NISTIR 5173

NIST Scoring Package Certification Procedures in Conjunction with *NIST Special Databases 2 and 6*

Michael D. Garris

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Advanced Systems Division
Gaithersburg, MD 20899

NIST

NISTIR 5173

NIST Scoring Package Certification Procedures in Conjunction with *NIST Special Databases 2 and 6*

Michael D. Garris

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Advanced Systems Division
Gaithersburg, MD 20899

April 1993

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST Scoring Package Certification Procedures
in Conjunction with *NIST Special Databases 2* and *6*

Michael D. Garris

National Institute of Standards and Technology
Advanced Systems Division
Image Recognition Group

April 14, 1993

1. Introduction

The following procedures have been developed by NIST in order to promote compliance with existing Scoring Package file formats.[1][2] Through certification, the proper use of the Scoring Package is promoted and the successful scoring of recognition system data is maximized. This is true whether it is for conferences such as the First Census Optical Character Recognition Conference[3] or for independent organizational use. The level of effort required to successfully score recognition system results is inversely related to the level of compliance with the certification process outlined here. Not following the details outlined herein will render system results unscoreable. Therefore, NIST strongly encourages Scoring Package certification.

The certification procedures presented in this document have been developed in conjunction with *NIST Special Database 2* (SD2)[4] and *NIST Special Database 6* (SD6)[5]. These two databases contain images of synthesized tax forms. The data entered on the forms appears real, but the values have been generated at random by a computer. NIST offers certification to any organization that has purchased the Scoring Package and requests the service. Requests for certification should be directed to the author. To receive further information related to the Scoring Package, SD2, and SD6, contact the Standard Reference Data Division at NIST.

2. Certification Package

Each organization requesting Scoring Package certification from NIST will be sent a Certification Package. This package is distributed on a data-grade 8mm cartridge tape in tar format at a density of 2.2 Gigabytes per tape. The top level directory on the distribution tape contains the three directories illustrated in Figure 1. The directory **image** contains a collection of form images, the directory **ref** contains reference files (one for each form in **image**), and the directory **util** contains scoring utilities and tables.

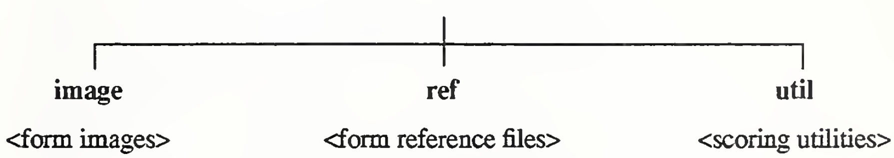

Figure 1: Top level directory of a Certification Package.

2.1 Certification Images

The set of form images in the directory **image** are based on forms distributed in SD2 and SD6. The organization of this directory is illustrated in Figure 2. The set is divided into two subdirectories, **block_0** and **block_1**. The directory **block_0** contains 5 submissions of synthesized tax forms, **r0000** through **r0004**, completed with machine print, and the directory **block_1** contains 5 submissions of synthesized tax forms, **r0005** through **r0009**, completed with hand print. Each submission contains up to 6 form face types from the 1988 IRS Package X: 1040 page 1, 1040 page2, Schedule A, Schedule B, 4562 page 1, and 4562 page 2. There are a total of 50 form images distributed across the 10 submission subdirectories. Each form image file name has as its base the name of the submission followed by a two-digit page index separated by an underscore, and the file name ends with the extension ".pct".

For example, the first form image in submission **r0003** has the file name **r0003_00.pct**, whereas the second form image in the submission has the file name **r0003_01.pct**.

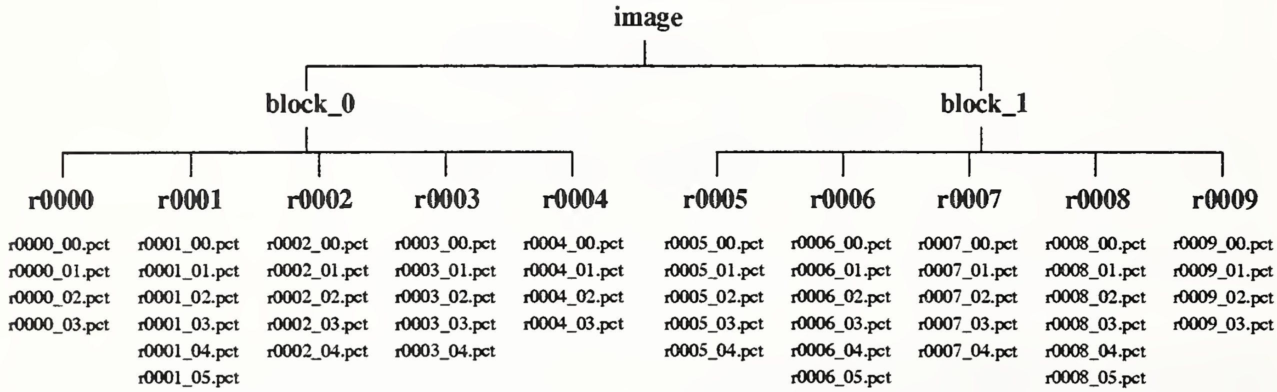

Figure 2: Directory hierarchy for the Certification Package directory image.

2.1.1 Image File Format

Each form image in the Certification Package is a completed tax form synthesized at 300 dots per inch binary, 2-dimensionally compressed using CCITT Group 4[6][7], and stored in the IHead format[4][5]. An IHead file is a raster image with a prefixed ASCII header. The 2-dimensional area of the form is divided into discrete locations according to the resolution of a specified grid. Each cell of this grid is represented by a single bit value 0 or 1 called a pixel; 0 represents a cell predominately white, 1 represents a cell predominately black. This 2-dimensional sampling grid is then stored as a 1-dimensional vector of pixel values in raster order, left to right, top to bottom. Successive scan lines (top to bottom), contain the values of a single row of pixels from the grid concatenated together.

Certain attributes of an image are required to be known to correctly interpret the 1-dimensional pixel data as a 2-dimensional image. Examples of such attributes are the pixel width and pixel height of the image. These attributes are stored in a machine readable ASCII header prefixed to the raster bit stream. A program used to manipulate the raster data of an image is able to first read the header and determine the proper interpretation of the data which follows it. Figure 3 illustrates this file format.

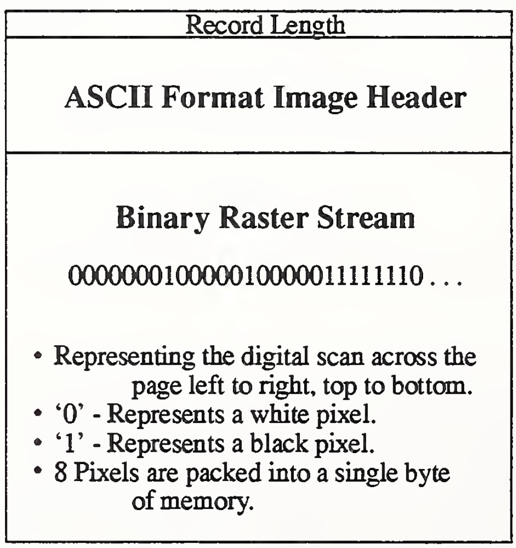

Figure 3: An illustration of the IHead raster file format.

The IHead header format has been developed for use as an image interchange format. Numerous image formats exist; some are widely supported on small personal computers, others supported on larger workstations; most are proprietary formats, few are public domain. The IHead header is a public domain image format that can be universally implemented across heterogeneous computer architectures and environments. Both documentation and source code for the IHead format are publicly available and included with SD2 and SD6. IHead has been designed with an extensive set of attributes in order to adequately represent both binary and gray level images, to represent images captured from different scanners and cameras, and to satisfy the image requirements of diversified applications including, but not limited to, image archiving/retrieval, character recognition, and fingerprint classification.

IHead has been successfully ported and tested on several systems including UNIX workstations and servers, DOS personal computers, and VMS mainframes. The attribute fields in IHead can be loaded into main memory in two distinct ways. Since the attributes are represented by the ASCII character set, the attribute fields may be parsed as null-terminated strings, an input/output format common in the 'C' programming language. IHead can also be read into main memory using record-oriented input/output. The fixed length of the header is prefixed to the front of the header as shown in Figure 3. The IHead structure definition as written in the 'C' programming language is listed in Figure 4.

```
/***********************************************************
        File Name: IHead.h
        Package:  NIST Internal Image Header
        Author:   Michael D. Garris
        Date:     2/08/90
***********************************************************/
/* Defines used by the ihead structure */
#define IHDR_SIZE         288     /* len of hdr record (always even bytes) */
#define SHORT_CHARS       8       /* # of ASCII chars to represent a short */
#define BUFSIZE           80      /* default buffer size */
#define DATELEN           26      /* character length of data string */

typedef struct ihead{
  char id[BUFSIZE];                  /* identification/comment field */
  char created[DATELEN];             /* date created */
  char width[SHORT_CHARS];           /* pixel width of image */
  char height[SHORT_CHARS];          /* pixel height of image */
  char depth[SHORT_CHARS];           /* bits per pixel */
  char density[SHORT_CHARS];         /* pixels per inch */
  char compress[SHORT_CHARS];        /* compression code */
  char complen[SHORT_CHARS];         /* compressed data length */
  char align[SHORT_CHARS];           /* scanline multiple: 8|16|32 */
  char unitsize[SHORT_CHARS];        /* bit size of image memory units */
  char sigbit;                       /* 0->sigbit first | 1->sigbit last */
  char byte_order;                   /* 0->highlow | 1->lowhigh*/
  char pix_offset[SHORT_CHARS];      /* pixel column offset */
  char whitepix[SHORT_CHARS];        /* intensity of white pixel */
  char issigned;                     /* 0->unsigned data | 1->signed data */
  char rm_cm;                        /* 0->row maj | 1->column maj */
  char tb_bt;                        /* 0->top2bottom | 1->bottom2top */
  char lr_rl;                        /* 0->left2right | 1->right2left */
  char parent[BUFSIZE];              /* parent image file */
  char par_x[SHORT_CHARS];           /* from x pixel in parent */
  char par_y[SHORT_CHARS];           /* from y pixel in parent */
}IHEAD;
```

Figure 4: IHead 'C' programming language definition.

Figure 5 lists the header values from an IHead file corresponding to the structure members listed in Figure 4. This header information belongs to the isolated box image displayed in Figure 6. Referencing the structure members listed in Figure 4, the first attribute field of IHead is the identification field, **id**. This field uniquely identifies the image file, typically by a file name. The identification field in this example not only contains the image's file name, but also the reference string the writer was instructed to print in the box. The reference string is delimited by double quotes.

IMAGE FILE HEADER
~~~~~~~~~~~~~~~~~~

Identity	: box_03.pct "0123456789"
Header Size	: 288 (bytes)
Date Created	: Thu Jan 4 17:34:21 1990
Width	: 656 (pixels)
Height	: 135 (pixels)
Bits per Pixel	: 1
Resolution	: 300 (ppi)
Compression	: 2 (code)
Compress Length	: 874 (bytes)
Scan Alignment	: 16 (bits)
Image Data Unit	: 16 (bits)
Byte Order	: High-Low
MSBit	: First
Column Offset	: 0 (pixels)
White Pixel	: 0
Data Units	: Unsigned
Scan Order	: Row Major, Top to Bottom, Left to Right
Parent	: hsf_0/f0000_14/f0000_14.pct
X Origin	: 192 (pixels)
Y Origin	: 732 (pixels)

Figure 5: The IHead values for the isolated subimage displayed in Figure 6.

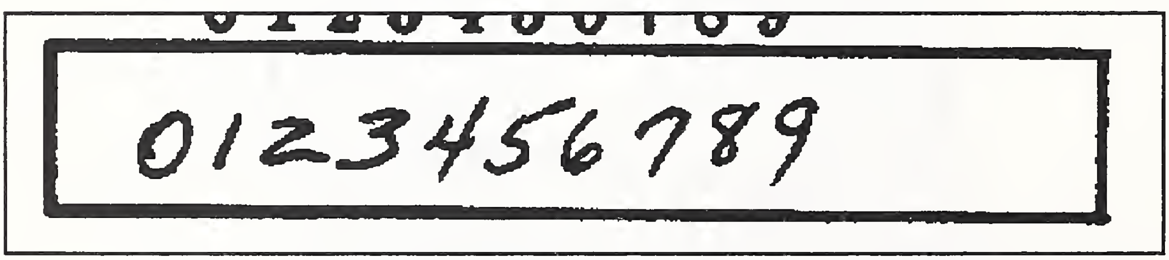

Figure 6: An IHead image of an isolated box.

The attribute field, **created**, is the date on which the image was captured or digitized. The next three fields hold the image's pixel **width**, **height**, and **depth**. A binary image has a pixel depth of 1 whereas a gray scale image containing 256 possible shades of gray has a pixel depth of 8. The attribute field, **density**, contains the scan resolution of the image; in this case, 300 dots per inch. The next two fields deal with compression.
~~~~~~~~~~~~~~~~~~

In the IHead format, images may be compressed with virtually any algorithm. The header is always uncompressed, even if the image data is compressed. This enables header interpretation and manipulation without the overhead of decompression. The **compress** field is an integer flag that signifies which compression technique, if any, has been applied to the raster image data following the header. If the compression code is zero, then the image data is not compressed, and the data dimensions: width, height, and depth, are sufficient to load the image into main memory. However, if the compression code is nonzero, then the **complen** field must be used in addition to the image's pixel dimensions. For example, the image described in Figure 5 has a compression code of 2. This signifies that CCITT Group 4 compression has been applied to the image data prior to file creation. In order to load the compressed image data into main memory, the value in **complen** is used to load the compressed block of data. Once the compressed image data has been loaded into memory, CCITT Group 4 decompression can be used to produce an image which has the pixel dimensions consistent with those stored in its header.

The attribute field, **align**, stores the alignment boundary to which scan lines of pixels are padded. Pixel values of binary images are stored 8 pixels (or bits) to a byte. Most images, however, are not an even multiple of 8 pixels in width. To minimize the overhead of ending a previous scan line and beginning the next scan line within the same byte, a number of pixels are provided in order to extend the previous scan line to an even byte boundary. Some digitizers extend this padding of pixels out to an even multiple of 8 pixels, other digitizers extend this padding of pixels out to an even multiple of 16 pixels. This field stores the image's pixel alignment value used in padding out the ends of raster scan lines.

The next three attribute fields identify binary interchanging issues among heterogeneous computer architectures and displays. The **unitsize** field specifies how many contiguous pixel values are bundled into a single unit by the digitizer. The **sigbit** field specifies the order in which bits of significance are stored within each unit; most significant bit first or least significant bit first. The last of these three fields is the **byte_order** field. If **unitsize** is a multiple of bytes, then this field specifies the order in which bytes occur within the unit. Given these three attributes, binary incompatibilities across computer hardware and binary format assumptions within application software can be identified and effectively dealt with.

The **pix_offset** attribute defines a pixel displacement from the left edge of the raster image data to where a particular image's significant image information begins. The **whitepix** attribute defines the value assigned to the color white. For example, the binary image described in Figure 5 is black text on a white background and the value of the white pixels is 0. This field is particularly useful to image display routines. The **issigned** field is required to specify whether the units of an image are signed or unsigned. This attribute determines whether an image with a pixel depth of 8 should have pixel values interpreted in the range of -128 to +127, or 0 to 255. The orientation of the raster scan may also vary among different digitizers. The attribute field, **rm_cm**, specifies whether the digitizer captured the image in row-major order or column-major order. Whether the scan lines of an image were accumulated from top to bottom, or bottom to top, is specified by the field, **tb_bt**, and whether left to right, or right to left, is specified by the field, **rl_lr**.

The final attributes in IHead provide a single historical link from the current image to its parent image; the one from which the current image was derived or extracted. In Figure 5, the **parent** field contains the full path name to the image from which the image displayed in Figure 6 was extracted. The **par_x** and **par_y** fields contain the origin point (upper left corner pixel coordinate) from where the extraction took place in the parent image. These fields provide a historical thread through successive generations of images and subimages. The IHead image format contains the minimal amount of ancillary information required to successfully manage binary and gray scale images.

2.2 Certification Reference Files

As can be seen in Figure 7, the directory **ref** contains the identical directory structure found in **image** shown in Figure 2, with the exception that form image files are replaced with form reference files ending with the extension ".fmt". The reference files are required by the Scoring Package to process recognition system results generated from the form images in the directory **image**.

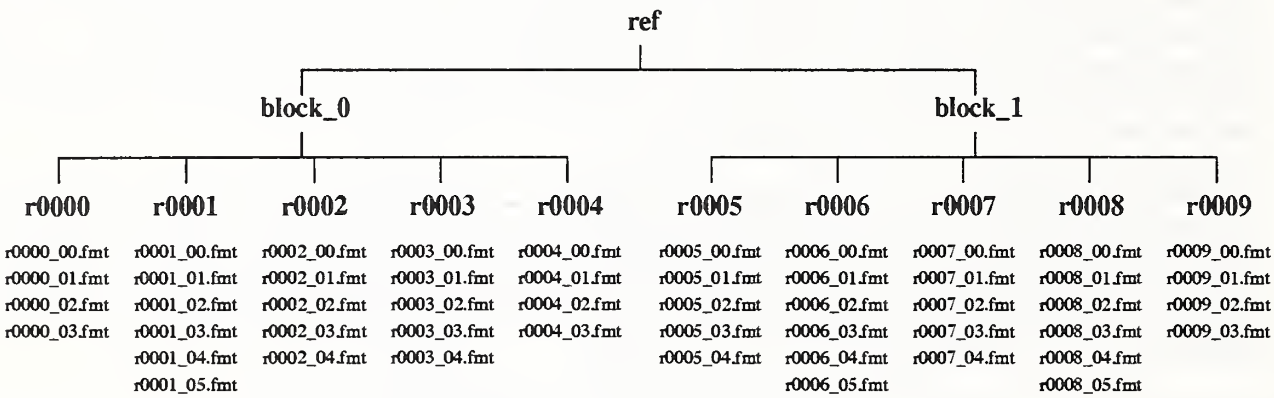

Figure 7: Directory hierarchy for the Certification Package directory ref.

2.2.1 File Format Terminology

To simplify file format descriptions, several terms must be defined. A Single-Value ASCII String Representation (SVASR) is a buffer of variable length containing any number of printable ASCII characters in the hexadecimal range 21 to 7E. A SVASR is void of any space characters, hexadecimal 20. A Multiple-Value ASCII String Representation (MVASR) is a buffer of variable length containing any number of printable ASCII characters in the hexadecimal range 20 to 7E including any number of space characters. An ASCII Delimiter Character (ADC) is a single space character, hexadecimal 20. The ADC is used to separate a line of contiguous SVASR's or to separate a SVASR followed by a MVASR. An ASCII Line Representation (ALR) is a buffer of variable length containing any number and combination of SVASRs, MVASRs, and ADCs terminated by the ASCII LF character, hexadecimal 0A. This means that the ASCII CR character, hexadecimal 0D, cannot occur anywhere in an ALR, or in place of, or in combination with the ASCII LF character 0A at the end of the ALR. Also note that all files described in this document do not contain any end-of-file marker or end-of-file character.

2.2.2 Reference File Format

For every form image in the Certification Package an associated reference file is provided in the directory ref. These reference files contain the identification of the form face contained in the form image followed by the actual data entered in each field on the form. The Scoring Package treats the form identification and entry field values recorded in the reference file as ground truth. The integrity of any test is completely dependent on the accuracy of these files. Appendix A contains an image of a completed first page of a 1988 1040 tax form and the reference file associated with the form image is listed in two adjacent text columns in Figure 22. Note that the information contained in the form was derived from a computer and does not contain real tax information.

A reference file is comprised of a variable number of ALRs with the first ALR identifying the image's form face followed by one ALR per entry field on the image. The form identification is the first ALR in the reference file and is represented as a SVASR. Each ALR following the form identification ALR corresponds to a specific entry field on the form. These entry field ALRs contain a required entry field identification string and a conditional entry field value. The identification string is represented as a SVASR and the entry field value is represented as an MVASR. If an entry field ALR contains the conditional entry field value, then the ALR is comprised of a SVASR and MVASR separated by an ADC. If an entry field ALR does not contain an entry field value, then the ALR is comprised of a SVASR representing the identification string only. If an entry field contains data, then its value contains exactly what was entered in the field. If an entry field is blank, then its value is omitted from the ALR including the omission of the ADC.

Figure 8 lists the first ten lines of a reference file for the first page of a 1988 1040 form. The first line identifies the form face contained in the form image. The remaining lines correspond to the first 9 entry fields contained on the first page of the 1040 form.

Notice that the first three entry fields (1040_1_L_H1_V1, 1040_1_L_H2_V1, 1040_1_L_H3_V1) have no entry field value entered in the reference file because their corresponding fields on the form were left empty. Figure 9 lists byte for byte the hexadecimal representation of the ten lines listed in Figure 8. Notice the hexadecimal 0A character terminating each line. Also notice that the first three entry field ALRs contain only a single SVASR representing identification strings without associated values. This represents three entry fields left blank on the form image. The remaining six entry field ALRs contain both identification strings and values. These are entry fields that were filled in on the form image. Notice that the identification strings are represented as SVASRs, the values are represented as MVASRs, and that there is a single ADC, hexadecimal 20, separating the two.

```
1040_1
1040_1_L_H1_V1
1040_1_L_H2_V1
1040_1_L_H3_V1
1040_1_L_H1_V2 Berry K. & Loras A. Boyle
1040_1_L_H2_V2 A15 86 7384
1040_1_L_H1_V3 7861 Fairfield Street
1040_1_L_H2_V3 A73 28 5386
1040_1_L_H1_V4 Boyle, MT 30073
1040_1_L_H1_V5 1
```

Figure 8: Top of an example reference file.

```
31 30 34 30 5F 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 32 5F 56 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 33 5F 56 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 32 20 42 65 72 72 79 20 4B 2E 20 26 20 4C 6F 72 61 73 20 41 2E 20 42 6F 79 6C 65 0A
31 30 34 30 5F 31 5F 4C 5F 48 32 5F 56 32 20 41 31 35 20 38 36 20 37 33 38 34 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 33 20 37 38 36 31 20 46 61 69 72 66 69 65 6C 64 20 53 74 72 65 65 74 0A
31 30 34 30 5F 31 5F 4C 5F 48 32 5F 56 33 20 41 37 33 20 32 38 20 35 33 38 36 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 34 20 42 6F 79 6C 65 2C 20 4D 54 20 33 30 30 37 33 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 35 20 31 0A.
```

Figure 9: Hexadecimal listing of the reference file portion listed in Figure 8.

Entry field 1040_1_L_H1_V5 is an example of an Icon entry field.[4][5] Notice that this field's value is a '1' which signifies that the field contains a box check mark. If the Icon entry field was empty on the form, then a value of '0' would be used in the reference file. This convention reflects a format change in the way Icon entry fields are represented in the reference file. In the past, such as in SD2 and SD6, the entry field value was left blank when no information was present, and in SD2 a value of "_ICON_" was used in place of the '1' when information was present.

The entry field identification strings listed in the reference file must match exactly in name and in order to the identification strings recorded in the Table A file associated with the image's form face. Table A files are distributed with SD2 and SD6 and are included in the Certification Package directory **util/tables**. The SVASR used for the form identification is embedded in the associated Table A file name. Notice that the form identification in the reference file example is "1040_1". The Table A file corresponding to this form face is **util/tables/1040_1.tab**. For historical reasons, the reference files used for form-based scoring have also been called format files. All reference files have a consistent extension of ".fmt".

2.3 Certification Utilities

The Certification Package directory **util** contains scoring utilities and Table A files for the form faces included in image. The contents of this directory is shown in Figure 10. The Scoring Package is implemented as two separate programs. The program **merge** combines form reference files with recognition system output files, and the program **score** computes statistics and performance measures on the data contained in the **merge** output files. The subdirectory **tables** contains Table A files to be used by the Scoring

Package program **merge** when processing recognition system output files. The subdirectory **bin** contains UNIX Bourne Shell (sh) scripts that can be used to **merge** and **score** the recognition system output files discussed in Section 3.1. Scoring instructions and the use of these utilities are discussed in Section 3.2. Together, the directories **image**, **ref**, and **util** comprise a Certification Package.

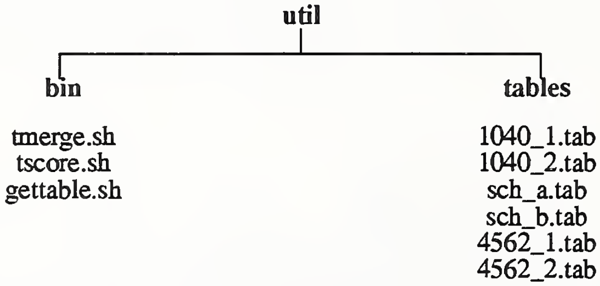

Figure 10: Contents of the Certification Package directory **util**.

3. Certification Return

Upon receipt of a Certification Package, the organization must process *all* of the form images in the package, generate recognition system output files, score the output files, and return to NIST both the recognition system output files and the Scoring Package output files. Together, the recognition system output files and the Scoring Package output files comprise a Certification Return. The organization requesting certification must submit a Certification Return to NIST on a data-grade 8mm cartridge tape in tar format at a density of 2.2 Gigabytes per tape according to the directory structures and file formats presented in this section. For example, the following commands load a Certification Package into the directory **/usr/local/cert/pckg** assuming the device **/dev/rst1** corresponds to a 8mm tape drive on the organization's computer system running UNIX.

```
# mkdir /usr/local/cert
# mkdir /usr/local/cert/pckg
# cd /usr/local/cert/pckg
# tar xvf /dev/rst1
```

Figure 11 illustrates the top level directory of a Certification Return. The directory **system** is to contain the organization's system output files and the Scoring Package's **merge** output files, and the directory **score** is to contain the Scoring Package's **score** output files. The following commands write a Certification Return to tape that was created by the organization in the directory **/usr/local/cert/rtn**.

```
# cd /usr/local/cert/rtn
# tar cvf /dev/rst1 ./system ./score
```

3.1 Recognition System Output Files

Recognition system output files are reported in the directory **system** for each form image in the Certification Package. These output files include hypothesis files, rejection files, and confidence files. The reporting of hypothesis files and rejection files is mandatory and the reporting of confidence files is optional. Hypothesis files are expected to contain occurrences of substituted, inserted, and deleted characters, rejection files are expected to contain both accepted and rejected characters, and confidence files (if provided) are expected to contain floating point numbers varying between 0.0 and 1.0. An example of a Certification Return directory **system** prior to scoring is shown in Figure 12. In this example, the organization reported all three types of files. The directory hierarchy within the directory **system** is identical to the directory hierarchy in the Certification Package directory **image**. The file name conventions for hypothesis, rejection, and confidence files use the same root name from the corresponding form image distributed in the Certification Package. For example, recognition system output files for the form image **r0007_00.pct** are the hypothesis file **r0007_00.HYP**, the rejection file **r0007_00.REJ**, and the confidence file **r0007_00.CON**.

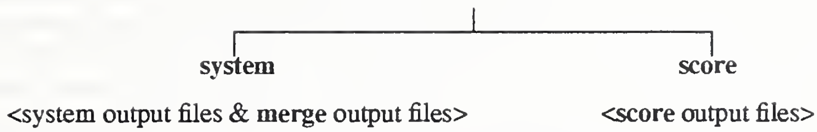

Figure 11: Top level directory of a Certification Return.

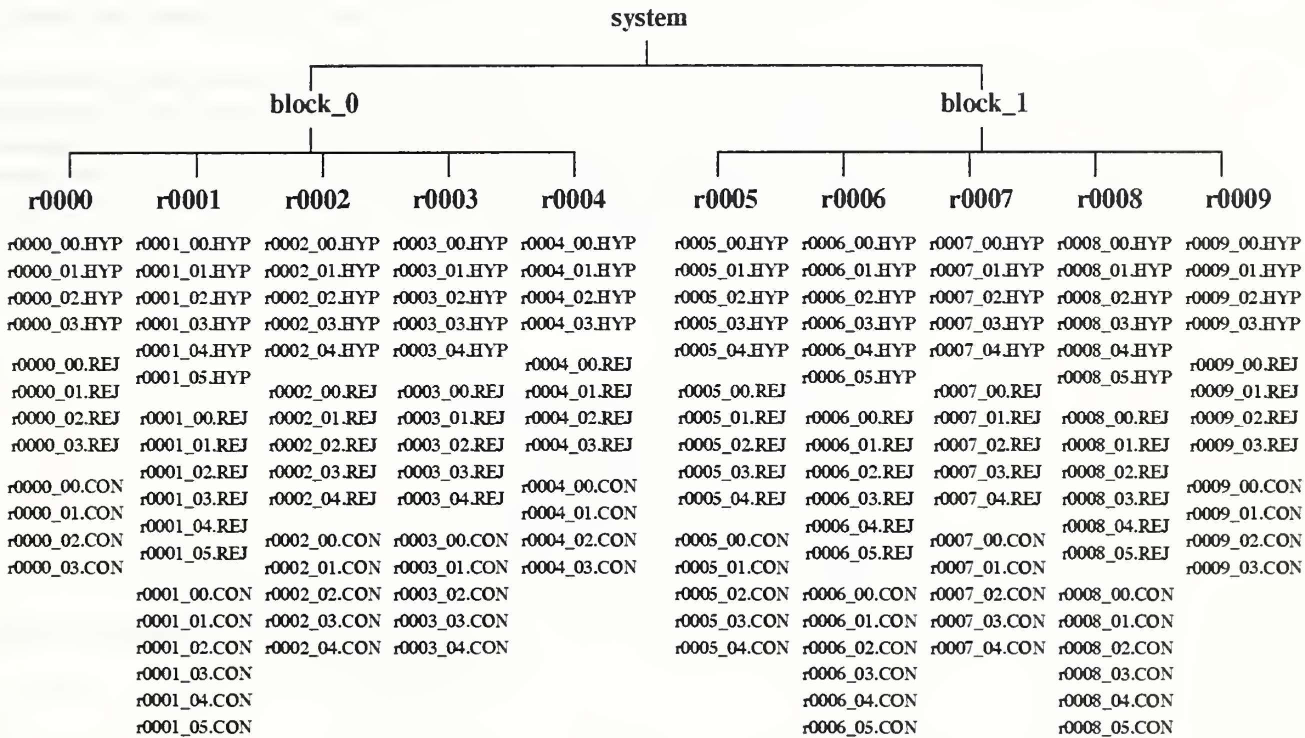

Figure 12: Directory hierarchy for the Certification Return directory **system** prior to scoring.

3.1.1 Hypothesis File Format

For every form image in the Certification Package, the organization requesting certification must return an associated hypothesis file. Each hypothesis file contains the form face identified by the recognition system followed by the results of what the system captured and recognized from each entry field on the form image. The Scoring Package aligns the results with the true entry field values contained in the form image's associated reference file in order to compute error rates. Appendix A contains an example of a hypothesis file corresponding to the completed form displayed in the appendix. The hypothesis file shown in Figure 23 is listed in two adjacent text columns.

Hypothesis files are identical in format to reference files. A hypothesis file is comprised of a variable number of ALRs with the first ALR containing the form face identified by the system followed by one ALR per entry field on the form. The form identification is the first ALR in the hypothesis file and is represented as a SVASR. Each ALR following the form identification ALR corresponds to a specific entry field on the form. These entry field ALRs contain a required entry field identification string and a conditional entry field value. The identification string is represented as a SVASR and the value is represented as an MVASR. If the system detected and captured data within an entry field, then the recognized value is included and the entry field ALR is comprised of a SVASR and MVASR separated by one ADC. If the system detected a blank field, then the value is omitted including the ADC, and the entry field ALR is comprised only of a SVASR representing the identification string.

It is common for alphanumeric entry fields to be made up of more than one word. Therefore, the recognition of spacing must be addressed. When capturing and recognizing fixed-spaced machine generated text, spaces between words are clearly detectable. When capturing and recognizing proportionally-spaced machine generated text, the detection of spaces becomes slightly obscure. When capturing and recognizing hand-printed data, the detection of spaces without the use of dictionaries and grammars becomes practically impossible. In light of this, the organization has the choice of reporting recognition results with or without the recognition of spaces. If the organization chooses to report the recognition of spaces, then the value of an entry field detected to contain multiple words will contain a space character wherever the system detected one. Remember that the value of an entry field ALR in the hypothesis file is a MVASR which includes the existence of space characters, hexadecimal 20. If the organization chooses not to report the recognition of spaces, then the value of all entry field ALRs, even if the entry field really is comprised of multiple words, will contain no space characters. The Scoring Package can handle either hypothesis format for entry field values.

Figure 13 lists the first ten lines of an example hypothesis file where the organization chose not to report the recognition of spaces. This example represents perfect recognition of the form corresponding to the reference file in Figure 8. Figure 14 lists byte for byte the hexadecimal representation of the ten lines listed in Figure 13. Notice that the multiple word values do not have any space characters, hexadecimal 20. Also note that the fields having recognized information retain the use of the ADC to separate the entry field identification string from the entry field value in the hypothesis file.

```
1040_1
1040_1_L_H1_V1
1040_1_L_H2_V1
1040_1_L_H3_V1
1040_1_L_H1_V2 BerryK.&LorasA.Boyle
1040_1_L_H2_V2 A15867384
1040_1_L_H1_V3 7861FairfieldStreet
1040_1_L_H2_V3 A73285386
1040_1_L_H1_V4 Boyle,MT30073
1040_1_L_H1_V5 1
```

Figure 13: Top of an example hypothesis file where the organization chose not to report the recognition of spaces.

```
31 30 34 30 5F 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 32 5F 56 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 33 5F 56 31 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 32 20 42 65 72 72 79 4B 2E 26 4C 6F 72 61 73 41 2E 42 6F 79 6C 65 0A
31 30 34 30 5F 31 5F 4C 5F 48 32 5F 56 32 20 41 31 35 38 36 37 33 38 34 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 33 20 37 38 36 31 46 61 69 72 66 69 65 6C 64 53 74 72 65 65 74 0A
31 30 34 30 5F 31 5F 4C 5F 48 32 5F 56 33 20 41 37 33 32 38 35 33 38 36 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 34 20 42 6F 79 6C 65 2C 4D 54 33 30 30 37 33 0A
31 30 34 30 5F 31 5F 4C 5F 48 31 5F 56 35 20 31 0A
```

Figure 14: Hexadecimal listing of the hypothesis file portion listed in Figure 13.

The entry field identification strings listed in the hypothesis file must match exactly in name and in order to the identification strings recorded in the Table A file associated with the image's form face. Table A files are distributed with SD2 and SD6 and are included in the Certification Package directory **util/tables**. All hypothesis file names should end with the extension ".HYP".

3.1.2 Rejection File Format

Rejection files are a second type of file required to be returned by the organization requesting certification. A recognition system may use very sophisticated methods for determining whether a recognition decision should be accepted or rejected. Therefore, rather than return raw confidence values in an optional confidence file, the organization must specify explicitly which classifications should be rejected and which should be accepted in a rejection file. Appendix A contains an example of a rejection file. Note that the line breaks within single entry field specifications in Figure 24 are due to the wrap-around properties of the listing and do not indicate the presence of new-line characters in the file.

Rejection files are comprised of a variable number of ALRs with the first ALR containing information as to whether the recognition system accepted or rejected the identification of the form face followed by one ALR per entry field on the form image. The rejection line corresponding to the form identification is the first ALR in the rejection file and consists of the form identification (also included in the hypothesis file) and a binary reject value. Both the form identification and the rejection value are represented as SVASRs separated by an ADC. Each ALR following the form identification ALR corresponds to a specific entry field on the form. These entry field ALRs contain a required entry field identification string and a conditional list of reject values. The identification string and reject values are represented as SVASRs separated by ADCs. If the system detected and captured data within an entry field, then the reject values are included and the entry field ALR contains the identification string and one reject value for each individual character captured and classified. If the system detected a blank field, then the reject values are omitted, and the entry field ALR contains a SVASR representing the entry field identification string only.

Reject values must be a binary number equal to the characters ‘0’ or ‘1’. A ‘1’ indicates that the classification should be scored as unknown rather than as a correct or incorrect classification. A ‘0’ indicates that the classification should be scored correct if the hypothesized character is identical to the reference character and scored incorrect otherwise. Regardless if an organization chooses to report the recognition results of spaces or not, the number of bytes comprising an entry field’s value MVASR in the hypothesis file must equal the number of individual reject values reported in the rejection file for the entry field. Failure of the number of bytes in the hypothesis file’s MVSAR to equal the number of reject values in the rejection file will result in the entry field being removed from the analysis conducted by the Scoring Package and a warning message will be displayed.

```
1040_1
1040_1_L_H1_V1
1040_1_L_H2_V1
1040_1_L_H3_V1 87
1040_1_L_H1_V2 B. Boyle
```

Figure 15: Top of a hypothesis file where the organization chose to report the recognition of spaces.

```
31 30 34 30 5f 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 31 5f 56 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 32 5f 56 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 33 5f 56 31 20 38 37 0a
31 30 34 30 5f 31 5f 4c 5f 48 31 5f 56 32 20 42 2e 20 42 6f 79 6c 65 0a
```

Figure 16: Hexadecimal listing of the hypothesis file portion listed in Figure 15.

```
1040_1 0
1040_1_L_H1_V1
1040_1_L_H2_V1
1040_1_L_H3_V1 0 0
1040_1_L_H1_V2 1 0 0 0 0 0 1 0
```

Figure 17: Top of a rejection file corresponding to the hypothesis file shown in Figure 15.

```
31 30 34 30 5f 31 20 30 0a
31 30 34 30 5f 31 5f 4c 5f 48 31 5f 56 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 32 5f 56 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 33 5f 56 31 20 30 20 30 0a
31 30 34 30 5f 31 5f 4c 5f 48 31 5f 56 32 20 31 20 30 20 30 20 30 20 30 20 30 20 31 20 30 0a
```

Figure 18: Hexadecimal listing of the rejection file portion listed in Figure 17.

Figure 17 lists the first five lines of a rejection file corresponding to the example hypothesis file listed in Figure 15. Notice that in the last line of Figure 17 there is a reject value ('0' or '1') for each and every byte of the MVASR listed in the last line of Figure 15 including a reject value for the space character. If the organization had chosen not to report the recognition results of space characters, then the reject value for the space character would be omitted.

The entry field identification strings listed in the rejection file must match exactly in name and in order to the identification strings recorded in the Table A file associated with the image's form face. Table A files are distributed with SD2 and SD6 and are included in the Certification Package directory **util/tables**. All rejection files should end with the extension ".REJ".

3.1.3 Confidence File Format

Character classifiers typically produce a floating point value on the range 0.0 to 1.0, representing how confident the classifier is of its recognition decision. By setting thresholds on these values, an organization requesting certification can tune its system to desired levels of performance trading off throughput for accuracy.[8] The Scoring Package is capable of conducting basic analyses with only the recognition system's hypothesis file aligned with the form image's reference file. However, through the optional use of confidence files, the Scoring Package can do additional analyses if the organization requesting certification provides confidence values for each character classified. Appendix A contains an example of a confidence file corresponding to the completed form displayed in the appendix. Note that the line breaks within single entry field specifications in Figure 25 are due to the wrap-around properties of the listing. Line breaks within an entry field specification do not indicate the presence of new-line characters in the file.

Confidence files are comprised of a variable number of ALRs with the first ALR containing the confidence of the recognition system's identification of the form face followed by one ALR per entry field on the form image. The confidence of the form identification is the first ALR in the confidence file and consists of the form identification (also included in the hypothesis file) and the actual confidence value. Both the form identification and the confidence value are represented as SVASRs separated by an ADC. Each ALR following the form identification ALR corresponds to a specific entry field on the form. These entry field ALRs contain a required entry field identification string and a conditional list of confidence values. The identification string and confidence values are represented as SVASRs separated by ADCs. If the system detected and captured data within an entry field, then the confidence values are included and the entry field ALR contains the identification string and one confidence value for each individual character captured and classified. If the system detected a blank field, then the confidence values are omitted, and the entry field ALR contains a SVASR representing the entry field identification string.

A confidence value must be a number ranging from 0.0 through 1.0. The number of digits to the right of the decimal point must be less than 17. Whether or not an organization chooses to report its recognition results of spaces, the number of bytes comprising an entry field's value MVASR in the hypothesis file must equal the number of individual confidence values reported in the confidence file for the entry field. Failure of the number of bytes in the hypothesis file's MVSAR to equal the number of confidence values in the confidence file will result in the entry field being removed from the analysis conducted by the Scoring Package and a warning message will be displayed.

Figure 15 lists the first five lines of an example hypothesis file where the organization chose to report its recognition results of spaces. Figure 19 lists the corresponding lines from an example confidence file. Notice that the last line in Figure 15 contains a space character in the MVASR "B. Boyle" which is listed as hexadecimal in Figure 16 as "42 2e 20 42 6f 79 6c 65". Also notice that the eight bytes in the MVASR are assigned exactly eight confidence values in the last line of Figure 19. Had the recognition results of spaces not been reported, then the MVASR of the last line in Figure 15 would be "B.Boyle" without the space character. The hexadecimal listing for the MVASR would be "42 2e 42 6f 79 6c 65", omitting the hexadecimal 20. In turn, the list of confidence values in Figure 19 would be reduced from eight values to seven with the confidence value "0.258367" omitted.

```
1040_1 0.989425
1040_1_L_H1_V1
1040_1_L_H2_V1
1040_1_L_H3_V1 0.786324 0.998934
1040_1_L_H1_V2 0.675347 0.994671 0.258367 0.683123 0.876284 0.391576 0.4987481 0.719952
```

Figure 19: Top of an example confidence file corresponding to the hypothesis file in Figure 15.

```
31 30 34 30 5f 31 20 30 2e 39 38 39 34 32 35 0a
31 30 34 30 5f 31 5f 4c 5f 48 31 5f 56 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 32 5f 56 31 0a
31 30 34 30 5f 31 5f 4c 5f 48 33 5f 56 31 20 30 2e 37 38 36 33 32 34 20 30 2e 39 39 38 39 33 34 0a
31 30 34 30 5f 31 5f 4c 5f 48 31 5f 56 32 20 30 2e 36 37 35 33 34 37 20 30 2e 39 39 34 36 37 31 \
        20 30 2e 32 35 38 33 36 37 20 30 2e 36 38 33 31 32 33 20 30 2e 38 37 36 32 38 34 \
        20 30 2e 33 39 31 35 37 36 20 30 2e 34 39 38 37 34 38 31 20 30 2e 37 31 39 39 35 32 0a
```

Figure 20: Hexadecimal listing of the confidence file portion listed in Figure 19.

The entry field identification strings listed in the confidence file must match exactly in name and in order to the identification strings recorded in the Table A file associated with the image's form face. Table A files are distributed with SD2 and SD6 and are included in the Certification Package directory **util/tables**. All confidence files should end with the extension ".CON".

3.2 Scoring Package Output Files

system

block_0					**block_1**				
r0000	**r0001**	**r0002**	**r0003**	**r0004**	**r0005**	**r0006**	**r0007**	**r0008**	**r0009**
r0000_00.HYP	r0001_00.HYP	r0002_00.HYP	r0003_00.HYP	r0004_00.HYP	r0005_00.HYP	r0006_00.HYP	r0007_00.HYP	r0008_00.HYP	r0009_00.HYP
r0000_01.HYP	r0001_01.HYP	r0002_01.HYP	r0003_01.HYP	r0004_01.HYP	r0005_01.HYP	r0006_01.HYP	r0007_01.HYP	r0008_01.HYP	r0009_01.HYP
r0000_02.HYP	r0001_02.HYP	r0002_02.HYP	r0003_02.HYP	r0004_02.HYP	r0005_02.HYP	r0006_02.HYP	r0007_02.HYP	r0008_02.HYP	r0009_02.HYP
r0000_03.HYP	r0001_03.HYP	r0002_03.HYP	r0003_03.HYP	r0004_03.HYP	r0005_03.HYP	r0006_03.HYP	r0007_03.HYP	r0008_03.HYP	r0009_03.HYP
r0000_00.REJ	r0001_04.HYP	r0002_04.HYP	r0003_04.HYP	r0004_00.REJ	r0005_04.HYP	r0006_04.HYP	r0007_04.HYP	r0008_04.HYP	r0009_00.REJ
r0000_01.REJ	r0001_05.HYP	r0002_00.REJ	r0003_00.REJ	r0004_01.REJ	r0005_00.REJ	r0006_05.HYP	r0007_00.REJ	r0008_05.HYP	r0009_01.REJ
r0000_02.REJ	r0001_00.REJ	r0002_01.REJ	r0003_01.REJ	r0004_02.REJ	r0005_01.REJ	r0006_00.REJ	r0007_01.REJ	r0008_00.REJ	r0009_02.REJ
r0000_03.REJ	r0001_01.REJ	r0002_02.REJ	r0003_02.REJ	r0004_03.REJ	r0005_02.REJ	r0006_01.REJ	r0007_02.REJ	r0008_01.REJ	r0009_03.REJ
r0000_00.CON	r0001_02.REJ	r0002_03.REJ	r0003_03.REJ	r0004_00.CON	r0005_03.REJ	r0006_02.REJ	r0007_03.REJ	r0008_02.REJ	r0009_00.CON
r0000_01.CON	r0001_03.REJ	r0002_04.REJ	r0003_04.REJ	r0004_01.CON	r0005_04.REJ	r0006_03.REJ	r0007_04.REJ	r0008_03.REJ	r0009_01.CON
r0000_02.CON	r0001_04.REJ	r0002_00.CON	r0003_00.CON	r0004_02.CON	r0005_00.CON	r0006_04.REJ	r0007_00.CON	r0008_04.REJ	r0009_02.CON
r0000_03.CON	r0001_05.REJ	r0002_01.CON	r0003_01.CON	r0004_03.CON	r0005_01.CON	r0006_05.REJ	r0007_01.CON	r0008_05.REJ	r0009_03.CON
r0000_00.mrg	r0001_00.CON	r0002_02.CON	r0003_02.CON	r0004_00.mrg	r0005_02.CON	r0006_00.CON	r0007_02.CON	r0008_00.CON	r0009_00.mrg
r0000_01.mrg	r0001_01.CON	r0002_03.CON	r0003_03.CON	r0004_01.mrg	r0005_03.CON	r0006_01.CON	r0007_03.CON	r0008_01.CON	r0009_01.mrg
r0000_02.mrg	r0001_02.CON	r0002_04.CON	r0003_04.CON	r0004_02.mrg	r0005_04.CON	r0006_02.CON	r0007_04.CON	r0008_02.CON	r0009_02.mrg
r0000_03.mrg	r0001_03.CON	r0002_00.mrg	r0003_00.mrg	r0004_03.mrg	r0005_00.mrg	r0006_03.CON	r0007_00.mrg	r0008_03.CON	r0009_03.mrg
	r0001_04.CON	r0002_01.mrg	r0003_01.mrg		r0005_01.mrg	r0006_04.CON	r0007_01.mrg	r0008_04.CON	
	r0001_05.CON	r0002_02.mrg	r0003_02.mrg		r0005_02.mrg	r0006_05.CON	r0007_02.mrg	r0008_05.CON	
	r0001_00.mrg	r0002_03.mrg	r0003_03.mrg		r0005_03.mrg	r0006_00.mrg	r0007_03.mrg	r0008_00.mrg	
	r0001_01.mrg	r0002_04.mrg	r0003_04.mrg		r0005_04.mrg	r0006_01.mrg	r0007_04.mrg	r0008_01.mrg	
	r0001_02.mrg					r0006_02.mrg		r0008_02.mrg	
	r0001_03.mrg					r0006_03.mrg		r0008_03.mrg	
	r0001_04.mrg					r0006_04.mrg		r0008_04.mrg	
	r0001_05.mrg					r0006_05.mrg		r0008_05.mrg	

Figure 21: Directory hierarchy for the Certification Return directory system after running the program merge.

Recognition system output files must be generated for each form image in the Certification Package and stored in the Certification Return directory **system** as shown in Figure 12. The organization requesting certification must then score its system output files using the Scoring Package. The files generated by the program **merge** are stored with the recognition system output files in the directory **system** as shown in Figure 21. One **merge** output file ending with the extension ".mrg" must be created for each hypothesis file reported. A UNIX Bourne Shell (sh) script **tmerge.sh** is provided in the Certification Package directory **util/bin**.

tmerge.sh [-c] *refdir hypdir*

This utility combines the form reference files in *refdir* with the recognition system output files in *hypdir* by invoking the program **merge**, creating one ".mrg" output file in *hypdir* for each form reported. The "-c" flag is used if confidence files are included with the recognition system's output. Otherwise, the flag is omitted. For example, if the Certification Package was loaded into the directory **/usr/local/cert/pckg** and the Certification Return was being built in the directory **/usr/local/cert/rtn** with confidence files reported, the tmerge.sh should be invoked as follows:

```
# cd /usr/local/cert/pckg/util/bin
# tmerge.sh -c /usr/local/cert/pckg/ref /usr/local/cert/rtn/system
```

If the shell script is not supported by the organization's computing environment, **merge** can be run independently with the following options:

```
-o quit,formtypes,conf={n|c},nrej=1,table_a_dir=../tables
```

For complete details on executing **merge**, refer to the *NIST Scoring Package User's Guide*.[2]

If confidence files are provided, then "conf=c" is used. Otherwise, "conf=n" is used. Also note that the script **tmerge.sh** must be invoked from within the Certification Package directory **util/bin** in order for the script to locate the Table A files included in the Certification Package directory **util/tables**.

The **merge** output files must then be processed by the program **score**. Another UNIX Bourne Shell (sh) script **tscore.sh** is provided in the Certification Package directory **util/bin**.

tscore.sh *mrgdir outdir*

The script scores all the **merge** output files found in the directory *mrgdir* and generates two scoring output files, a summary report **system.sum** and a fact sheet **system.fct**, storing them in the directory *outdir*. In the appendix, Figure 26 shows an example of a Scoring Package summary report, and Figure 27 shows an example of a corresponding fact sheet. Assuming the same directories used in the previous **tmerge.sh** example, **tscore.sh** should be invoked as follows:

```
# cd /usr/local/cert/pckg/util/bin
# tscore.sh /usr/local/cert/rtn/system /usr/local/cert/rtn/score
```

If the shell script is not supported by the organization's computing environment, **score** can be run independently with the following options:

```
-o quit,nowhite -s output=FCItd,of=system.sum,cf=system.fct
```

For complete details on executing **score**, refer to the *NIST Scoring Package User's Guide*.[2]

4. Score Verification and Certification

Certification is available as a service to purchasers of the NIST Scoring Package. Upon request, the organization will be sent a Certification Package. Requests for certification should be directed to the author, and to receive further information related to the Scoring Package, contact the Standard Reference Data Division at NIST. Upon receiving a Certification Return from an organization, NIST runs the Scoring Package on the recognition system output files in the Certification Return directory system and generates another set of scoring output files. The organization must submit its Certification Return in full compliance with the procedures and formats listed herein, and all recognition system output files must comply precisely with the file formats required by the Scoring Package. In the event that compliance is not achieved, the organization is contacted with an explanation of existing problems and is required to resubmit its Certification Return correcting all problems.

After successful scoring, the scores generated by NIST are compared against the scores submitted by the organization requesting certification. If discrepancies exist, the organization is notified, and the organization must resolve the problem and resubmit a new Certification Return. If no discrepancies between the sets of scores exist, the NIST scores are presented to the organization along with a Score Verification Form. The organization requesting certification must sign and return the form verifying that the scores generated by NIST are accurate. NIST in turn signs the Score Verification Form signifying successful completion of the certification process, retains one copy for NIST files, and returns the original to the organization as proof of certification.

5. References

[1] M. D. Garris and S. A. Janet. Scoring Package Release 1.0, Technical Report Special Software 1, SP, National Institute of Standards and Technology, October 1992.

[2] M. D. Garris and S. A. Janet. NIST Scoring Package User's Guide, Release 1.0. Technical Report NISTIR 4950, National Institute of Standards and Technology, October 1992.

[3] R. A. Wilkinson, et al. The first Census optical character recognition system conference. Technical Report NISTIR 4912, National Institute of Standards and Technology, July 1992.

[4] D. L. Dimmick, M. D. Garris, and C. L. Wilson. Structured Forms Database, Technical Report Special Database 2, SFRS, National Institute of Standards and Technology, December 1991.

[5] D. L. Dimmick and M. D. Garris. Structured Forms Database 2, Technical Report Special Database 6, SFRS2, National Institute of Standards and Technology, September 1992.

[6] Department of Defense, "Military Specification - Raster Graphics Representation in Binary Format, Requirements for, MIL-R-28002," 20 Dec 1988.

[7] CCITT, "Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, Fascicle VII.3 - Rec. T.6," 1984.

[8] M. D. Garris, et al. Massively parallel implementation of character recognition systems. In *Conference on Character Recognition and Digitizer Technologies*, volume 1661, pages 269-280, San Jose California, February 1992. SPIE.

Appendix A: Form-Based Scoring Files

Form **1040** Department of the Treasury—Internal Revenue Service
U.S. Individual Income Tax Return **1988**

For the year Jan.-Dec. 31, 1988, or other tax year beginning July, 1988, ending July, 1988 OMB No. 1545-0074

Label

Use IRS label. Otherwise, please print or type.

LABEL HERE

Your first name and initial (if joint return, also give spouse's name and initial) Last name: Brainerd A. & Erskine W. Mitchell

Your social security number: A11 88 1304

Present home address (number, street, and apt. no. or rural route). (If a P.O. Box, see page 6 of Instructions.): 99225 Lee Streer

Spouse's social security number: A59 02 1948

City, town or post office, state, and ZIP code: Russell, NJ 61920

For Privacy Act and Paperwork Reduction Act Notice, see Instructions.

Presidential Election Campaign

Do you want $1 to go to this fund? X Yes ☐ No

If joint return, does your spouse want $1 to go to this fund? X Yes ☐ No

Note: Checking "Yes" will not change your tax or reduce your refund.

Filing Status

Check only one box.

1. X Single
2. ☐ Married filing joint return (even if only one had income)
3. ☐ Married filing separate return. Enter spouse's social security no. above and full name here.
4. ☐ Head of household (with qualifying person). (See page 7 of Instructions.) If the qualifying person is your child but not your dependent, enter child's name here.
5. ☐ Qualifying widow(er) with dependent child (year spouse died ▶ 19). (See page 7 of Instructions.)

Exemptions

(See Instructions on page 8.)

6a X Yourself If someone (such as your parent) can claim you as a dependent, do not check box 6a. But be sure to check the box on line 33b on page 2.

No. of boxes checked on 6a and 6b: 1

b ☐ Spouse

c Dependents:

(1) Name (first, initial, and last name)	(2) Check if under age 5	(3) If age 5 or older, dependent's social security number	(4) Relationship	(5) No. of months lived in your home in 1988
Rider Harlan		A97 20 3766	Aunt	6
Tulane Banks		A97 08 1904	S-Law	12
Hunter Bell		A39 26 7506	Da-Law	2

If more than 6 dependents, see Instructions on page 8.

No. of your children on 6c who:
- lived with you: 8
- didn't live with you due to divorce or separation:

No. of other dependents listed on 6c:

Add numbers entered on lines above ▶ 9

d If your child didn't live with you but is claimed as your dependent under a pre-1985 agreement, check here ▶ ☐

e Total number of exemptions claimed

Income

Please attach Copy B of your Forms W-2, W-2G, and W-2P here.

If you do not have a W-2, see page 6 of Instructions.

Please attach check or money order here.

7	Wages, salaries, tips, etc. *(attach Form(s) W-2)*	3878
8a	Taxable interest income *(also attach Schedule B if over $400)*	
b	Tax-exempt interest income (see page 11). DON'T include on line 8a — 8b	
9	Dividend income *(also attach Schedule B if over $400)*	
10	Taxable refunds of state and local income taxes, if any, from worksheet on page 11 of Instructions	
11	Alimony received	
12	Business income or (loss) *(attach Schedule C)*	0
13	Capital gain or (loss) *(attach Schedule D)*	
14	Capital gain distributions not reported on line 13 (see page 11)	
15	Other gains or (losses) *(attach Form 4797)*	
16a	Total IRA distributions — 16a; 16b Taxable amount (see page 11)	
17a	Total pensions and annuities — 17a; 17b Taxable amount (see page 12)	
18	Rents, royalties, partnerships, estates, trusts, etc. *(attach Schedule E)*	
19	Farm income or (loss) *(attach Schedule F)*	
20	Unemployment compensation (insurance) (see page 13)	
21a	Social security benefits (see page 13) — 21a	
b	Taxable amount, if any, from the worksheet on page 13	
22	Other income (list type and amount—see page 13) Travel allowance	0
23	Add the amounts shown in the far right column for lines 7 through 22. This is your **total income** ▶	3878

Adjustments to Income

(See Instructions on page 13.)

24	Reimbursed employee business expenses from Form 2106, line 13		
25a	Your IRA deduction, from applicable worksheet on page 14 or 15	2574	
b	Spouse's IRA deduction, from applicable worksheet on page 14 or 15		
26	Self-employed health insurance deduction, from worksheet on page 15		
27	Keogh retirement plan and self-employed SEP deduction		
28	Penalty on early withdrawal of savings		
29	Alimony paid (recipient's last name ____ and social security no. ____)		
30	Add lines 24 through 29. These are your **total adjustments** ▶		2574

Adjusted Gross Income

31	Subtract line 30 from line 23. This is your **adjusted gross income**. *If this line is less than $18,576 and a child lived with you, see "Earned Income Credit" (line 56) on page 19 of the Instructions. If you want IRS to figure your tax, see page 16 of the Instructions.* ▶	1303

```
1040_1
1040_1_L_H1_V1 July
1040_1_L_H2_V1 July
1040_1_L_H3_V1 88
1040_1_L_H1_V2 Brainerd A. & Erskine W. Mitchell
1040_1_L_H2_V2 A11 88 1304
1040_1_L_H1_V3 99225 Lee Street
1040_1_L_H2_V3 A59 02 1948
1040_1_L_H1_V4 Russell, NJ 61920
1040_1_L_H1_V5 1
1040_1_L_H2_V5 0
1040_1_L_H1_V6 1
1040_1_L_H2_V6 0
1040_1_1 1
1040_1_2 0
1040_1_3_H1 0
1040_1_3_H2
1040_1_4_H1 0
1040_1_4_H2
1040_1_5_H1 0
1040_1_5_H2
1040_1_6a 1
1040_1_6b_H1 0
1040_1_6b_H2 1
1040_1_6c_H1_V1 Rider Harlan
1040_1_6c_H2_V1 0
1040_1_6c_H3_V1 A97 20 3760
1040_1_6c_H4_V1 Aunt
1040_1_6c_H5_V1 6
1040_1_6c_H6_V1 8
1040_1_6c_H1_V2 Tulane Banks
1040_1_6c_H2_V2 0
1040_1_6c_H3_V2 A97 08 1904
1040_1_6c_H4_V2 Si-Law
1040_1_6c_H5_V2 12
1040_1_6c_H6_V2
1040_1_6c_H1_V3 Hunter Bell
1040_1_6c_H2_V3 0
1040_1_6c_H3_V3 A39 26 756
1040_1_6c_H4_V3 Da-Law
1040_1_6c_H5_V3 2
1040_1_6c_H6_V3
1040_1_6c_H1_V4
1040_1_6c_H2_V4 0
1040_1_6c_H3_V4
1040_1_6c_H4_V4
1040_1_6c_H5_V4
1040_1_6c_H1_V5
1040_1_6c_H2_V5 0
1040_1_6c_H3_V5
1040_1_6c_H4_V5
1040_1_6c_H5_V5
1040_1_6c_H1_V6
1040_1_6c_H2_V6 0
1040_1_6c_H3_V6
1040_1_6c_H4_V6
1040_1_6c_H5_V6
1040_1_6d 0
1040_1_6e 9
1040_1_7 3878
1040_1_8a
1040_1_8b
1040_1_9
1040_1_10
1040_1_11
1040_1_12 0
1040_1_13
1040_1_14
1040_1_15
1040_1_16a
1040_1_16b
1040_1_17a
1040_1_17b
1040_1_18
1040_1_19
1040_1_20
1040_1_21a
1040_1_21b
1040_1_22_H1 Travel allowance
1040_1_22_H2 0
1040_1_23 3878
1040_1_24
1040_1_25a 2574
1040_1_25b
1040_1_26
1040_1_27
1040_1_28
1040_1_29_V1
1040_1_29_H1_V2
1040_1_29_H2_V2
1040_1_30 2574
1040_1_31 1303
```

Figure 22: Listing of a reference file corresponding to the form displayed on the previous page.

```
1040_1
1040_1_L_H1_V1 July
1040_1_L_H2_V1 July
1040_1_L_H3_V1 88
1040_1_L_H1_V2 BnairerndA.&ErskinW.Mitchell
1040_1_L_H2_V2 A11881384
1040_1_L_H1_V3 99225LeeStret
1040_1_L_H2_V3 A59021948
1040_1_L_H1_V4 Russell,NJ61920
1040_1_L_H1_V5 1
1040_1_L_H2_V5 0
1040_1_L_H1_V6 1
1040_1_L_H2_V6 0
1040_1_1 1
1040_1_2 0
1040_1_3_H1 0
1040_1_3_H2
1040_1_4_H1 0
1040_1_4_H2
1040_1_5_H1 0
1040_1_5_H2
1040_1_6a 1
1040_1_6b_H1 0
1040_1_6b_H2 1
1040_1_6c_H1_V1 RiderHarlan
1040_1_6c_H2_V1 0
1040_1_6c_H3_V1 A97203760
1040_1_6c_H4_V1 Aunt
1040_1_6c_H5_V1 6
1040_1_6c_H6_V1 8
1040_1_6c_H1_V2 TulaneBanks
1040_1_6c_H2_V2 0
1040_1_6c_H3_V2 A97081904
1040_1_6c_H4_V2 Si-Law
1040_1_6c_H5_V2 12
1040_1_6c_H6_V2
1040_1_6c_H1_V3 HunterBell
1040_1_6c_H2_V3 0
1040_1_6c_H3_V3 A9326
1040_1_6c_H4_V3 Da-Law
1040_1_6c_H5_V3 2
1040_1_6c_H6_V3
1040_1_6c_H1_V4
1040_1_6c_H2_V4 0
1040_1_6c_H3_V4
1040_1_6c_H4_V4
```

```
1040_1_6c_H5_V4
1040_1_6c_H1_V5
1040_1_6c_H2_V5 0
1040_1_6c_H3_V5
1040_1_6c_H4_V5
1040_1_6c_H5_V5
1040_1_6c_H1_V6
1040_1_6c_H2_V6 0
1040_1_6c_H3_V6
1040_1_6c_H4_V6
1040_1_6c_H5_V6
1040_1_6d 0
1040_1_6e 9
1040_1_7 3873
1040_1_8a
1040_1_8b
1040_1_9
1040_1_10
1040_1_11
1040_1_12
1040_1_13
1040_1_14
1040_1_15
1040_1_16a
1040_1_16b
1040_1_17a
1040_1_17b
1040_1_18
1040_1_19
1040_1_20
1040_1_21a
1040_1_21b
1040_1_22_H1 Travelalbewance
1040_1_22_H2 0
1040_1_23 3878
1040_1_24
1040_1_25a 25174
1040_1_25b
1040_1_26
1040_1_27
1040_1_28
1040_1_29_V1
1040_1_29_H1_V2
1040_1_29_H2_V2
1040_1_30 2574
1040_1_31 03
```

Figure 23: Listing of a hypothesis file corresponding to the completed form.

```
1040_1
1040_1_L_H1_V1 0 0 0 0
1040_1_L_H2_V1 0 0 0 0
1040_1_L_H3_V1 0 0
1040_1_L_H1_V2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1040_1_L_H2_V2 0 0 0 0 0 0 0 0 0
1040_1_L_H1_V3 0 0 0 0 0 0 0 0 0 0 0 0 0
1040_1_L_H2_V3 0 0 0 0 0 0 0 0 0
1040_1_L_H1_V4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1040_1_L_H1_V5 0
1040_1_L_H2_V5 0
1040_1_L_H1_V6 0
1040_1_L_H2_V6 0
1040_1_1 0
1040_1_2 0
1040_1_3_H1 0
1040_1_3_H2
1040_1_4_H1 0
1040_1_4_H2
1040_1_5_H1 0
1040_1_5_H2
1040_1_6a 0
1040_1_6b_H1 0
1040_1_6b_H2 0
1040_1_6c_H1_V1 0 0 0 0 0 0 0 0 0 0 0
1040_1_6c_H2_V1 0
1040_1_6c_H3_V1 0 0 0 0 0 0 0 0 0
1040_1_6c_H4_V1 0 0 0 0
1040_1_6c_H5_V1 0
1040_1_6c_H6_V1 0
1040_1_6c_H1_V2 0 0 0 0 0 0 0 0 0 0 0
1040_1_6c_H2_V2 0
1040_1_6c_H3_V2 0 0 0 0 0 0 0 0 0
1040_1_6c_H4_V2 0 0 1 0 0 0
1040_1_6c_H5_V2 0 0
1040_1_6c_H6_V2
1040_1_6c_H1_V3 0 0 0 0 0 0 0 0 0 0
1040_1_6c_H2_V3 0
1040_1_6c_H3_V3 0 0 0 0 0
1040_1_6c_H4_V3 0 0 1 0 0 0
1040_1_6c_H5_V3 0
1040_1_6c_H6_V3
1040_1_6c_H1_V4
1040_1_6c_H2_V4 0
1040_1_6c_H3_V4
1040_1_6c_H4_V4
```

```
1040_1_6c_H5_V4
1040_1_6c_H1_V5
1040_1_6c_H2_V5 0
1040_1_6c_H3_V5
1040_1_6c_H4_V5
1040_1_6c_H5_V5
1040_1_6c_H1_V6
1040_1_6c_H2_V6 0
1040_1_6c_H3_V6
1040_1_6c_H4_V6
1040_1_6c_H5_V6
1040_1_6d 0
1040_1_6e 0
1040_1_7 0 0 0 0
1040_1_8a
1040_1_8b
1040_1_9
1040_1_10
1040_1_11
1040_1_12
1040_1_13
1040_1_14
1040_1_15
1040_1_16a
1040_1_16b
1040_1_17a
1040_1_17b
1040_1_18
1040_1_19
1040_1_20
1040_1_21a
1040_1_21b
1040_1_22_H1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1040_1_22_H2 0
1040_1_23 0 0 0 0
1040_1_24
1040_1_25a 0 0 0 0 0
1040_1_25b
1040_1_26
1040_1_27
1040_1_28
1040_1_29_V1
1040_1_29_H1_V2
1040_1_29_H2_V2
1040_1_30 0 0 0 0
1040_1_31 0 0
```

Figure 24: Listing of a rejection file corresponding to the completed form.

1040_1
1040_1_L_H1_V1 0.85 0.86 0.90 0.81
1040_1_L_H2_V1 0.84 0.89 0.94 0.90
1040_1_L_H3_V1 0.99 0.83
1040_1_L_H1_V2 0.83 0.85 0.85 0.84 0.91 0.94 0.90 0.90 0.98 0.92 0.93 0.89 0.87 0.88 0.82 0.80 0.90 0.81 0.99 0.97 0.94 0.83 0.83 0.93 0.96 0.95 0.98 0.81
1040_1_L_H2_V2 0.92 0.92 0.90 0.94 0.87 0.87 0.82 0.81 0.89
1040_1_L_H1_V3 0.99 0.80 0.95 0.80 0.84 0.95 0.83 0.88 0.91 0.97 0.92 0.95 0.83
1040_1_L_H2_V3 0.93 0.90 0.91 0.90 0.98 0.82 0.84 0.82 0.92
1040_1_L_H1_V4 0.99 0.98 0.99 0.97 0.87 0.97 0.93 0.94 0.99 0.98 0.89 0.90 0.99 0.90 0.94
1040_1_L_H1_V5 0.99
1040_1_L_H2_V5 0.91
1040_1_L_H1_V6 0.88
1040_1_L_H2_V6 0.92
1040_1_1 0.82
1040_1_2 0.83
1040_1_3_H1 0.98
1040_1_3_H2
1040_1_4_H1 0.89
1040_1_4_H2
1040_1_5_H1 0.85
1040_1_5_H2
1040_1_6a 0.80
1040_1_6b_H1 0.91
1040_1_6b_H2 0.99
1040_1_6c_H1_V1 0.87 0.89 0.88 0.90 0.83 0.99 0.94 0.92 0.95 0.98 0.90
1040_1_6c_H2_V1 0.80
1040_1_6c_H3_V1 0.82 0.81 0.98 0.87 0.85 0.85 0.84 0.91 0.95
1040_1_6c_H4_V1 0.94 0.89 0.82 0.99
1040_1_6c_H5_V1 0.98
1040_1_6c_H6_V1 0.89
1040_1_6c_H1_V2 0.90 0.92 0.91 0.98 0.94 0.84 0.98 0.87 0.87 0.80 0.84
1040_1_6c_H2_V2 0.84
1040_1_6c_H3_V2 0.98 0.99 0.99 0.99 0.93 0.94 0.98 0.99 0.93
1040_1_6c_H4_V2 0.83 0.80 0.78 0.93 0.90 0.92
1040_1_6c_H5_V2 0.90 0.91
1040_1_6c_H6_V2
1040_1_6c_H1_V3 0.89 0.83 0.90 0.94 0.94 0.93 0.99 0.91 0.98 0.80
1040_1_6c_H2_V3 0.88
1040_1_6c_H3_V3 0.92 0.94 0.95 0.92 0.91
1040_1_6c_H4_V3 0.99 0.99 0.03 0.93 0.95 0.99
1040_1_6c_H5_V3 0.92
1040_1_6c_H6_V3
1040_1_6c_H1_V4
1040_1_6c_H2_V4 0.93
1040_1_6c_H3_V4
1040_1_6c_H4_V4
1040_1_6c_H5_V4
1040_1_6c_H1_V5
1040_1_6c_H2_V5 0.93
1040_1_6c_H3_V5
1040_1_6c_H4_V5
1040_1_6c_H5_V5
1040_1_6c_H1_V6
1040_1_6c_H2_V6 0.99
1040_1_6c_H3_V6
1040_1_6c_H4_V6
1040_1_6c_H5_V6
1040_1_6d 0.99
1040_1_6e 0.91
1040_1_7 0.89 0.89 0.82 0.88
1040_1_8a
1040_1_8b
1040_1_9
1040_1_10
1040_1_11
1040_1_12
1040_1_13
1040_1_14
1040_1_15
1040_1_16a
1040_1_16b
1040_1_17a
1040_1_17b
1040_1_18
1040_1_19
1040_1_20
1040_1_21a
1040_1_21b
1040_1_22_H1 0.92 0.84 0.85 0.84 0.80 0.98 0.92 0.92 0.90 0.95 0.96 0.96 0.98 0.87 0.87
1040_1_22_H2 0.80
1040_1_23 0.90 0.81 0.84 0.85
1040_1_24
1040_1_25a 0.99 0.93 0.98 0.98 0.82
1040_1_25b
1040_1_26
1040_1_27
1040_1_28
1040_1_29_V1
1040_1_29_H1_V2
1040_1_29_H2_V2
1040_1_30 0.92 0.84 0.84 0.89
1040_1_31 0.87 0.86

Figure 25: Listing of a confidence file corresponding to the completed form.

```
Summary:
  TOTALS  ( output=FCItdA,of=form.sum,cf=form.fct )

Draft standard measures:
Accumulators: TP=1648 FP=43 M=36 RT=45 RF=18 RM=164
 Character recognition decision:
  :                    accuracy:  88.8410%   ( 1648 / 1855 )
  :         accuracy (form right):  97.4571%   ( 1648 / 1691 )
 Character output:
  :                    accuracy:  98.4644%   ( 1603 / 1628 )
 Field accuracy:
  :       accuracy (including icons):  81.2762%   ( 777 / 956 )

Character rejection rates:
  :                        all:  3.3475%   ( 63 / 1882 )
  :              all hypotheses:  3.7256%   ( 63 / 1691 )
  :                    matches:  2.7306%   ( 45 / 1648 )
  :               substitutions:  44.1176%   ( 15 / 34 )
  :                  insertions:  33.3333%   ( 3 / 9 )
  :          all (due to form type):  8.7141%   ( 164 / 1882 )

Fields (excluding icons):
  :                    accuracy:  81.7010%   ( 634 / 776 )
  :       accuracy (with form right):  90.1849%   ( 634 / 703 )
  :       rejected (due to form type):  9.4072%   ( 73 / 776 )
  :       deleted (due to form wrong):  0.0000%   ( 0 / 776 )

Fields (including icons):
  :                    accuracy:  81.2762%   ( 777 / 956 )
  :       accuracy (with form right):  89.8266%   ( 777 / 865 )
  :       rejected (due to form type):  9.5188%   ( 91 / 956 )
  :       deleted (due to form wrong):  0.0000%   ( 0 / 956 )

Characters:
  :                    accuracy:  85.1753%   ( 1603 / 1882 )
  :       accuracy (with form right):  94.7960%   ( 1603 / 1691 )
  :       rejected (due to form type):  8.7141%   ( 164 / 1882 )
  :       deleted (due to form wrong):  0.0000%   ( 0 / 1882 )

Icons:
  :                    accuracy:  79.4444%   ( 143 / 180 )
  :       accuracy (with form right):  88.2716%   ( 143 / 162 )
  :       rejected (due to form type):  10.0000%   ( 18 / 180 )
  :       deleted (due to form wrong):  0.0000%   ( 0 / 180 )

Form type identification:
  :                    accuracy:  90.9091%   ( 10 / 11 )
  :                failure rate:  9.0909%   ( 1 / 11 )
  :     accuracy (excluding rejected):  100.0000%   ( 10 / 10 )
  : failure rate (excluding rejected):  0.0000%   ( 0 / 10 )
  :                    rejected:  9.0909%   ( 1 / 11 )
```

Figure 26: Example of a Scoring Package summary report.

```
form type:
count: 11
  rejected: 1
  not rejected, right: 10
  not rejected, wrong: 0

icon fields:
count: 180
  form type rejected: 18
  form type wrong and not rejected: 0
  form type right and not rejected: 162
    right: 143
    wrong: 19
    rejected: 15
    not rejected: 147
    matches: 157
      rejected: 14
      not rejected: 143
    mismatches: 5
      rejected: 1
      not rejected: 4
    not present / not found: 115
    not present / found: 3
    present / not found: 2
    present / found: 42

character fields:
count: 776
  form type rejected: 73
  form type wrong and not rejected: 0
  form type right and not rejected: 703
    right: 634
    wrong: 69

characters:
  in alignments: 1891
  hypothesis: 1691
  reference: 1882
    form type rejected: 164
    form type wrong and not rejected: 0
    form type right and not rejected: 1691
      rejected: 63
      not rejected: 1628
      correct: 1648
        rejected: 45
        not rejected: 1603
      substitutions: 34
        rejected: 15
        not rejected: 19
      insertions: 9
        rejected: 3
        not rejected: 6
      deletions: 36

Accumulators: TP=1648 FP=43 M=36 RT=45 RF=18 RM=164
```

Figure 27: Example of a Scoring Package fact sheet.