



Batch Edit Checks

Clinical DataFax Systems Inc.

<http://www.datafax.com>

Disclaimer

This document contains proprietary information of Clinical DataFax Systems Inc. The terms of your license agreement with Clinical DataFax Systems Inc. apply.

Although every effort has been made to ensure up-to-date and accurate information, the program described in this document is still under development. Before the final release date Clinical DataFax Systems Inc. reserves the right to make changes to both the program and this documentation.

Acknowledgements

Our thanks to Crane Softwrights Ltd. for the gracious contribution of the software used to create this presentation.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Batch Edit Checks



-
- Introduction - Batch Edit Checks
 - Module 1 - What are Batch Edit Checks?
 - Module 2 - Input Control File
 - Module 3 - Processing Batch Edit Checks
 - Module 4 - Batch Edit Checks Output
 - Module 5 - Implementation Considerations
 - Module 6 - Complete Example
 - Module 7 - Creating Your Own Stylesheet(s)

Pre-requisites:

- knowledge of edit checks language
- familiarity with plain text editor
- familiarity with UNIX command line

Outcomes:

- understanding of batch edit checks process
- awareness of appropriateness and applicability

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Batch Edit Checks



Introduction

This presentation describes the batch edit checks component of DataFax.

The first official release of batch edit checks was as part of DataFax 3.5, but has been available via the Early Access Program since November 2000.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

<http://www.datafax.com>

Batch Edit Checks

Copyright © Clinical DataFax Systems Inc.

Notes:

Module 1 - What are Batch Edit Checks?



-
- Introduction - Overview
 - Lesson 1 - Why are Batch Edit Checks Needed?
 - Lesson 2 - Requirements for Batch Edit Checks
 - Lesson 3 - Information Flow

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Overview



Module 1: What are Batch Edit Checks?

This module overviews the justification for batch edit checks and the design of the solution.

Problem:

- requirements for data integrity checking evolve over the length of a DataFax study
- study databases grow to be very large
- not feasible to interactively (re-)apply evolving data integrity checks

Requirements:

- use existing edit check language
- unattended execution
- comprehensive logging

Information Flow:

- input control file
 - non-interactive processing environment
 - output log file
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Loose Coupling



Module 1: What are Batch Edit Checks?

Lesson 1-1: Why are Batch Edit Checks Needed?

The edit check environment and the database are loosely coupled.

- edit checks can be changed independent of the database
 - existing logic can be changed
 - how would one re-validate a single modified edit check?
 - edit checks can be added
 - edit checks can be removed/deleted
- data in the database can be changed independent of the edit checks
 - edit checks can be disabled in the validation tool
 - edit checks are not applied to data changes via `DFimport.rpc`
 - study schema can be changed in the setup tool

A mechanism is needed for synchronizing edit checks and the database.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Limits



Module 1: What are Batch Edit Checks?

Lesson 1-1: Why are Batch Edit Checks Needed?

Study databases grow very large.

- validation tool "restricts" retrieval sets to 4096 records

Interactive validation can be a tedious task.

- humans are error-prone
 - humans have limited attention spans
-

CDSI::DFUG::BATCHEBITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Re-use existing edit checks



Module 1: What are Batch Edit Checks?

Lesson 1-2: Requirements for Batch Edit Checks

Most users have a significant investment in their existing edit checks.

- the edit checks language should not change
 - any additions must be backwards compatible
- capabilities of interactive edit checks must be accommodated
 - sensible defaults are needed where interaction is not possible

Use of batch edit checks is not required to achieve what you can already do with existing edit checks.

CDSI::DFUG::BATCHEBITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Execute non-interactively



Module 1: What are Batch Edit Checks?

Lesson 1-2: Requirements for Batch Edit Checks

Take advantage of the scheduling facilities available in the host environment.

Requires a non-interactive mechanism for specifying what to do.

- control file that specifies record retrieval and/or edit check selection criteria

During processing, there is no (graphical) user interface.

- cannot query the user when something unexpected happens

Logging is critical.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

<http://www.datafax.com>

Batch Edit Checks

Copyright © Clinical DataFax Systems Inc.

Notes:

Log everything



Module 1: What are Batch Edit Checks?

Lesson 1-2: Requirements for Batch Edit Checks

Create a complete record of what happened.

Make log creation independent of presentation.

- this allows the log to be (re-)viewed in different ways without affecting the integrity of the log
- a detailed view may be needed by the programmer that is debugging
- a summary view may be needed by the database manager

Allow for easy identification of incremental changes.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Input Control File



Module 1: What are Batch Edit Checks?
Lesson 1-3: Information Flow

Specifications are defined in a text file.

- familiarity with a text editor is essential
- a subsequent version may include a graphical editor for creation of the input file

Specifications must address two needs.

- selection criteria for records to retrieve and/or edit checks to execute
 - actions to take as edit checks are applied
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Processing Environment

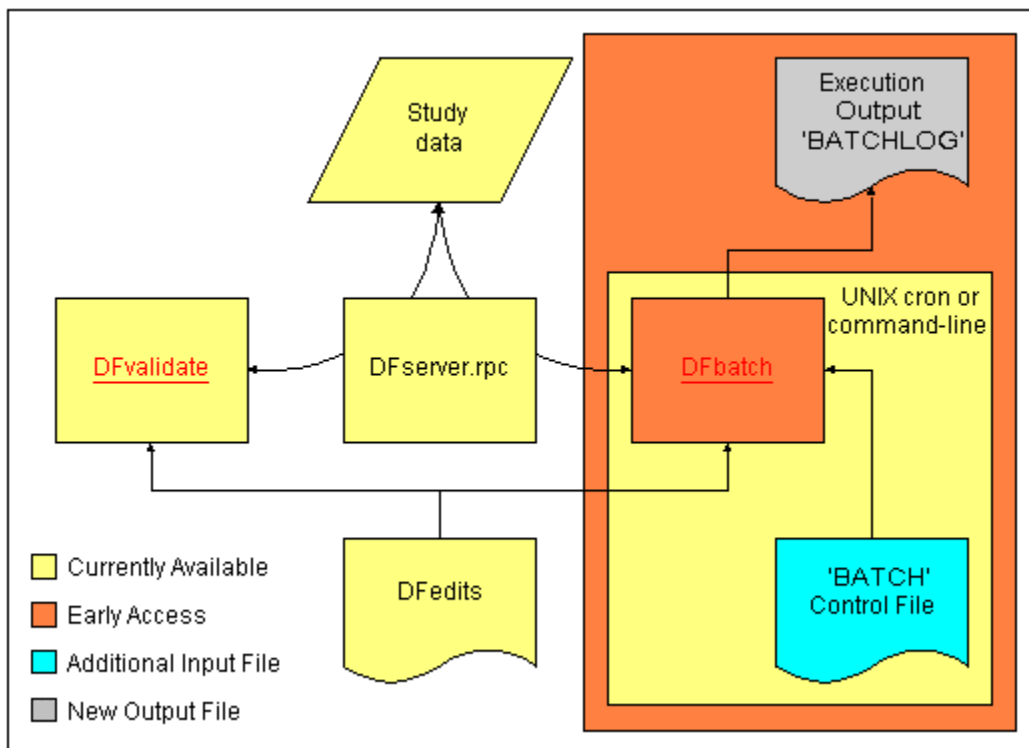


Module 1: What are Batch Edit Checks?

Lesson 1-3: Information Flow

Processing model is analogous to the validation tool, minus the interactive component, plus the logging component

- control file specifies records to process
- records flow via `DFserver.rpc` from database to batch environment
- edit checks are applied to each record
- changed records and new/deleted QC notes flow via `DFserver.rpc` from batch environment back to database
- actions are recorded in execution output



CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Output Log File

Module 1: What are Batch Edit Checks?
Lesson 1-3: Information Flow



Log is written to a text file.

- simplifies optional post-processing steps
- is also human-readable

Logs context in which batch execution occurred.

Logs all records that met selection criteria.

- input control file can stipulate that only changed records be logged

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Module 2 - Input Control File



-
- Introduction - Overview
 - Lesson 1 - XML Syntax and Semantics
 - Lesson 2 - DataFax Semantics
 - Lesson 3 - Element Reference

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Overview



Module 2: Input Control File

This module overviews the creation and content of input files.

The language of the input file is eXtensible Markup Language (XML).

- XML will be frequently used for textual information in future versions of DataFax so what you learn now will also benefit you in the future

XML is about markup and the rules of markup.

The eXtensibility comes from you and the vendors that you interoperate with (CDSI in this case).

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

XML Syntax



Module 2: Input Control File
Lesson 2-1: XML Syntax and Semantics

The language building blocks are elements and elements within elements, known as nested elements.

Each element is demarcated by a start (open) tag and an end (close) tag.

- `<CRITERIA>` is the start tag for the element named CRITERIA
- `</CRITERIA>` is the end tag for the element named CRITERIA

Nested elements must be properly balanced.

- `<A>` is not properly balanced
 - `<A>` is properly balanced
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

XML Syntax (cont.)

Module 2: Input Control File
Lesson 2-1: XML Syntax and Semantics



The body (content) of an element is the combination of text and nested elements wholly contained within the start and end tag.

Elements may optionally have attributes.

- attribute values must be contained in matching quotes, " " or ' '
- attributes are specified in the start tag of the element as in `<CRITERIA sort="+id">` which defines the attribute value "+id" for the attribute `sort` of the `CRITERIA` element.
- attribute names must be unique within the list of attributes in use by an element

Empty elements are valid and are typically used to convey information via their attributes.

- `<APPLY which="none" />` is the empty `APPLY` element with an attribute of `which`

Each file must have exactly one root element within which all other elements are nested.

- it is an error for text to appear after the close tag of the root element

Characters that have special meaning to the XML parser, in particular, `<`, `>`, `"`, `'`, and `&`, must be protected.

Comments in the document are denoted in HTML/XML style.

- `<!-- this is a comment -->`

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

XML Semantics



Module 2: Input Control File
Lesson 2-1: XML Syntax and Semantics

Element names come from a vocabulary (schema).

- the vocabulary describes the element names that are valid and in what contexts they are valid
- the vocabulary describes the attributes that each element can have and optionally provides default values for those attributes
- the vocabulary may also limit the element names so that no other names are valid

Element and attribute names are case sensitive, unlike HTML.

- element names may not contain white-space
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Input File Semantics

Module 2: Input Control File
Lesson 2-2: DataFax Semantics



Using XML terminology, the input file is a *document*.

By convention, element names are upper-case and attribute names are lower-case.

- this may change in a future release

The input control file vocabulary is limited to the defined element names.

- inclusion of unexpected element names causes a parsing error which halts processing before records are retrieved

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Input File Elements



Module 2: Input Control File
Lesson 2-2: DataFAX Semantics

The document root element is always BATCHLIST.

A BATCHLIST contains one or more BATCH elements.

- most users will generally define one batch per document

Each BATCH element is identified by a unique value for the required name attribute.

- note that there is no element or attribute that identifies a study number
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Input File Elements (cont.)



Module 2: Input Control File
Lesson 2-2: DataFAX Semantics

Input file elements serve one of four purposes:

- descriptive
- processing directive
- action
- record selection

Descriptive Elements:

- these elements are present for identification and documentation purposes only
- TITLE and DESC

Processing Directive Elements:

- these elements allow the user to modify the behaviour and limits of the processing system
- CONTROL and MOVETO

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Input File Elements (cont.)



Module 2: Input Control File
Lesson 2-2: DataFax Semantics

Action Elements:

- these elements are used to specify to the processing system what actions are to be taken as edit checks are executed
- ACTION, APPLY, LOG, and ODRF

Record Selection Elements:

- these elements are used to select from the database the records that are to be processed
 - CRITERIA, IDRF, ID, PLATE, VISIT, CREATE, MODIFY, LEVEL, STATUS, and EDIT
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

<http://www.datafax.com>

Batch Edit Checks

Copyright © Clinical DataFax Systems Inc.

Notes:

Example Input File



Module 2: Input Control File
Lesson 2-2: DataFAX Semantics

```
<?xml version="1.0"?>
<BATCHLIST version="1.0">
  <BATCH name="batch1">
    <TITLE>Example batch</TITLE>
    <DESC>This batch logs all edit check actions
    for the aeCode edit check on records at least
    at level 3.
    </DESC>
    <ACTION>
      <APPLY which="none"/>
      <LOG which="data msg qc" history="yes"
      mode="write" file="batch1_out.xml"
      share="no" when="changes"/>
    </ACTION>
    <CRITERIA>
      <LEVEL include="3-7"/>
      <EDIT>aeCode</EDIT>
    </CRITERIA>
  </BATCH>
</BATCHLIST>
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFAX Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFAX Systems Inc.

Notes:

BATCHLIST, BATCH, TITLE and DESC



Module 2: Input Control File
Lesson 2-3: Element Reference

```
<BATCHLIST version="1.0">  
  <BATCH name="required_name">  
    <TITLE>optional title of batch</TITLE>  
    <DESC>optional description of batch</DESC>  
    ...  
  </BATCH>  
</BATCHLIST>
```

- the name attribute for BATCH is required
 - multiple BATCHES may appear in one input file, in which case the value of the name attribute for each must be unique
 - the version of BATCHLIST must be 1.0 - future releases may change this
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

ACTION and descendants



Module 2: Input Control File
Lesson 2-3: Element Reference

```
<ACTION>
  <APPLY when="changes|all" level="1|2|3|4|5|6|7"
  which="none msg data qc"/>
  <LOG when="changes|all" which="none msg data qc" file="batchname_out.xml"
  mode="create|write" share="yes|no" history="yes|no" />
  <ODRF when="changes|all" which="none msg data qc" file="batchname.drf"
  mode="create|write" share="yes|no"/>
</ACTION>
```

Shared attributes:

- which controls which parts of changes the action is applied to
 - attribute values may be combined
 - msg catches all dferror(), dfwarning(), and dfmessage() results
 - data catches all assignments to database variables
 - qc catches all dfaddqc(), dfaddmpqc(), and dfdelmpqc() results
- when controls whether the action occurs always, when="all", or only when a change occurs, when="changes"
 - for ACTION, every selected record is sent back to the database, even when it had no changes, if when="all"
 - for LOG, every edit check is logged if when="all", otherwise only those edit checks that do something loggable are logged
 - for ODRF, every selected record is written to the DRF, even when it had no changes, if when="all"

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

ACTION and descendants (cont.)



Module 2: Input Control File
Lesson 2-3: Element Reference

APPLY controls what actions are applied back to the database.

- `level` indicates the validation level to assign to records when they are written back to the database

LOG controls what actions are logged.

- `file` specifies the name of the log file, relative to the source file
- `mode` indicates whether the file must be newly created, `create`, (in which case it is an error for a file with that name to already be present), or whether it can simply be written, `write`
- `share` specifies whether others in the same group can read and write the log file
- `history` controls whether the log file marks differences in the log relative to the last execution

ODRF controls what actions cause DataFax Retrieval File records to be written.

- `file` specifies the name of the DRF, relative to the study work directory
- `mode` indicates whether the file must be newly created, `create`, (in which case it is an error for a file with that name to already be present), or whether it can simply be written, `write`
- `share` specifies whether others in the same group can read and write the DRF

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

CRITERIA and descendants



Module 2: Input Control File
Lesson 2-3: Element Reference

```
<CRITERIA sort="+id;+visit;+plate;-img">
  <PATIENT include="list"/>
  <VISIT include="list"/>
  <PLATE include="list"/>
  <LEVEL include="list"/>
  <STATUS include="list"/>
  <CREATE include="list"/>
  <MODIFY include="list"/>
  <IDRF file="input.drf"/>
  <EDIT>ecname1, ecname2, ...</EDIT>
</CRITERIA>
```

The `sort` attribute determines what sort order, if any, is to be applied to records meeting the criteria before processing.

- one or more sort keys from the list: `id`, `visit`, `plate`, `img`, may be used
- `+` indicates ascending order, `-` indicates descending order

The `IDRF` element is mutually exclusive of the other elements.

Elements may appear in any order and, except for `EDIT`, each element may appear at most once.

- `EDIT` may list multiple edit check names in one element body, or in multiple element bodies

The attribute value for `include` follows the same semantics as the validation tool's retrieval criteria.

- individual values, ranges of values, or any combination thereof are valid
- `today` is a legal value for the `CREATE` and `MODIFY` elements

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Module 3 - Processing Batch Edit Checks



-
- Introduction - Overview
 - Lesson 1 - Batch Edit Checks Invocation
-

CDSI::DFUG::BATCHEBITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Overview



Module 3: Processing Batch Edit Checks

Processing of batch input files, from the command-line or off-line environment, is very straightforward.

Post-processing can be performed as part of the batch process, or separately as a subsequent step.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Command-line Invocation



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

The program that does the processing of batch input files is named DFbatch.

The simplest way to invoke DFbatch from the command-line is with the command:

DFbatch study inputfile

- this processes, in file order, all of the BATCHES that appear in inputfile
- the study number and input file arguments are both required and must appear in the order shown

The exit status of DFbatch is 0 if the program executed successfully.

- when running DFbatch several times sequentially through a series of input files, check the exit status of each run before continuing to the next

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Scheduled, Unattended Invocation



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

UNIX has the built-in `cron` program for scheduling and executing programs.

To use `cron` with `DFbatch`, simply include the needed `DFbatch` command-line equivalents in the `crontab`.

```
0 23 * * * DATAFAX_DIR=/opt/datafax; \  
export DATAFAX_DIR; $DATAFAX_DIR/bin/DFbatch study inputfile
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Processing subsets of batches



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

Remember that the root element, BATCHLIST, of the input file can contain many batches.

The command:

```
DFbatch -b simple 254 test_in.xml
```

processes only the batch named `simple` from the input file, independent of how many other batches are defined.

- if the input file defines only the batch named `simple` then execution of `DFbatch` with and without `-b simple` are equivalent

If the input file defines three BATCHES named `simple`, `hard`, and `difficult`, the command:

```
DFbatch -b "simple difficult" 254 test_in.xml
```

would process the two batches `simple` and `difficult` (in that order) while skipping the batch named `hard`.

- when batches are selected with `-b`, they are always processed in the order that they appear in the argument, independent of file order

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Post-processing batch log files



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

Post-processing is relevant only if the batch definition in the input file uses a `<LOG file="batch_out.xml"/>` element.

- there is no other information to post-process if logging has not been enabled

Simplest invocation is with the command:

```
DFbatch -p xsl study inputfile > htmlfile
```

which invokes a subsequent XSL processor to generate an HTML view, in htmlfile, of the log information.

- the log file defined in the `<LOG file="..." />` is not affected by this post-processing, and so post-processing can subsequently be repeated to achieve different views of the same log
- the `-p` and `-b` options can be used together, they are not mutually exclusive

This applies the default stylesheet for a BATCHLOG document root element to the log file, creating HTML output.

- stylesheets are not limited to creating HTML output, although this is their most common use

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Post-processing batch log files (cont.)



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

To apply a non-default stylesheet to the log file at batch processing time, use the command:

```
DFbatch -p XSL=name study inputfile > htmlfile
```

which applies the stylesheet named name to the log file.

- Clinical DataFax Systems Inc will provide customers with several stylesheets
 - you can add your own, or modify ours, without fear of incompatibility
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Post-processing batch log files (cont.)



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

Stylesheet naming is done via the shared file `$DATAFAX_DIR/lib/stylesheets.xml`, which is itself an XML file (using a different vocabulary).

```
<?xml version='1.0'?>
<stylesheetlist version="1.0">
  <stylesheet dtd="BATCHLOG" version="1.0">
    <name>Grouped Chronologically</name>
    <src>lib/xsl/batchlog.xsl</src>
  </stylesheet>
  <stylesheet dtd="BATCHLOG" version="1.0">
    <name>Grouped by Edit Check</name>
    <src>lib/xsl/byeditcheck.xsl</src>
  </stylesheet>
</stylesheetlist>
```

The `dtd` attribute of the `stylesheet` element specifies the document type for which the stylesheet is relevant.

It is valid, and expected, that there be more than one stylesheet associated with any given DTD.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Re-styling log files independent of creation



Module 3: Processing Batch Edit Checks
Lesson 3-1: Batch Edit Checks Invocation

The DFstyle program can be invoked at any time to apply a stylesheet to an XML file with the following command:

```
DFstyle -p xsl infile > outfile
```

which applies the default stylesheet for the input file's vocabulary, or

```
DFstyle -p XSL=name infile > outfile
```

which applies the named stylesheet.

- infile can be any XML file, although in the context of batch edit checks, it is an existing log file from a previous execution
- the study number argument is not needed for DFstyle
- the default stylesheet for any dtd is the first one, by file position, that appears in \$DATAFAX_DIR/lib/stylesheet.xml.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Module 4 - Batch Edit Checks Output



·Lesson 1 - Log Output File

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

XML format



Module 4: Batch Edit Checks Output
Lesson 4-1: Log Output File

The log information from a batch execution is itself in XML format.

The log includes all of the information needed to give a complete and accurate view of what occurred.

The log contains elements that identify:

- the input file,
- the study,
- the user who executed the batch,
- the history of execution,
- the records processed,
- the edit checks applied at each field, and
- the action(s) of each edit check.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

XML format (cont.)

Module 4: Batch Edit Checks Output
Lesson 4-1: Log Output File



Log files can be very large and so, in the interest of space-saving, the element names of the output vocabulary are significantly less descriptive than in the input vocabulary.

- this does not however diminish the effectiveness of the output log as the element names, while terse, have a fixed meaning that can be utilized during post-processing

```
<R>
<K i='45678' v='1' p='2' />
<A s='2' l='1' im='0013R0014001' />
<V n='PINIT002'>
  <E w='fn' n='SetInit'>
    <M fr='1' t='w'>Initials EGB and * differ.</M>
  </E>
</V>
<V n='WTLB'>
  <E w='fx' n='WeightEquiv'>
    <Q fr='2' c='1' u='1' f='2' p='1'>
      <V n='WTLB' />
      <QR>Please provide weight in lbs or kgs</QR>
    </Q>
  </E>
</V>
...
</R>
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

XML format (cont.)

Module 4: Batch Edit Checks Output
 Lesson 4-1: Log Output File



Subsequent to post-processing, this log information might have the following appearance:

Keys: Id, Visit, Plate	Variable	Log Entry
1. 45678, 1, 2 2, 1, 0013R0014001	<u>PINIT002</u> <i>Setinit</i>	a. dfwarning Initials EGB and * differ.
	<u>WTLB</u> <i>WeightEquiv</i>	a. Add QC External, Fax/Refax QC added for missing value to: WTLB <u>Query</u> : Please provide weight in lbs or kgs

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
 Batch Edit Checks

<http://www.datafax.com>
 Copyright © Clinical DataFax Systems Inc.

Notes:

XML format (cont.)



Module 4: Batch Edit Checks Output
Lesson 4-1: Log Output File

The log file uses relations between elements that represent the history of execution and the action elements.

This batch was executed at 11:37:07 on 2000-11-09 (as well as other times)

```
<HN fd='2'>  
<DT><YY>2000</YY><MM>11</MM><DD>09</DD></DT>  
<TM><HR>11</HR><MI>37</MI><SC>07</SC></TM>  
</HN>
```

During that execution this QC note was first added (and has been added with every batch execution since).

```
<Q fr='2' c='1' u='1' f='2' p='1'>  
<V n='WTLB' />  
<QR>Please provide weight in lbs or kgs</QR>  
</Q>
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Module 5 - Implementation Considerations



-
- Introduction - Overview
 - Lesson 1 -
 - Lesson 2 - Defaults for interactive functions
 - Lesson 3 - What is not possible
 - Lesson 4 - What is discouraged
 - Lesson 5 - Caveats
 - Lesson 6 - Recommended Usage

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Overview



Module 5: Implementation Considerations

Before deploying batch edit checks in your production environment, consider the following:

- how will the results be reviewed
- behaviour of interactive functions
- what cannot be done
- what should not be done
- caveats and open issues

Some recommendations for usage are included.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Web publishing of batch edit checks logs



Module 5: Implementation Considerations
Lesson 5-1:

The logs have lots of useful information in them, so you want to encourage those responsible to review the logs.

- make it easy to review the logs.
 - automatically style the logs to HTML and publish to a known intranet location.
 - build a small intranet for your DataFax studies.
-

CDSI::DFUG::BATCHEBITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Default return values



Module 5: Implementation Considerations
Lesson 5-2: Defaults for interactive functions

Certain features of the edit checks language are intrinsically interactive.

When these features are encountered in DFbatch, they must be handled intelligently and consistently.

Functions that are not listed below behave identically in both environments.

`dfask(query, dflt, accept, cancel)` always returns `dflt`.

`dflookup(table, var, dflt, method)` always return `dflt` when `method` has any value other than `-1`.

- if `method=-1`, `dflookup` behaves the same in both environments, returning the result from `table` if an exact match can be made, `dflt` otherwise.
- disregarding this behaviour, as in the following edit check, can lead to disastrous results

```
string s;

s = dflookup( "TABLE", @(T-1), "", -1 );
if ( s != "" )
    @T = s;
else
    @T = dflookup( "TABLE", @(T-1), "", 0 );
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Default return values (cont.)



Module 5: Implementation Considerations
Lesson 5-2: Defaults for interactive functions

Revised implementation of `dflookup` that is sensitive to batch edit checks behaviour.

```
# Look-up the value in @(T-1) and store the result in @T

if ( dfbatch() ) {
  # Use the result only if an exact match is available and then
  # only if the field is not already coded
  result = dflookup( "TABLE", @(T-1), "", -1 );
  if ( result == "" )
    return;
  if ( dfblank( @T ) )
    @T = result;
  else if ( result != @T )
    dferror( "Previously coded value of ", @T, " and new result ",
            result, " do not match." );
} else {
  # insert existing dflookup code here
}
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Default return values (cont.)



Module 5: Implementation Considerations
Lesson 5-2: Defaults for interactive functions

`dfillegal(@T)` always returns 0.

- setting the illegal field colour is meaningful in the validation tool only

`dfbatch()` always returns 1 in batch, 0 in the validation tool.

The behaviour of `dfaddqc()`, `dfaddmpqc()`, and `dfdelmpqc()` is controlled by the `which` attribute of the `APPLY` element.

The behaviour of `dfmessage()`, `dfwarning()`, and `dferror()` is controlled by the `which` attribute of the `APPLY` or `LOG` elements.

- All other errors that would cause a dialog to appear in interactive edit checks cause a log message to be written to the batch log file, if there is one, or standard error, otherwise.

CDSI::DFUG::BATCHEBITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Not possible with batch edit checks



Module 5: Implementation Considerations
Lesson 5-3: What is not possible

Not possible to change the status of a record.

- a clean record must remain clean - it is not possible to assign an illegal or missing, required value to a data field on a clean record
- one exception: batch edit checks will change the status of a clean record to dirty if one or more QC notes are added to the record

Not possible to change the primary/secondary status of a record.

- there is no opportunity to interact with the dup resolve dialog

Not possible to change the value of any key field.

- the id, visit, and plate numbers cannot be changed, regardless of whether or not those fields are barcoded
- this limits the usefulness, in batch, of edit checks that calculate and assign the visit number of a CRF

Not possible to apply batch edit checks to new (level 0) records, or lost records.

Not possible to set the validation level of any record to 0, or a level higher than permitted by DataFax user permissions.

- this isn't really a limitation of batch edit checks, as this is also not possible in the validation tool

Not possible to enter data that cannot also be entered via the validation tool.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Not recommended with batch edit checks



Module 5: Implementation Considerations
Lesson 5-4: What is discouraged

The FDA's Guidance for Industry: Computerized Systems Used in Clinical Trials explicitly states "Features that automatically enter data into a field when that field is bypassed should not be used"

- although not stated, the spirit of this guidance clearly relates to reported values
 - calculated or derived values that are not part of the source document should be exempt from this
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Issues to remember



Module 5: Implementation Considerations
Lesson 5-5: Caveats

There is no rollback feature.

- once data changes are applied, there is no built-in mechanism for going back
- changes are applied as they occur, and with the ability to process tens of records per second, they occur very quickly

Currently, a user's permissions to execute batch edit checks are the same as their validation tool permissions.

- this will change in DataFax 4 as batch processes are merged into the audit trail and permissions database
-

CDSI::DFUG::BATCHEBITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Tips to Remember



Module 5: Implementation Considerations
Lesson 5-6: Recommended Usage

Instead of replacing reported values with calculated values, issue an error message when the reported value and the expected value do not match.

```
# s contains the calculated value for what should be in @T
s = "some calculated value";
if ( s != @T )
    dferror( "The expected value, ", s,
            ", and the reported value, ", @T,
            ", do not match." );
```

Exclude secondary records from record selection criteria.

- always use `<STATUS include="primary" />` in the CRITERIA section of the input file
- whether they are included or not, secondary records are never involved in QC note actions

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Tips to Remember (cont.)



Module 5: Implementation Considerations
Lesson 5-6: Recommended Usage

When selecting records with the PLATE and EDIT, remember that records must satisfy both criteria.

```
<CRITERIA>  
  <PLATE include="10-20"/>  
  <EDIT>checkEligibility</EDIT>  
</CRITERIA>
```

Test your edit checks heavily in the validation tool before using them in batch.

- remember that in the 10-20 seconds required to validate a record interactively, a batch will process hundreds of records

Test your edit checks in batch.

- the best way to do this is with extensive use of `<LOG which="data msg qc".../>` and `<APPLY which="none"/>`
 - review the logs very carefully
 - subsequently, and if appropriate, switch to `<APPLY which="data msg qc".../>`
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Tips to Remember (cont.)



Module 5: Implementation Considerations
Lesson 5-6: Recommended Usage

Create a `batch` sub-directory below the study directory and keep your input and output files in that directory.

- keep the styled output in an intranet accessible directory

Implement a naming convention and follow-it.

- use `.xml` extension for all input and output files. This default extension allows you to leverage other XML tools.
- consider using `_in` and `_out` suffixes in the input and output file names, respectively. For example, `batchAes_in.xml` and `batchAes_out.xml`.

Avoid batch input files that execute all edit checks on all records.

- this will lead to very large log files that are difficult to post-process and efficiently apply history to
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Module 6 - Complete Example



-
- Introduction - Overview
 - Lesson 1 - Create, execute, and review a batch
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Overview



Module 6: Complete Example

The purpose of the example is to increase familiarity by:

- seeing an input file,
 - applying the input file to a small database, and
 - reviewing the process results via a web browser.
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Create Input File



Module 6: Complete Example
Lesson 6-1: Create, execute, and review a batch

Create an input file that applies all edit checks to level 1, primary records.

Apply no changes to the database.

Log potential changes only to a log file, `example_out.xml`

Audit all potentially changed records to a DRF, `example.drf`

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

<http://www.datafax.com>

Batch Edit Checks

Copyright © Clinical DataFax Systems Inc.

Notes:

One Possible Input File



Module 6: Complete Example

Lesson 6-1: Create, execute, and review a batch

```
<?xml version="1.0"?>
<BATCHLIST version="1.0">
  <BATCH name="example">

    <TITLE>Example input file</TITLE>
    <DESC>This batch applies all edit checks to existing
    level 1 records, and logs the results only.</DESC>

    <ACTION>
      <!-- ensure that no changes are made -->
      <APPLY which="none"/>

      <!-- log everything to a shareable file -->
      <LOG which="data msg qc" when="changes"
        file="example_out.xml" mode="write"
        share="yes" history="yes"/>

      <!-- record all to a DRF as well -->
      <ODRF which="data msg qc" when="changes"
        file="example.drf" mode="write"
        share="yes"/>
    </ACTION>

    <CRITERIA sort="+id;+visit;+plate">
      <LEVEL include="1"/>
      <STATUS include="primary"/>
    </CRITERIA>
  </BATCH>
</BATCHLIST>
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Process Input File



Module 6: Complete Example
Lesson 6-1: Create, execute, and review a batch

The input file is processed with the following command:

```
DFbatch -p xsl 254 example_in.xml > /opt/www/studies/val254/example.html
```

In addition to creating the requested log file, `example_out.xml`, the log file is post-processed with the default XSL transformation to create `example.html`.

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Review Styled Output



Module 6: Complete Example
Lesson 6-1: Create, execute, and review a batch

Using Internet Explorer 5.5 to review the post-processed log file.

Batch Edit Checks Log for example, sorted by reverse chronological order

Environment	Summary	Total	Success	Failure	Truncated
Title: Example for EC Applications course	Records meeting criteria:	13			
Description: This batch applies all edit checks to existing level 1 records, and logs the results only.	Records logged:	7			
Study: 254	System messages:	0			
Started on: 2001-01-30 09:55:15	Data changes:	24	19	0	5
Performed by: eric	Messages:	2	2		
Working Directory: /opt/studies/vai254/batch	QC notes:	1	1	0	
Control File: example_in.xml	Missing page QC notes:	0	0	0	
DRF: /opt/studies/vai254/work/example.drf	Missing page QC notes deleted:	0	0	0	
Log File: /opt/studies/vai254/batch/example_out.xml	Elapsed time (seconds):	0			

Execution History

Date	# of Log Entries	Date	# of Log Entries
2001-01-30 09:55:15 (current)	22		

2001-01-30 09:55:15 (current)

Keys: Id, Visit, Plate, Meta: Status, Level, CRF	Variable Edit Check	Log Entry
1. 99002, 1, 2 1, 1, 9807/0024003	BDATE CheckAge	dferror AGE ERROR: Birth Date = 05/12/50 Entry Date = 08/01/97 Calculated age = 46. Age on Form 1 = 64.
2.		Add QC External, Fax/Refax QC added for inconsistent value to: BDATE Query:

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Review Styled Output (cont.)



Module 6: Complete Example
 Lesson 6-1: Create, execute, and review a batch

Using Netscape 6 to review the post-processed log file.

Batch Edit Checks Log for example, sorted by reverse chronological order

Environment		Summary			
Title:	Example for EC Applications course	Total	Success	Failure	Truncated
Description:	This batch applies all edit checks to existing level 1 records, and logs the results only.	Records meeting criteria:	13		
Study:	254	Records logged:	7		
Started on:	2001-01-30 09:55:15	System messages:	0		
Performed by:	enc	Data changes:	24	19	0
Working Directory:	/opt/studies/val254/batch	Messages:	2	2	0
Control File:	example_in.xml	QC notes:	1	1	0
DRF:	/opt/studies/val254/work/example.drf	Missing page QC notes:	0	0	0
Log File:	/opt/studies/val254/batch/example_out.xml	Missing page QC notes deleted:	0	0	0
		Elapsed time (seconds):	0		

Execution History			
Date	# of Log Entries	Date	# of Log Entries
2001-01-30 09:55:15 (current)	22		

2001-01-30 09:55:15 (current)

Keys: Id, Visit, Plate	Variable	Log Entry
Meta: Status, Level, CRF	Edit Check	
1. 99002, 1, 2 1, 1, 9807/0024003	BDATE Check Age	dferror AGE ERROR: Birth Date = 05/12/50 Entry Date = 08/01/97 Calculated age = 46. Age on Form 1 = 64.
2.		Add QC External, Fax/Refax QC added for inconsistent value to: BDATE Query: Birth date does not match age on Form 1

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFAX Systems Inc. - <http://www.datafax.com>
 Batch Edit Checks

<http://www.datafax.com>
 Copyright © Clinical DataFAX Systems Inc.

Notes:

Module 7 - Creating Your Own Stylesheet(s)



-
- Lesson 1 - What Options Are Available
 - Lesson 2 - Implementing your own XSLT stylesheet
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Non-XSLT Solutions



Module 7: Creating Your Own Stylesheet(s)
Lesson 7-1: What Options Are Available

XML+CSS

- understood by most modern browsers, including Netscape, Internet Explorer, Opera, Amaya
- linear styling and some filtering
- no transformation, aggregation

XML::Perl

- similar capabilities to XML+CSS
- the power of Perl

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

XSLT Solutions

Module 7: Creating Your Own Stylesheet(s)
Lesson 7-1: What Options Are Available



Roll your own stylesheet

Customize one of ours

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Create the stylesheet



Module 7: Creating Your Own Stylesheet(s)

Lesson 7-2: Implementing your own XSLT stylesheet

```
<xsl:include href="lib/xsl/batchlog.xsl" />
```

- inherit all templates from named file
- create own templates and templates to override inherited templates

```
<xsl:import href="lib/xsl/batchlog.xsl" />
```

- create own templates
 - instantiate imported templates when needed using `<xsl:apply-imports/>`
-

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes:

Create the stylesheet (cont.)



Module 7: Creating Your Own Stylesheet(s)
Lesson 7-2: Implementing your own XSLT stylesheet

This stylesheet, `test.xsl`, re-formats the appearance of the record's key fields, so that each key is identified on a separate line, and the presentation order is different.

```
<?xml version="1.0"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">

<xsl:include href="/opt/datafax/lib/xsl/batchlog.xsl"/>

<xsl:template name="PRINT_KEYS">
  <SPAN class="keys">
    ID: <xsl:value-of select="@i"/><BR />
    PLATE: <xsl:value-of select="@p"/><BR />
    VISIT: <xsl:value-of select="@v"/><BR />
  </SPAN>
</xsl:template>

</xsl:stylesheet>
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Add the stylesheet to the registry



Module 7: Creating Your Own Stylesheet(s)
Lesson 7-2: Implementing your own XSLT stylesheet

Edit \$DATAFAX_DIR/lib/stylesheet.xml

- Identify the stylesheet with a unique name
- Specify the physical location of the stylesheet
- Remember that position in the file implies priority

```
<?xml version='1.0'?>
<stylesheetlist version="1.0">
  <stylesheet dtd="BATCHLOG" version="1.0">
    <name>Grouped Chronologically</name>
    <src>lib/xsl/batchlog.xsl</src>
  </stylesheet>
  <stylesheet dtd="BATCHLOG" version="1.0">
    <name>Grouped by Edit Check</name>
    <src>lib/xsl/byeditcheck.xsl</src>
  </stylesheet>
  <stylesheet dtd="BATCHLOG" version="1.0">
    <name>Test with explicit keys</name>
    <src>/opt/studies/val254/lib/test.xsl</src>
  </stylesheet>
</stylesheetlist>
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40/EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks

<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.

Notes:

Re-style one or more log files



Module 7: Creating Your Own Stylesheet(s)
Lesson 7-2: Implementing your own XSLT stylesheet

Note that we are using `DFstyle` to re-style without changing the log output.

- Replacing `DFstyle` with `DFbatch` (and including the study number) would re-execute the batch AND re-style that resulting log file.

```
DFstyle -p XSL="Test with explicit keys" \  
/opt/studies/val254/batch/example_out.xml > \  
/opt/www/studies/val254/example.html
```

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

*Clinical DataFax Systems Inc. - <http://www.datafax.com>
Batch Edit Checks*

*<http://www.datafax.com>
Copyright © Clinical DataFax Systems Inc.*

Notes:

Re-style one or more log files (cont.)



Module 7: Creating Your Own Stylesheet(s)
 Lesson 7-2: Implementing your own XSLT stylesheet

Displaying the re-styled results in Internet Explorer 5.5.

Batch Edit Checks Log for example, sorted by reverse chronological order

Environment		Summary			
Title:	Example for EC Applications course	Total	Success	Failure	Truncated
Description:	This batch applies all edit checks to existing level 1 records, and logs the results only.	Records meeting criteria:	13		
Study:	254	Records logged:	7		
Started on:	2001-01-30 09:55:15	System messages:	0		
Performed by:	eric	Data changes:	24	19	0
Working Directory:	/opt/studies/vai254/batch	Messages:	2	2	0
Control File:	example_in.xml	QC notes:	1	1	0
DRF:	/opt/studies/vai254/work/example.drf	Missing page QC notes:	0	0	0
Log File:	/opt/studies/vai254/batch/example_out.xml	Missing page QC notes deleted:	0	0	0
		Elapsed time (seconds):	0		

Execution History			
Date	# of Log Entries	Date	# of Log Entries
2001-01-30 09:55:15 (current)	22		

2001-01-30 09:55:15 (current)

Keys: Id, Visit, Plate Meta: Status, Level, CRF	Variable Edit Check	Log Entry
1. ID: 99002 PLATE: 2 VISIT: 1 1, 1, 9807/0024003	BDATE CheckAge	dferror AGE ERROR: Birth Date = 05/12/50 Entry Date = 08/01/97 Calculated age = 46. Age on Form 1 = 64.
2.		Add QC External, Fax/Rafax QC added for inconsistent value to: BDATE Query:

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>
 Batch Edit Checks

<http://www.datafax.com>
 Copyright © Clinical DataFax Systems Inc.

Notes:

Batch Edit Checks



- 01: Batch Edit Checks (Prelude)
- 02: Batch Edit Checks
- 03: Batch Edit Checks
- 04: Module 1 - What are Batch Edit Checks?
- 05: Overview
- 06: Loose Coupling
- 07: Limits
- 08: Re-use existing edit checks
- 09: Execute non-interactively
- 10: Log everything
- 11: Input Control File
- 12: Processing Environment
- 13: Output Log File
- 14: Module 2 - Input Control File
- 15: Overview
- 16: XML Syntax
- 17: XML Syntax (cont.)
- 18: XML Semantics
- 19: Input File Semantics
- 20: Input File Elements
- 21: Input File Elements (cont.)
- 22: Input File Elements (cont.)
- 23: Example Input File
- 24: BATCHLIST, BATCH, TITLE and DESC
- 25: ACTION and descendants
- 26: ACTION and descendants (cont.)
- 27: CRITERIA and descendants
- 28: Module 3 - Processing Batch Edit Checks
- 29: Overview
- 30: Command-line Invocation
- 31: Scheduled, Unattended Invocation
- 32: Processing subsets of batches
- 33: Post-processing batch log files
- 34: Post-processing batch log files (cont.)
- 35: Post-processing batch log files (cont.)
- 36: Re-styling log files independent of creation
- 37: Module 4 - Batch Edit Checks Output
- 38: XML format
- 39: XML format (cont.)

- 40: XML format (cont.)
- 41: XML format (cont.)
- 42: Module 5 - Implementation Considerations
- 43: Overview
- 44: Web publishing of batch edit checks logs
- 45: Default return values
- 46: Default return values (cont.)
- 47: Default return values (cont.)
- 48: Not possible with batch edit checks
- 49: Not recommended with batch edit checks
- 50: Issues to remember
- 51: Tips to Remember
- 52: Tips to Remember (cont.)
- 53: Tips to Remember (cont.)
- 54: Module 6 - Complete Example
- 55: Overview
- 56: Create Input File
- 57: One Possible Input File
- 58: Process Input File
- 59: Review Styled Output
- 60: Review Styled Output (cont.)
- 61: Module 7 - Creating Your Own Stylesheet(s)
- 62: Non-XSLT Solutions
- 63: XSLT Solutions
- 64: Create the stylesheet
- 65: Create the stylesheet (cont.)
- 66: Add the stylesheet to the registry
- 67: Re-style one or more log files
- 68: Re-style one or more log files (cont.)

CDSI::DFUG::BATCHEDITS Batch Edit Checks 2000-01-29 10:40//EN

Clinical DataFax Systems Inc. - <http://www.datafax.com>

Batch Edit Checks

<http://www.datafax.com>

Copyright © Clinical DataFax Systems Inc.

Notes: