong together at a scheduled visit. Visit and sequence numbers are referred to as assessments in iDataFax.

In some cases, such as the adverse event and concomitant medication example below, the visit number will be entered as a series. In this example, adverse event page 1 will be visit 6001, page 2 will be 6002, etc.

A series can be entered using a dash (-) or a tilde (~). If a dash is used, a new blank page with the next visit number will be added in iDataFax as soon as the first page is entered. This method is useful for EDC. If a tilde is used, the visit does not appear in the iDataFax record navigation list until it is received. This method is useful for paper studies or for visits that occur rarely that you do not want to display in the iDataFax navigation list.

Be careful not to use the same value for visit and sequence numbers. For example if visit numbers for scheduled visits fall in the range 0-20, sequence numbers for (unscheduled) adverse events might fall in the range 101-199.

Acronym: A short acronym can be entered in the third column of the table. The acronym is used to identify the first, last, and next scheduled visits in the patient status summary section of the DataFax QC report.

Label: The visit label, entered in the fourth column of the table, is used in overdue visit queries. Therefore, it should clearly identify each visit.

Visit Date Field: For scheduled visits, select the plate with the visit date field (defined using the VisitDate style) for that visit.

Visit Scheduling: If a visit is scheduled a scheduling method must be specified. Most visits are scheduled a specified number of days following the cycle baseline visit.

Overdue Allowance: Visits are considered overdue when the overdue allowance has expired. For example, if a visit is scheduled to occur on day 100 with a 10 day window, it can be scheduled as “due 100 days after baseline” and “overdue when 10 days late.” For paper studies, the overdue allowance should be increased beyond the window to allow additional time for completion and faxing of data forms.
**Required Plates:** A comma delimited list of plate numbers and ranges can be specified, e.g., 5,10-13,20,27-29. The order of the plates in the visit map determines the order in which they will appear in the patient binders in **DataFax**.

**Optional Plates:** A comma delimited list of plate numbers and ranges can be specified. The order of the plates in the visit map determines the order in which they will appear in the patient binders in **DataFax** patient binders. Optional plates must be included in the visit map to make them available for data entry.

**Missed Visit Plates:** If a special missed visit plate has been defined to record reasons for missed visits, identify it here.

Even if a study has no scheduling requirements, (e.g., a survey), a single entry must be made in the visit map. This entry should be defined as visit type “Screening” or “Baseline” and should include the information listed above.

### 16.2 Example of a Single Cycle Visit Map

The following example shows a visit map as it appears in the DFvisit_map configuration file. Each record has the format:

Visit # | Type | Label | VisitDatePlate | VisitDateField | Due Day | Overdue Allowance | Required Plates | Optional Plates | Missed Visit Plate | Termination Window
--- | --- | --- | --- | --- | --- | --- | --- | --- | --- | ---
1 | X | Screening | 1 | 9 | 0 | 0 | 1 | | |
2 | P | Pre-Baseline Lab Test | 2 | 10 | -14 | 0 | 2-3 | | |
3 | B | Baseline | 4 | 9 | 0 | 0 | 4-7 | 8 | |
4 | S | 1 Month Follow-up | 10 | 9 | 30 | 5 | 10-14 | | 15 |
5 | S | 2 Month Follow-up | 10 | 9 | 60 | 5 | 10-15 | | 15 |
6 | S | 3 Month Follow-up | 10 | 9 | 90 | 5 | 10-15 | | 15 |
7 | T | Subject Termination | 20 | 9 | 90 | 10 | 20 | | |
98 | A | Lost to Follow-up | 30 | 9 | 0 | 0 | 30 | | |
99 | A | Subject Death | 40 | 9 | 0 | 0 | 40 | | |

The number shown in field #5 for all records in the above example represents the field number of the DataFax VisitDate field, as defined in **DFsetup**. Only dates defined with the VisitStyle are used for subject scheduling and must appear in the study visit map. A more detailed description of VisitDates appears later in this chapter.

The above example begins with a screening visit (visit type X, visit 1) that contains a VisitDate date on plate 1, field 9 but does not contain a due date or overdue allowance. Screening visits cannot be scheduled and are not required unless subsequent scheduled visits arrive.

Following Screening is a required pre-baseline lab test (visit type P, visit 2), due 14 days before baseline. This visit is considered the first of the scheduled visits. The next visit, visit 3, represents Baseline or subject randomization. For baseline, due date and overdue allowance are both set to 0. DataFax schedules all visits from Baseline.

Three scheduled visits follow baseline at 1, 2, and 3 months. The due day of each visit is specified in days from baseline (day 0). Each has been given an overdue allowance of
5 days. Thus, visit 6 (3-Month Follow-up) will not be queried as overdue until the 96th day after baseline.

Following the scheduled visits is a termination visit (visit type T, visit 7) that is due at the same time as the 3-month scheduled visit (visit 6). A 10-day overdue allowance has been defined for this termination form to give investigators several additional days to review and finalize termination data for the subject.

The last two entries in the visit map do not apply to actual subject visits. Two Abort visits have been defined to deal with subjects who are lost to follow or who die while enrolled in the study. Upon arrival of an abort visit, overdue visit notification is limited to those scheduled visits that were due to occur prior to the abort date.

### 16.3 Numbering Visits in a Visit Map

Visit numbering within a visit map should be given some thought and not simply assigned arbitrarily. Here are a few tips that will ensure that your visit map will correctly schedule and report the various study assessments:

- **Visit/sequence numbers must be integers in the range of 0-65535.** Negative numbers are not allowed.

- **Numbers must be unique for each visit within a study.**

- **Although visit numbers are typically chosen sequentially, this is not a requirement.** Gaps in numbering are legal and suggested to allow for the addition of extra visits after study start-up, if necessary.

- **Event reports (e.g., adverse events, medication logs) should be assigned a range of sequence numbers and placed at the end of the visit map, following all scheduled visits.** Be sure that sequence numbers do not match or fall within the range of the scheduled visit numbers.
16.4 Visit Dates and the VisitDate Style

Visit date fields are used to record the day on which a visit occurred or on which patient follow-up was terminated. They are used for patient scheduling and should be specified as follows:

- All visit date fields must use the same date format and be defined using the VisitDate style. The date format must be defined at the style level (not the field level).
- A visit date field must appear on at least one CRF page for each baseline and scheduled follow-up visit. They may also appear on optional visits but are only required if the optional visit may become required when some condition is met as specified in the conditional visit map.
- A visit date field must exist for any event that terminates patient follow-up (e.g., final visit date, early termination date, date of death, etc.). These dates are used to halt visit scheduling to prevent unnecessary overdue visit queries.
- The VisitDate style must not be used for unscheduled events like the onset dates of an adverse event.
- Each CRF page can have only one visit/assessment number (field 6) and thus can also have only one VisitDate field identifying when that visit occurred.

Often there will only be one VisitDate field defined for each visit in the visit map. If two or more CRF plates completed at a particular visit each contain a VisitDate date field, the same date should be recorded in each of the fields. If DataFax finds inconsistencies between dates using the VisitDate style for the same visit, it will report the problem (when DF_XXkeys is run) and will generate a DataFax retrieval file that can be used to review the inconsistent cases.

While it is illegal to have different visit dates recorded for the same visit, it is legal to have the same visit date recorded for different visits. However, within the ordered list of study visits, defined within screening and in-study cycles in the study visit map, it is illegal for visit dates to be out of chronological order. Thus the same visit date should only appear, if at all, for adjacent visits.

Once the plates and visit map have been defined in DFsetup, the following two DataFax reports should be run to check for visit date inconsistencies:

1. DF_SSvisitmap: Displays the visit number, visit type, target day, overdue allowance, required plates and all date fields defined with the VisitDate style in the study schema.
2. DF_ICvisitmap: Checks the visit map specifications for compliance with DataFax visit map rules and checks that VisitDate dates have been defined for all visits that require them.

**Note:** The on-line documentation for each of the above reports provides a detailed explanation of each report. This documentation can be accessed in the Reports Tool by selecting the desired report from the reports list, then selecting the Explain button.
16.5 Multi-cycle Visit Maps

Multi-cycle visit maps are used in studies where subjects go through more than one cycle of treatment and follow-up (e.g., a cancer trial). Visit scheduling within each cycle is relative to a cycle baseline visit, and cycles may be required, optional or conditional based on some event.

The general principles for defining visit types, VisitDate dates, and missing and overdue visits described above also apply to multi-cycle visit maps. However, there are some additional rules for multi-cycle visit maps that do not apply to single cycle visit maps. These are briefly discussed below. For a more detailed description of multi-cycle visit maps, see the DFsetup User Guide.

- Each cycle must begin with a cycle definition record, which identifies the cycle number, label, type (required, optional, conditional) and scheduling method. A cycle start record might appear as follows:

  1 | C | Treatment Cycle 1 | R | 0 | 0 |

  Where
  
  cycle number = 1; cycle definition record = C; label = Treatment Cycle 1; type = R (required); scheduling = due on day 0, overdue allowance of 0 days.

- A screening visit (visit type X) and optional visits (type O) may precede the cycle start record for the first cycle.

- Scheduling methods include the following options:
  
  - From the first visit of cycle 1.
  - From either the baseline or termination of the previous cycle.
  - From the visit number of the record used to trigger a conditional cycle.
  - From a specified preceding visit number.

- The first cycle is always considered required (type R).

- Optional cycles (type O) are not scheduled. However, arrival of any visit defined within an optional cycle makes that cycle required and activates the scheduling of visits within the cycle.

- For conditional cycles, the condition indicating that the cycle is required is triggered by a specified value for a specified data field. The trigger for a conditional cycle is best defined as a simple Yes/No question and should be considered when designing the study CRFs. It is legal to have more than one conditional cycle triggered by the same value, provided that the cycles occur at different points in time.

- All visits defined after the cycle start record but before the next cycle start record belong to that cycle.

- Following the cycle start record, each cycle must begin with either a baseline (type B) or a pre-baseline (type P) visit. Each cycle must also contain a single termination (type T) visit.
• Within each cycle, visit scheduling is defined in days since the cycle baseline visit. The due day and overdue allowance for each baseline visit must be set to zero.

• An end cycles record must be defined at the end of all cycles. An end cycles record might appear as follows:

  5 | C | End Cycles | E | 0 | 0 |

  Where
cycle number = 5; cycle record = C; label = End Cycles; type = E (end cycles); scheduling = due on day 0, overdue allowance is 0 days (the end cycles record is not a scheduled cycle; it merely signals the end of all study cycles).

• Optional (type O) and abort (type A) visits that do not belong to a specific cycle may be defined after the end cycles record. Only type O and type A visits are allowed after the end cycles record.

• A termination (type T) or an early termination (type E) visit only terminates the cycle in which the visit is defined. This rule also applies to termination plates (defined in the Plate dialog of DFsetup), i.e., arrival of one of these plates in a cycle only terminates the cycle containing the assessment number recorded on the termination plate.

• A Final (type F) or an abort (type A) visit terminates all cycles. Type F visits have the status of both baseline and termination visits and are most often defined as a single visit in the last cycle. Only type R, O, E and A visits may follow a type F visit.
16.6 Missing Pages, Overdue Visits, and Missed Visit Forms

Missing pages and overdue visits are identified by DataFax's quality control report programs, specifically DF_QCupdate. To identify missing pages, DF_QCupdate checks the pages that have arrived for each visit against the list of required plates in the visit map. Missing pages are flagged with quality control (QC) notes that get added to the QC database. Overdue visits are flagged when the visit’s due date plus overdue allowance has expired and no plates for that visit have been received.

To check for overdue visits, DF_QCupdate identifies all visit numbers and dates defined with the VisitDate style that have arrived for each subject. It then compares the list of visits that have arrived with its own calculation of the visits that should have arrived (as defined in the visit map), and creates overdue visit notifications for any visit that it determines to be overdue.

If an overdue visit has been identified and it is known that it will never be received, DataFax must be notified. This will prevent an overdue visit notification from re-appearing in QC reports. There are two ways to do this:

1. Mark an assessment lost in iDataFax.
2. Instruct investigators to complete a Missed Visit CRF and fax it to the DataFax study.

Option #1 may be tedious. Thus, if you anticipate missed visits, it is a good idea to define a missed visit plate in the study visit map. Arrival of this plate, with the missed visit number recorded as its assessment number, will stop DataFax from complaining that the visit is missing or overdue.

- A Missed Visit Notification page need only consist of a single CRF. Data fields must include the Subject ID and the visit number of the visit that has been missed. Optional fields might include a date on which the form was completed and a reason why the visit was missed.

16.7 Early Termination Plates

In addition to defining visits that signal termination (T, W, E, A, and F), it is also possible to define early termination plates (using the “Plate Triggers Early Termination” column in View-Plates). These plates will signal early termination of the cycle with which they are associated. Therefore they are like termination (T) and early termination (E) visits. They cannot be used to define an abort event, like final (F) and abort (A) visits.
17.0 Conditional Maps

Some studies have very basic visit scheduling requirements. Others are more complex and have visits and CRFs that may be required or optional based on certain conditions. The conditional plate, visit, cycle, and termination maps can be used to meet these requirements. The conditional maps are accessed in DFsetup by going to the View menu and choosing Conditional Plate, Conditional Visit, etc.

Each map has the same simple structure: Each condition begins with a description, which is followed by a database test and a list of actions to be performed if the test is true.
17.1 Conditional Plate Map

The conditional plate map is used to make certain plates required, optional, or excluded at particular visits. For example:

- Subjects randomized to a surgical treatment arm may require different forms compared to subjects in a medical treatment arm.
- Male and female subjects may require different forms at the same visit.

The basic rules for a conditional plate map are as follows:

- There is no limit on the number of conditions that may be included but each condition must be unique.
- The conditional plate map works for all visit types, so it can be used for both optional (O) and scheduled (S) visits. However, the visit that triggers the condition must be defined in the visit map.
- The visit number may be replaced by * in both the test and action lists. When * appears in the test it means that the condition is to be tested at all visits. When * appears in an action list, it means that the plates are required at the visit that triggered the condition.
- It is legal to include more than one action following each test. It is also legal for an asterisk to appear for some of these actions and for visit numbers to be specified for others.
- It is legal for a conditional plate map to specify the same conditional plate requirement (i.e., same plate at the same visit) via more than one condition. If a particular conditional plate requirement is triggered by more than one condition DF_QCupdate will only generate one missing page QC note.
- Plate numbers may be listed in any order in the list of conditional plates. Both plate ranges and single plate numbers are valid and must be separated by a comma or a single space.
17.1.1 The “Value” Field

The value that is used in the condition test may be any of the following:

- Number
- String
- Missing value code
- The word “missing” (to represent any missing value code)
- The word “blank” (to represent a blank field)

**Note:** Do not include double quotes (""") when using the key words “blank” or “missing.”

- Blank (which also represents a blank field)

The value may be specified as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Condition is met if:</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The field is equal to the value (e.g., 2)</td>
</tr>
<tr>
<td>!value</td>
<td>The field is not equal to the value (e.g., !2)</td>
</tr>
<tr>
<td>&lt;value</td>
<td>The field is less than the value (e.g., &lt;2)</td>
</tr>
<tr>
<td>&gt;value</td>
<td>The field is greater than the value (e.g., &gt;2)</td>
</tr>
<tr>
<td>~value</td>
<td>The field contains the string value (e.g., ~AE)</td>
</tr>
<tr>
<td>blank</td>
<td>The field is blank</td>
</tr>
<tr>
<td>!blank</td>
<td>The field is not blank</td>
</tr>
<tr>
<td>missing</td>
<td>The field contains any missing value code</td>
</tr>
<tr>
<td>!missing</td>
<td>The field does not contain a missing value code</td>
</tr>
</tbody>
</table>

**Note:** Check fields have a numeric code when they are blank, thus they are never blank in the database.
17.2 Conditional Visit Map

The conditional visit map is used to make certain visits required, optional, or excluded. All of the conditional maps perform in the same way. Much has already been discussed in this section and will not be repeated here. For more details please see the User Guide.

The following example shows how the conditional visit map can be used to check for gaps in a sequential sequence numbers:

In this example, the Adverse Event Log (plate 200) uses sequence numbers (field 6) 20001 through 20999. If an AE log page is entered with a sequence number greater than 20001 (e.g., 20007), then all AEs with sequence number 20001 through 20007 should also have been received.
17.3 Conditional Cycle Map

The conditional cycle map is used to make certain cycles required, optional, or excluded.

The following example shows how conditions can be used to make certain cycles required or excluded.

In this example, if the patient tests positive, \((\text{field } 9 = 2)\) then cycles 1,2,3,4, and 5 are excluded.

17.4 Conditional Termination Map

The conditional termination map is used to specify early termination (for the current cycle) or abort all follow-up (for all cycles) based on certain conditions.

The following example illustrates how screen failures are handled in one study:

If field 12 on plate 1 at visit 100 equals 1 (yes), then patient will be triggered for early termination.

When conditional termination is triggered, the visit date (defined using the VisitDate style) for the trigger visit becomes the termination date. Therefore, you should not trigger termination using a visit for which there is no visit date.
18.0 **DataFax Page Map**

The page map is used to convert plate and visit number combinations into human-readable text to be used in QC reports and in the iDataFax Queries and Data views.

Queries are identified in iDataFax and in QC reports by: patient ID, visit number, plate number, and field description. While patient ID and field description are always meaningful, in many studies the visit number and plate number are meaningless to the clinical sites and should be replaced by a more descriptive label.

The page map is accessed by going to the View menu and choosing Page Map. The control buttons at the bottom of the window are the same for all study configuration dialogs. [See Centers Database (page 35) for a description.]

The Plate/Visit Title is optional, and is only relevant in the short version of QC reports, where it appears as a title over the page map labels on the refax section of each report. If specified, it should be no more than 17 characters long. If not specified, the default title is 'PLATE SEQNO.'

![Page Map Image]

The plate and visit columns may contain a single value or range, a comma delimited list of values and ranges, or an asterisk (*) to represent all possible values.

In the long version of QC reports, page map labels have a maximum length of 32 characters. This is achieved when variable labels are set to their maximum length (40 characters) in the Global Settings. More compact QC reports can be created by setting variable labels to 25 characters in Global Settings. Page map labels are then limited to 17 characters. Page map labels will be truncated in QC reports if they exceed these limits.
18.1 Page Map Meta Characters

Meta characters can be used to simplify page map specifications. When used in a page label, these characters are replaced with a plate number, a sequence number, or a data field value. Meta characters should not be confused with Meta Words, which are used to simplify variable attributes.

Meta characters include the following:

- An asterisk (*) can be used to represent all plate or all sequence numbers
- %P is replaced by the plate number without zero padding.
- %S is replaced by the visit number without zero padding.
- %{P.n} is replaced by the last n digits of the 3 digit zero padded plate number.
- %{n.P} is replaced by the first n digits of the 3 digit zero padded plate number.
- %{S.n} is replaced by the last n digits of the zero padded sequence number.
- %{n.S} is replaced by the first n digits of the zero padded sequence number.
- %{S.a.b} is replaced by b characters of the zero padded sequence number starting from character position a.
- %n is replaced by the value of field number n from the relevant data record. Only one field value substitution is allowed per label.
- %n:d is for check and choice fields; field n is decoded and its label is used. Only one of these substitutions is allowed per label.

In the following example, only one entry is made in the Page Map for a CRF that appears at most required and many optional visits. Instead of spelling out each permutation (e.g. PERS 01.0, 03.2, 17.1, etc.), meta characters were used:

| 121 | 0301 0401... | %{S.1.2} %{S.3.1} FVS |
| 121 | 0211 0221... | %{S.1.2} %{S.3.1} FVS |
| 135 | 901    | %{S.1.2} %{S.3.1} RCI |
| 135 | 0211 0221... | %{S.1.2} %{S.3.1} RCI |
19.0 Missing Value Codes

The missing value map is used to define missing value codes and labels for the study. In iDataFax these codes can be assigned to any optional or required data field, and can be thought of as pre-approved reasons for missing data. Missing value codes cannot be applied to data fields defined as essential.

New missing value codes can be added during a study, but you should not remove or reassign missing value codes that are in use (this could render some data fields invalid). There is no harm in updating the label on existing missing value codes (e.g., to correct spelling mistakes or poor wording), as long as you do not change the meaning. Only the code is stored in the study database. Thus any change to the label will apply to both past and future use of that missing value code.

DataFax imposes no restrictions on the choice of missing value codes, but you should consider restrictions imposed by other software you plan to use (such as SAS).

The missing value map is accessed by going to the View menu and choosing Missing Value Codes.
20.0 Edit Checks Dialog

Edit check programs are written, checked, and published using the Edit Checks dialog (accessible by going to the View menu and choosing Edit Checks). The Edit Checks dialog is briefly described below. See the Edit Checks chapter in the DataFax Programmer Guide for a description of the edit check language.

![Edit Checks Dialog](image)

The buttons at the bottom of the dialog are used to manage the edit check source files, which are listed in the left panel. These buttons include:

Add: Create a new empty edit check file.
Clone: Create a copy of the selected edit check file.
Rename: Rename the selected edit check file.
Delete: Delete the selected edit check file.
Import: Import a copy of an edit check file from another study.
Save: Commit changes made to an edit check file to the server. As for all configuration dialogs, saves are immediate and any changes not saved before DFsetup exits or times out will be lost.

Publish: Make edit checks available to iDataFax. Only control files DFedits and DFedits_test can be published. Switch to the “Results” section at the bottom of the dialog before you click the Publish button to see any error messages.

Done: Close the Edit Checks dialog.

When a source file is selected, it is retrieved from the server and displayed in the text edit window on the right where it can be modified and saved back to the server.

Any edit check source file can be checked for syntax errors by selecting it and then clicking the “Check Syntax” button within the “Results” section at the bottom of the dialog, but only DFedits and DFedits_test can be published for iDataFax. These special files correspond to the Production and Test edit check options available in iDataFax. All of the other edit check source files must be referenced in DFedits and DFedits_test using include statements.

Users with DataFax-Developer permission are able to reload both test and production versions of published edit checks while working in iDataFax. This is particularly useful when testing edit checks. Users without DataFax-Developer permission get the most recently published version of the production edit checks (from file DFedits) when they login to iDataFax.
21.0 Lookup Tables

Lookup tables are used in iDataFax to select results from a list of pre-defined values for insertion into DataFax string data fields. Lookup tables can also be used for query, note, and reason fields in QC notes.

Lookup tables are created and modified in DFsetup by going to the View menu and choosing Lookup Tables. The available lookup tables are listed in the left panel. Click on the name of a table and it will displayed in the text edit window on the right. Only study level lookup tables can be modified and saved. Lookup tables stored at the DataFax level are displayed in blue. These files must be managed by a UNIX system administrator.

With the Lookup Table dialog, you can edit existing tables or add, clone, rename, delete, and import new tables. The User Guide provides detailed instructions on how to create and modify various types of tables.
21.1 Lookup Table Map

A study may use multiple lookup tables, so the lookup table map is used to associate simple lookup table names with full pathnames of the actual lookup tables. The Lookup Tables Map dialog is used to enter a lookup table name by which each lookup file will be known in edit checks and in iDataFax. It is accessed by going to the View menu and choosing Lookup Tables Map.

![Lookup Tables Map Dialog](image)

There are 3 reserved lookup table names: QC for standard queries, QCNOTE for standard query note text and REASON for standard reasons. These names tell iDataFax where to find these lookup tables when a user clicks the lookup buttons in the add query and add reason dialogs. All other names are for use in edit checks and can be created using any string of letters and numbers.
22.0 Modifying Plates after Study Startup

Ideally, you won’t need to make any changes to the database after the study has started. However, in reality most studies require database changes: new CRFs, new fields, deleted fields, etc.

Because the field numbers on each plate match the order in which fields are stored in the study database, inserting or deleting fields after data has been collected must be done with care.

Beginning with version 3.9, DataFax provides support for field numbering changes that occur after data has been collected, because fields have been reordered, added, inserted or deleted. Such changes can only be made using the Exclusive or Developer modes in DFsetup. When the changes are saved DataFax will:

• reformat the existing data records to match the new field order
• update queries and reasons so they remain attached to the correct data fields
• save the old setup specification, and add setup revision records to the study journal files so that the audit trail report will be able to keep track of the field changes

However, this is all that DataFax can do automatically. In addition, you may need to make the following revisions manually:

• Edit checks that refer to fields on the revised plate by number or using the @T relative position notation may need to be revised because of field number and/or position changes.
• Shell scripts that use DFexport.rpc to export data records for custom reports, plate arrival triggers, dfexecute() edit check functions, or other tasks performed by custom programs will need to be modified to deal with the new field order.
• DFSas job files used to create SAS data sets containing data from revised plates will need to be modified to account for any new or deleted fields.
• Task and List View definitions for the revised plate may need to be corrected.
• If DFsqlloader is used to export study data to an SQL database the table definition will be updated and the restructured data reloaded the next time DFsqlloader is run. However, you may need to revise SQL procedures stored in your SQL database to deal with the plate revisions.