DATABASE SYNCHRONIZATION DESIGN REPORT

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

Aaron R. DeWispelare
Joseph H. Cooper
Robert L. Marshall

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

May 1995

3/18

CONTENTS

4/18

FIGURES

5/18

TABLES

6/18

ABBREVIATIONS

CNWRA	Center for Nuclear Waste Regulatory Analyses
DWM	Division of Waste Management
FTP	File Transfer Protocol
HLW	High-Level [Radioactive] Waste
ID	Identifier
NRC	Nuclear Regulatory Commission
NUDOCS	Nuclear Document System
OCR	Optical Character Recognition
RPC	Remote Procedure Call
TDOCS	Technical Reference Document Database System

7/18

ACKNOWLEDGMENTS

This report was prepared to document work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract NRC-02-93-005. The activities reported here were performed on behalf of the NRC Division of Waste Management (DWM). The report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

The following products are discussed in this report:

- Solaris is a trademark of Sun Microsystems, Inc.

  Sun Microsystems, Inc.
  2550 Garcia Avenue
  Mountain View, CA 94043

- TOPIC is a trademark of Verity

  Verity Inc.
  1550 Plymouth Street
  Mountain View, CA 94303

8/18

1 INTRODUCTION

1.1 BACKGROUND

The technical staff of the Nuclear Regulatory Commission (NRC), Division of Waste Management (DWM), performs technical reviews and analyses in conducting required prelicensing and licensing activities in support of the high-level waste (HLW) repository program. These technical reviews require reference to, and use of, large volumes of technical information in the form of technical data and technical documents that must be available for referencing, synthesizing, and incorporating in the staff analyses. The Technical Document Reference Database (TDOCS) was designed as a text and image database with full-text search and retrieval capabilities to permit staff members to have all the necessary information available to them through their individual computer workstations (Johnson et al., 1993).

The initial implementation of the TDOCS at the Center for Nuclear Waste Regulatory Analyses (CNWRA) was completed in FY94. This initial implementation also functions as the CNWRA document management system for such holdings as quality assurance records, correspondence indexing, and technical document bibliographic records. TDOCS will be installed at the DWM in FY95. However, TDOCS will not become an official repository of documents since that is the purpose of the Nuclear Document System (NUDOCS) within the NRC.

The existence of parallel TDOCS systems at the DWM and CNWRA, each with its own scanning and document loading capabilities, implies the need for system capabilities that assure documents required at both locations (e.g., those related to HLW) are exchanged appropriately. Therefore, database synchronization facilities and procedures are required.

1.2 OVERVIEW OF SYSTEM DESIGN

The TDOCS is a prototype application that uses relatively new technology to capture a broad spectrum of technical materials and to make them available to the DWM and CNWRA staffs (DeWispelare et al., 1995a). Capabilities for scanning and loading electronic copies of documents are provided at each site, along with full-text search and retrieval and image viewing facilities. Technical documents submitted to the scanning station for routine loading are scanned, processed for optical character recognition (OCR), and edited in a document cleanup process to correct scanning or OCR errors. Output from the scanning, OCR, and cleanup processes includes a text file for the entire document and image files for each page. A bibliographic header is also prepared for each document and associated with the text and image files. Alternatively, electronic copies of documents may be submitted for loading. A bibliographic header is prepared and submitted with the electronic copy of the document. The header for each document, along with the document's text and image files, is submitted to a batch process that loads the documents in an unobtrusive manner on a scheduled basis during the evening hours. The header for each document is loaded into a relational database, and text and image files are loaded and indexed in the full-text database.

Document loading processes will occur asynchronously at the DWM and CNWRA. To avoid duplication of effort in scanning and loading the same documents at both sites, a loading plan will be published in FY95 (DeWispelare et al., 1995b) that identifies the classes of documents, documents to be shared and the prioritization of documents for loading and subsequently for sharing. To ensure that the designated documents to be shared between the databases in the two systems are the same, the software for

automatic document exchange and loading will be the same. This synchronization of the TDOCS databases at the DWM and CNWRA will permit staff at both sites to have access to the same technical materials.

To accomplish synchronization of the DWM and CNWRA databases, the local document submission processes at the DWM and CNWRA must identify documents loaded in the current cycle that are appropriate for sharing and that are needed at the remote site. Documents submitted at each local site that are selected for sharing must be copied and forwarded to the database synchronization process at the remote site for loading. Similarly, updates and deletions of shared documents must be synchronized.

This report provides the TDOCS database synchronization system design including issues, system design, and a proposed plan for implementation.

- Chapter 2 discusses database synchronization design
- Chapter 3 discusses implementation
- Chapter 4 lists references

9/18

2 DATABASE SYNCHRONIZATION DESIGN

This chapter describes the high-level design for the synchronization of the TDOCS databases at the DWM and CNWRA, which will be accomplished daily on a scheduled batch basis. New acquisitions may be acquired through scanning, OCR, and document cleanup or, alternatively, through loading electronic copies of documents. The new acquisitions, document updates, and document deletions from the DWM TDOCS system will be accumulated as electronic files and transmitted to the CNWRA. Similarly, the CNWRA TDOCS system will accumulate electronic copies of new acquisitions, document updates, and document deletions and transmit them to the DWM TDOCS system. When these synchronization transactions are received, they will be processed in much the way that normal electronic loading and document maintenance is handled to add, update, and delete documents in the corresponding TDOCS system:

- Document text and image files received from the remote system will be stored or updated in the local TDOCS repository.
- Bibliographic headers received from the remote system will be loaded or updated in the relational database.
- Full-text indexes and hyperlinks will be updated.

2.1 SINGLE-SITE DESIGN

The synchronization design is presented in the context of the current single-site processes. In the current processes, documents are submitted, updated, and deleted during the day to be batch processed for loading and maintenance in the relational database and full-text and image repositories. To be nonobtrusive to users, the batches are processed during the evening hours. The current loading context is presented in Section 2.1.1 followed by the proposed synchronization design in Section 2.3.

2.1.1 Current Design for Document Submission

The TDOCS uses a client/server architecture, in which database functions are performed on a central server computer and user interaction is supported through functionality implemented on individual client computers. This approach balances the processing load by distributing the client computing requirements to the individual users' personal computers or workstations. In addition to the general search and retrieval capabilities available to all users, selected users known as *custodians* are permitted to submit new documents, and update or delete existing documents.

In the present single-site implementation, custodian users submit documents for loading during the day through the routine loading process.

2.1.1.1 Client Processing for Document Submission

The client processing for document submission is illustrated in Figure 2-1:.

- The custodian enters a header for the document through the client header entry process.

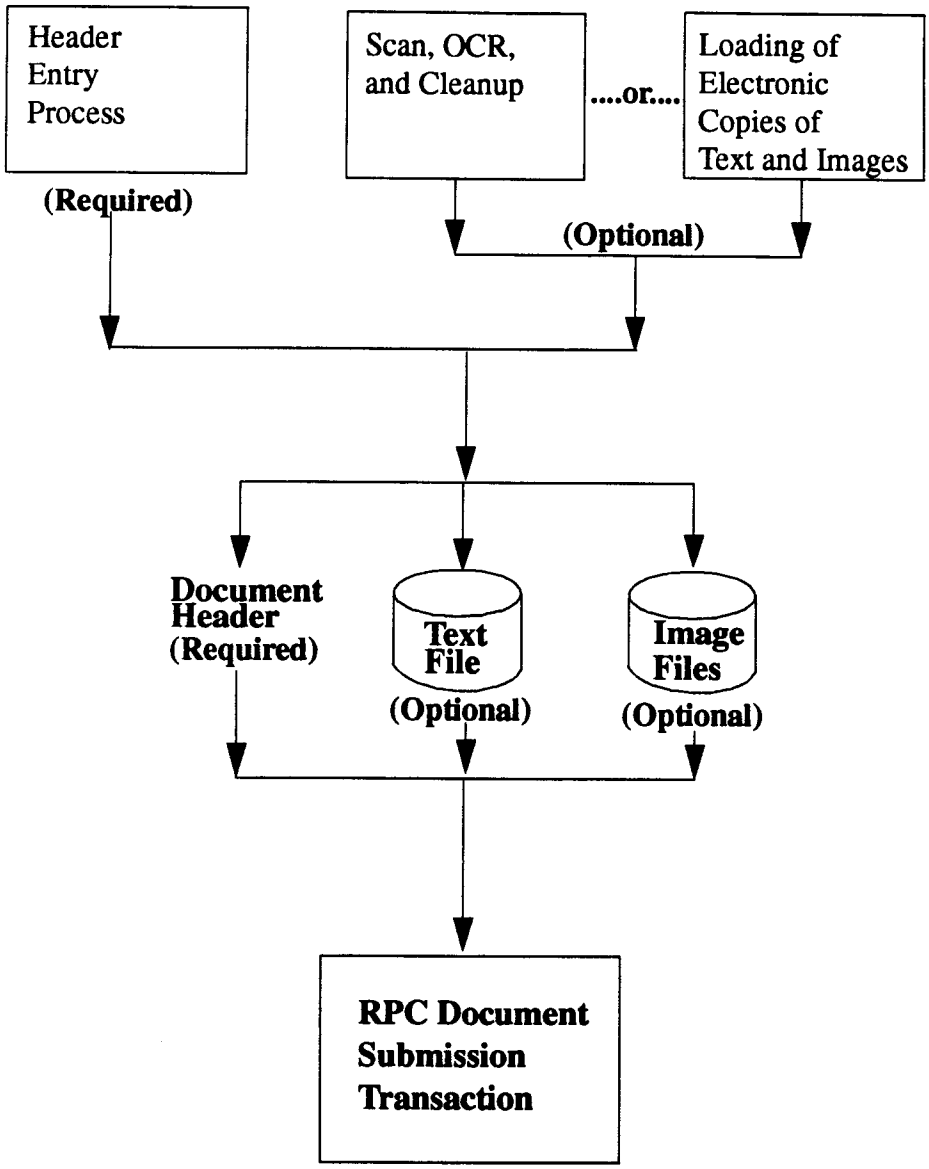

Figure 2-1. Client processing for a single-site document submission

10/18

- The custodian receives any text and image files for the document, either: (i) through scanning, OCR, and cleanup; or (ii) through electronic submission.

- All text and image files obtained by the custodian are stored in the *submit* directory on the client platform.

- The client document submission process transfers the completed header to the server using Remote Procedure Call (RPC)

2.1.1.2 Server Processing for Document Submission

The server uses the information in the RPC document submission transaction to prepare the document for loading into the TDOCS repository through the batch process. The effect of the server processing the document submission is: (i) to accumulate headers in the relational database, and (ii) to rename and store text and image files in the server's temporary *incoming* document set directories. The directory structure used by the server for input files during document submission and updating is illustrated in Figure 2-2:

- An external document number is generated as part of the header. The external document number is maintained uniquely across all TDOCS databases and serves to identify permanently the document.

- An internal document identifier (ID) is generated, stored as part of the header, and used to rename the text and image files associated with the document. The internal document ID is unique within each TDOCS database but may be duplicated between sites.

- The server stores the completed header in the relational database and marks it for submission.

- The server uses File Transfer Protocol (FTP) to move any text and image files from the client *submit* directory to a temporary *incoming* directory in the server file system.

- All text and image files are renamed, using the internal document ID, and moved from the temporary *incoming* directory to the subdirectory for the appropriate document set.

- The server returns the document number to the client through an RPC reply.

2.1.2 Current Design for Document Updates

In the present single-site implementation, custodian users update documents during the day. The system accumulates these document update transactions for batch processing during the evening hours. TDOCS document updates are always handled as complete replacements of the existing document header record and, optionally, the document text and image files. Therefore, document updates may take one of two forms:

- Header-only updates—in which only the document header is changed, and no change is made to the text or image files

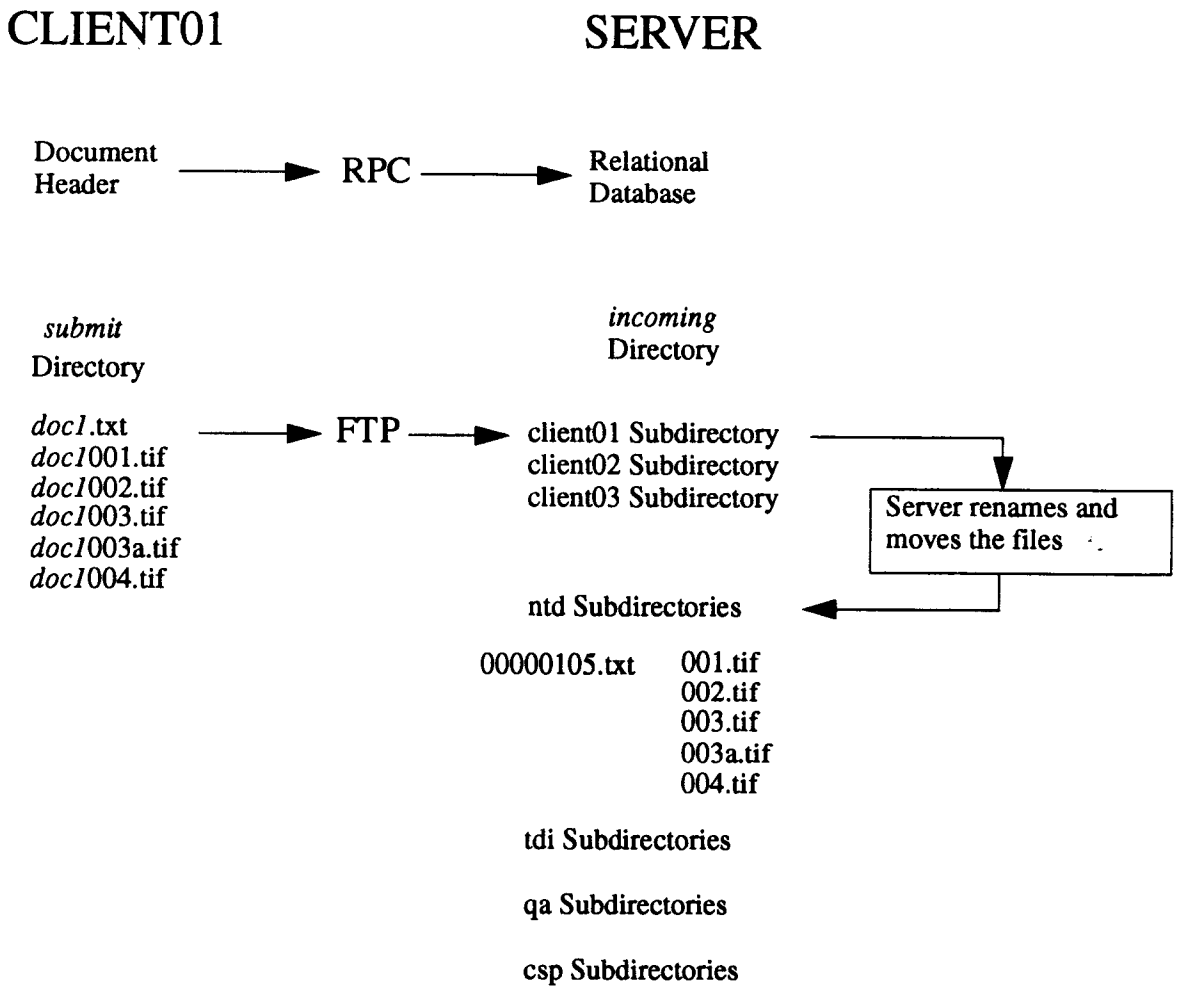

Figure 2-2. Directory structure for document submission and updating

- Full-document updates—in which the text and/or image files are changed

Header updates affect only the header stored in the relational database, but full-document updates require complete replacement of the header in the relational database as well as all text and image files.

2.1.2.1 Client Processing for Document Updates

The client processing for document updates begins with the retrieval and update of the document header when:

11/18

- The custodian enters the external document number for the desired document through an entry process.

- The external document number is transferred to the server as an RPC transaction.

- The server retrieves the document header record from the relational database and returns it to the client code through an RPC reply.

- The custodian changes the header information and selects the appropriate push-button on the screen to complete or cancel the document update transaction.

  If the *Update* push-button is selected from the user interface, the updated header information is transferred to the server through RPC as a header-only update transaction.

  If the *Import* push-button is selected from the user interface, a screen is displayed to permit the custodian either to initiate scanning or to specify the locations and types of the revised text and image files:

  — The custodian obtains any text and image files for the document either: (i) through scanning, OCR, and cleanup; or (ii) through electronic submission.

  — All text and image files obtained by the custodian are stored in the *submit* directory on the client platform.

  — The custodian selects the *OK* push-button, and the updated header information is transferred to the server through RPC as a full-document-update transaction.

2.1.2.2 Server Processing for Document Updates

The server processes information in the document update RPC transaction, updates the header record in the relational database, and prepares any updated text and image files for loading into the TDOCS repository through the batch process. The effect of the server's processing the document update is: (i) to revise headers in the relational database, and (ii) to rename and store text and image files in the server's temporary *incoming* document set directories:

- The server uses the external document number to obtain the internal document ID, stores the revised header in the relational database, and marks it to be updated.

- The server uses FTP to move any text and image files from the client *submit* directory to a temporary *incoming* directory in the server file system.

- All text and image files are renamed using the internal document ID and moved from the temporary *incoming* directory to the subdirectory for the appropriate document set.

- The server returns the completion status to the client through an RPC reply.

2.1.3 Current Design for Document Deletions

In the present single-site implementation, custodian users delete documents throughout the day. The system accumulates these document deletion transactions for batch processing during the evening hours.

2.1.3.1 Client Processing for Document Deletions

The client processing for document deletions begins with the retrieval of the document header:

- The custodian enters the external document number for the desired document through an entry process.
- The external document number is transferred to the server as an RPC transaction.
- The server retrieves the document header record from the relational database and returns it to the client code through an RPC reply.
- The client displays header information to the custodian and waits for confirmation.
- The custodian indicates confirmation by selecting the *OK* push-button or terminates the transaction by selecting the *Cancel* push-button.

If the *OK* push-button is selected from the user interface, the client formats and transmits a record deletion RPC transaction to the server.

2.1.3.2 Server Processing for Document Deletions

The server processes information in the document deletion RPC transaction and marks the header record in the relational database for deletion from the TDOCS repository through the batch process:

- The server uses the external document number to obtain the internal document ID, stores the revised header in the relational database, and marks it for deletion.
- The server returns the completion status to the client through an RPC reply.

2.2 CURRENT DESIGN FOR THE BATCH PROCESSING

To prevent interference for users resulting from re-indexing of the full-text database during normal hours of operation, certain functions are reserved for the evening and are referred to as batch processing.

Headers in the relational database that have been marked for document submission, update, and deletion control the nightly batch-processing cycle. The marked headers in the relational database and the files associated with them in the appropriate *incoming* document set subdirectories serve as an input queue for the batch process. Each night during off-hours, the batch process is initiated to update, submit, and delete documents.

12/18

The batch process retrieves and processes marked header records from the relational database in a specified sequence:

- Updates—headers marked for update are processed first.
- Submissions—headers marked for submission are processed following the processing of all updates.
- Deletions—headers marked for deletion are processed following all update and submission processing.

2.2.1 Directory Structures Supporting the Full-Text Database

To support the full-text search and retrieval functionality of the TDOCS, specialized text and image files must be created and maintained. Figure 2-3 illustrates the directory structure for these text and image files:

- *archive* directory—contains the text of each document as it was originally submitted or last updated, and any binary format word processing files
- *images* directory—contains the current full-page images for each document along with any other images for figures, equations, etc.
- *data* directory—contains a *link* subdirectory for formatted files used as input for the TOPIC index and link utilities and a *load* subdirectory that contains output files following the execution of the TOPIC load utility

2.2.2 Batch Update of Documents

The batch process transfers the accumulated files for documents marked for update from the temporary location in the *incoming* directory of the server file system and prepares them for loading into the full-text and image repositories through the following steps, as illustrated in Figure 2-4:

- All document header records in the relational database that are marked for update are retrieved.
- For each selected document in which only the header is being updated, any old text and image files are moved back to the appropriate *incoming* directory.
- For each selected document in which text and/or image files along with the header are being updated, the old files are deleted.
- The text file and the binary word processing format file, if any, are retrieved from the temporary area in the *incoming* directory of the file system and copied to the appropriate *archive* directory.

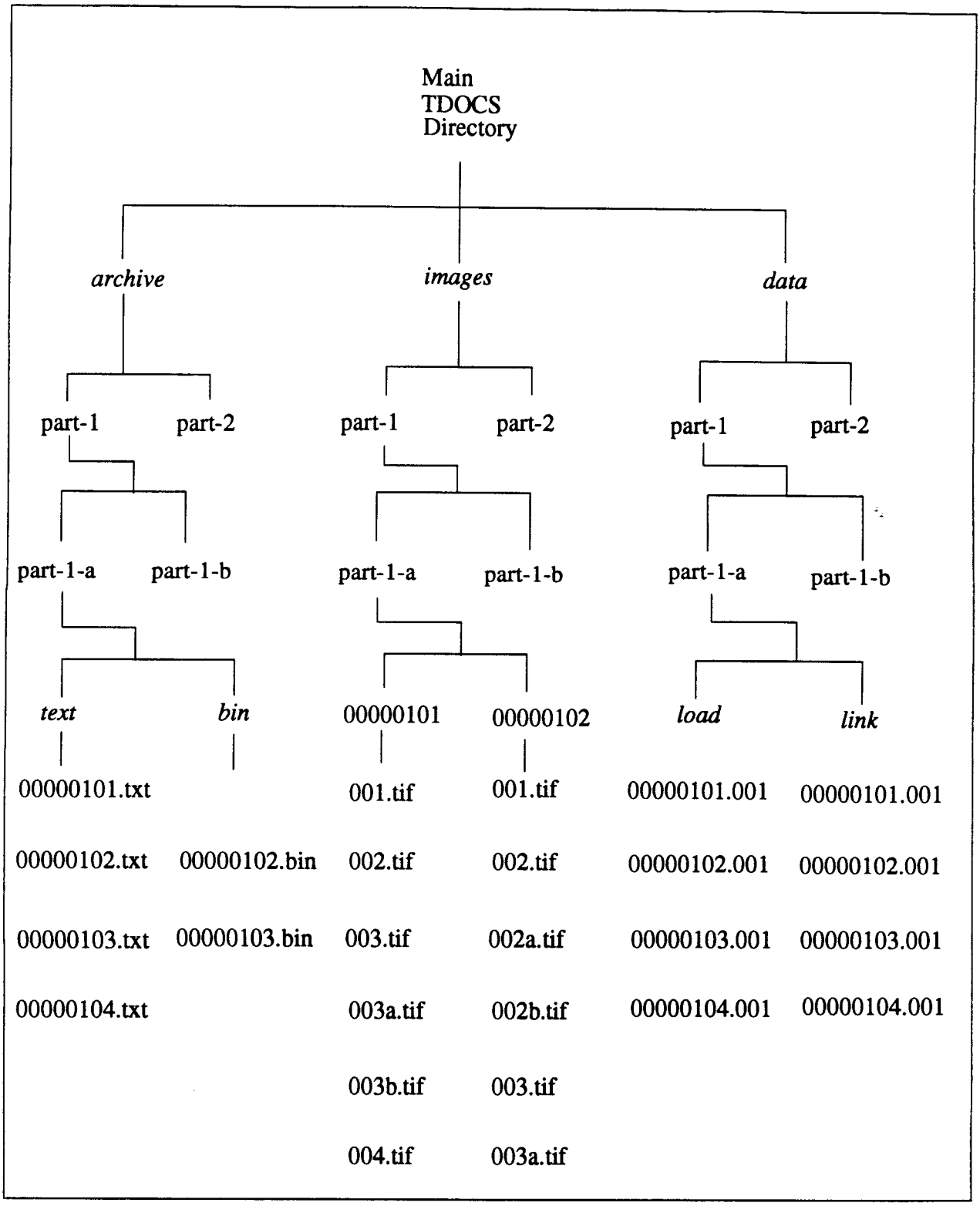

Figure 2-3. Directory structures in the UNIX file system

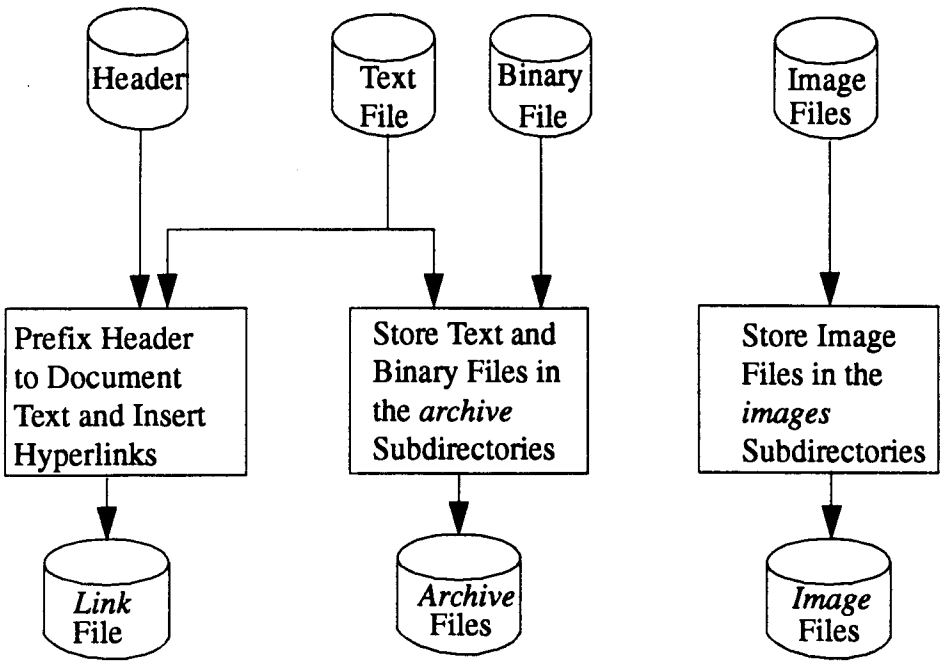

Figure 2-4. Single-site batch processing for document updates

- A formatted copy of the updated header is prefixed to the text file to permit the TOPIC full-text search and retrieval system to parse and store the header fields. Formatted flags are also inserted into the document to indicate the locations of the hyperlinks for display of the full-page images. This formatted file is stored in the appropriate *link* directory.

- Any image files are moved to the appropriate *images* directory.

The effect of the batch processing for updates is to prepare the input files for the updated documents so that they can be indexed for full-text search and retrieval.

2.2.3 Batch Submission of Documents

The batch process transfers the accumulated files for documents that are marked for submission from the temporary location in the server file system to the full-text and image repositories through the following steps illustrated in Figure 2-5:

- All document header records in the relational database that are marked for submission are retrieved.

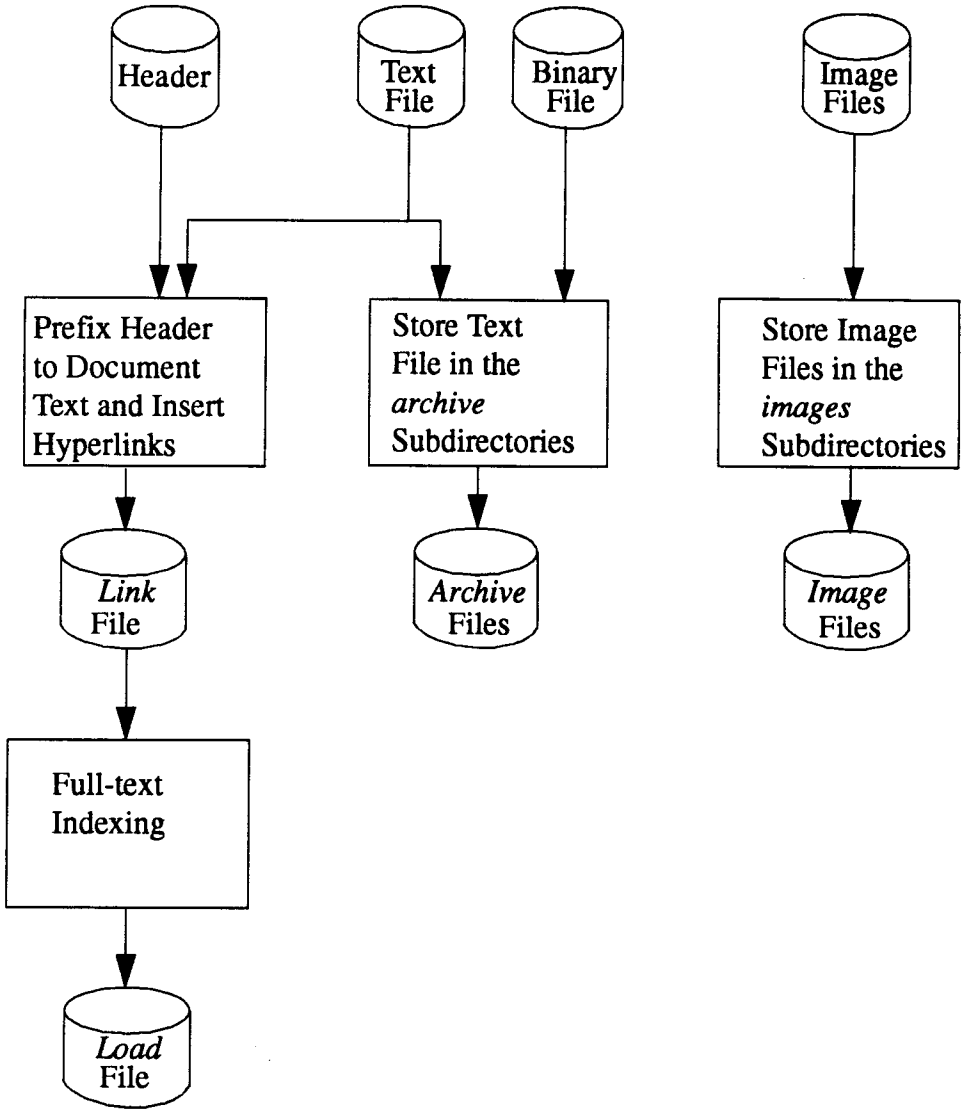

Figure 2-5. Single-site batch processing for document submissions

- For each selected document header, the text file and any binary word processing formatted file are retrieved from the temporary area in the file system and copied to the appropriate *archive* directory.

- A formatted copy of the header is prefixed to the text file to permit the TOPIC full-text search and retrieval system to parse and store the header fields. Formatted flags are also inserted into

the document to indicate the locations of the icons for display of the full-page images. This formatted file is stored in the appropriate *link* directory.

- All submitted image files are moved to the appropriate *images* directory.

- After all selected documents have been formatted and moved to the *archive*, *images*, and *link* directories, the TOPIC load process is started. This process indexes all prepared documents, including those prepared for update and those prepared for submission, for full-text search and retrieval.

- If the TOPIC indexing step is successful, all the selected headers are updated in the relational database to indicate that the documents have been processed.

- After completion of the document submission and TOPIC indexing functions, the document text and image files are deleted from the temporary area in the *incoming* directory of the server file system.

2.2.4 Batch Deletion of Documents

The batch process, illustrated in Figure 2-6, retrieves the headers for all documents marked for deletion from the relational database, deletes the text and image files from the server file system, and removes all references to the document from the TOPIC full-text search and retrieval indexes. All document header records in the relational database that are marked for deletion are retrieved.

- For each selected header, all text and image files are removed from the file system.

- A TOPIC utility program is executed to remove references to the deleted document from the full-text search and retrieval indexes.

- Records marked for deletion are updated and marked as deleted when the process is successfully completed.

A loading report is prepared and forwarded to the system administrator by electronic mail (E-Mail), including all submitted, updated, and deleted documents.

2.3 PROPOSED SYNCHRONIZATION DESIGN

The overall approach to database synchronization is to provide a mechanism through which each local database may insert documents to be shared into the input queue for batch processing on the other system. Thus, as documents are processed for submission, update, or deletion during the day at the DWM, those documents that are to be shared will be transferred to the input queue for batch processing at the CNWRA. Similarly, as documents are processed for submission, update, or deletion during the day at the CNWRA, those documents that are to be shared will be transferred to the input queue for batch processing at the DWM. The advantage of this approach is that daily activities at each local site are propagated to the remote site automatically. Since the synchronization occurs by inserting documents into the batch input queue of the remote system, the process is completely transparent to the batch process on the remote system. No modifications to the existing batch processes are required.

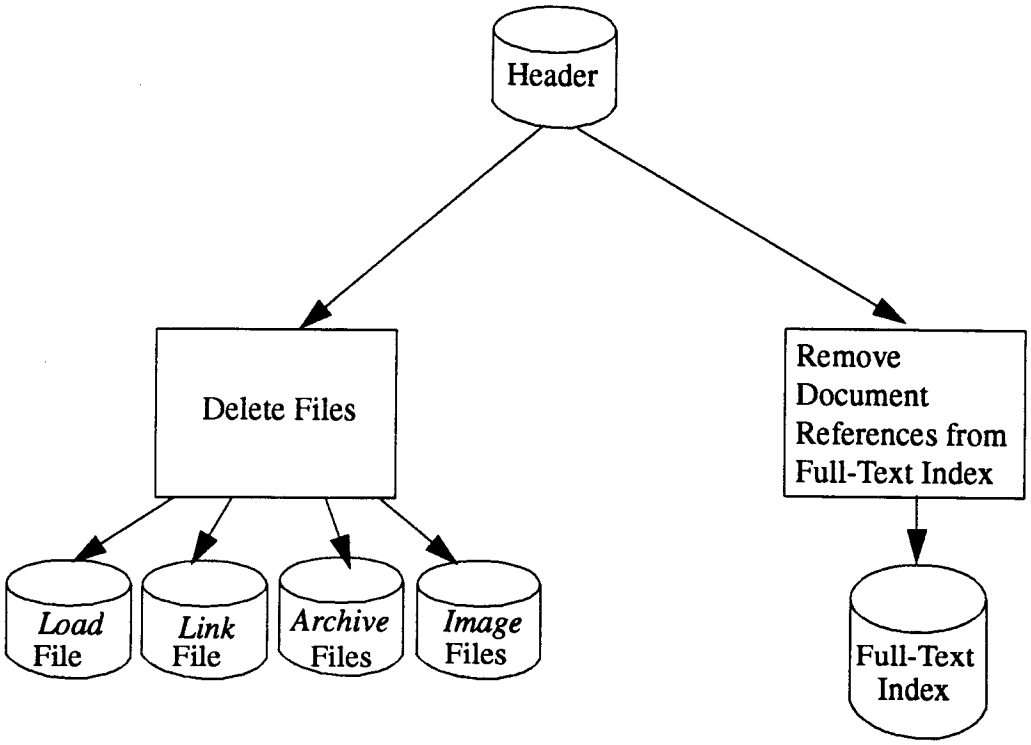

Figure 2-6. Single-site batch processing for document deletions

Implementation of this database synchronization approach requires that each local system has an output queue to hold documents selected for synchronization with its corresponding remote system. As documents are processed for submission, update, or deletion during the day, the headers, text files, and image files for documents to be shared are copied to the output queue. The output queue for each local system is visible to the server of the corresponding remote system. Periodically, the server in each system activates a process that copies any accumulated documents from the output queue of the corresponding system and inserts those documents in the local input queue for the batch process.

The complete synchronized document loading process is illustrated in Figure 2-7. Documents are copied throughout the day from the output queue of each system to the input queue for the batch process on the other system. This process also performs many of the same functions that would occur in a local document submission update or deletion. The result is an entry in the local batch input queue that is indistinguishable from similar entries generated through local document processing.

2.3.1 Proposed Synchronization Design for Document Submission

Submission transactions are affected only by whether or not the submitted document is to be shared. If the document is not to be shared, it is submitted to the local database, and no submission transaction is queued for the remote database. The actions required to properly handle document

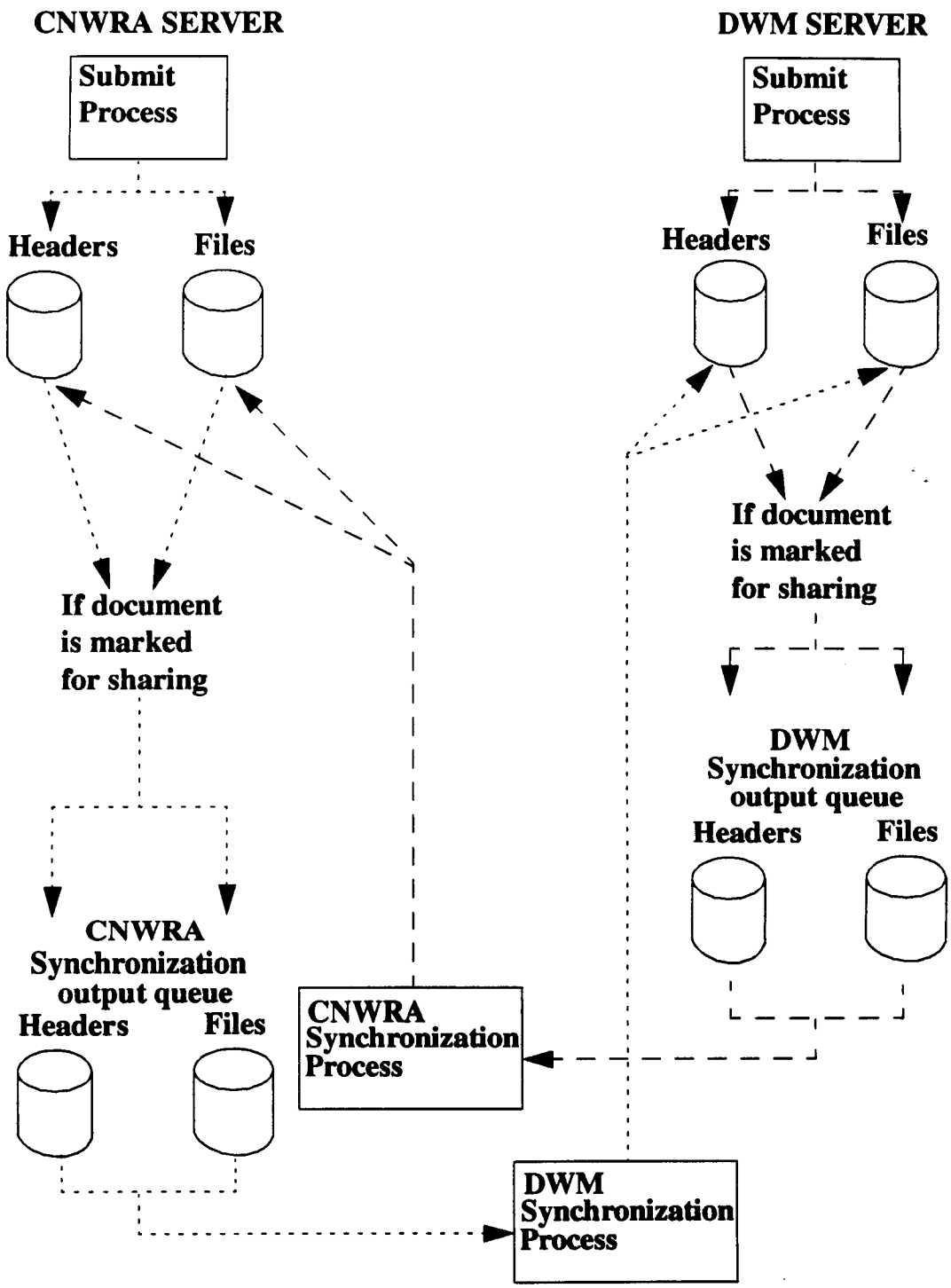

Figure 2-7. Synchronization process

submission transactions are summarized in Table 2-1. Submitted documents to be shared are stored in the local system's output queue. The synchronization process in the remote system periodically accesses the local system's output queue and performs the following steps to move the submitted document to the remote system's input queue for the batch process:

- The text and image files are moved from the output queue to the server's *incoming* directory.
- The relational database is accessed to obtain the next available internal document ID.
- The text and image files are renamed using the new internal document number.
- The header is stored in the relational database including the new internal document ID.
- The renamed files are moved to the appropriate *incoming* document set subdirectory.
- The header, text, and image information is removed from the output queue so that it will not be copied repetitively.

During the next execution of the batch process, the accumulated document records in each local input queue, including both local and synchronization submissions, are added to their respective local TDOCS repositories as described in Section 2.2.3.

Table 2-1. Required processing for synchronization of document submission transactions

Sharing Status	Actions Required In The Local Database	Actions Required In The Remote Database
Shared	Submit the document to the local database Copy the document submission transaction to the local output queue	Copy the submission transaction from the output queue of the local system to the input queue of the remote system and submit it to the remote database through the batch process
Unshared	Submit the transaction only to the local database	No action required in the remote database

2.3.2 Proposed Synchronization Design for Document Updates

When documents are maintained in one repository of a synchronized database system, synchronization transactions must be generated to assure corresponding updates occur in the remote database. If the document status requires sharing both before and after the update, the update transaction is copied to the output queue of the local system for synchronization. The remote system then accesses the output queue and moves the document to the batch input queue on the remote system:

- The text and image files are moved from the output queue to the server's *incoming* directory.
- The relational database is accessed using the document's external document number to obtain the remote system's internal ID for that document.

16/18

- The text and image files are renamed using the document's internal ID.
- The header is stored in the relational database.
- The renamed files are moved to the appropriate document set subdirectory.

Document updates are more complex than document submissions because they are affected by both the prior and updated sharing status of the document. The actions required to handle document update transactions properly are summarized in Table 2-2.

2.3.3 Proposed Synchronization Design for Document Deletions

When documents are maintained in one repository of a synchronized database system, synchronization transactions must be generated to assure that corresponding deletions occur in the other database. Table 2-3 summarizes the actions required for deleting shared and unshared documents. If the status of the document to be deleted indicates that it has been shared with the other database, the document deletion transaction is copied to the output queue of the local system for synchronization. The remote system then accesses the output queue and moves the deletion transaction to the batch input queue on the remote system:

- The relational database is accessed using the document's external document number to retrieve the remote system's header for that document.
- The text and image files are renamed using the document's internal ID.
- The header is flagged for deletion and stored in the remote system's relational database.

2.4 SUMMARY

The TDOCS database synchronization design is very straightforward while retaining maximum functionality and flexibility. The primary element of the design is maintenance of output queues at each site for documents to be shared. As the server at each site inserts documents to be shared into its output queue, the server at the other site copies those documents into its own input queue. This approach utilizes the existing batch processing functionality to accomplish synchronization in a uniform manner because synchronization transactions are processed exactly as any other transaction appearing in the batch processing input queue.

Table 2-2. Required processing for synchronization of document update transactions

Prior Sharing Status	Updated Sharing Status	Actions Required In The Local Database	Actions Required In The Remote Database
Shared	Unshared	Update the transaction in the local database Generate a document deletion transaction and store it in the local output queue	Copy the deletion transaction from the output queue of the local system to the input queue of the remote system and delete the document in the remote database through the batch process
Unshared	Shared	Update the transaction in the local database Generate a document submission transaction and store it in the local output queue	Copy the submission transaction from the output queue of the local system to the input queue of the remote system and submit the document to the remote database through the batch process
Shared	Shared	Update the transaction in the local database Copy the document update transaction to the local output queue	Copy the update transaction from the output queue of the local system to the input queue of the remote system and update the document in the remote database through the batch process
Unshared	Unshared	Update the document in the local database	No action required in the remote database

Table 2-3. Required processing for synchronization of document deletion transactions

Prior Sharing Status	Actions Required In The Local Database	Actions Required In The Remote Database
Shared	Delete the document from the local database Copy the document deletion transaction to the local output queue	Copy the document deletion transaction from the output queue of the local system to the input queue of the remote system and delete the document from the remote database through the batch process
Unshared	Delete the transaction from the local database	No action required in the remote database

17/18

3 IMPLEMENTATION PLAN

This chapter provides an implementation plan for the synchronization of the TDOCS databases at the DWM and CNWRA.

Functionality described in Chapter 2 will be coded in the C programming language for both the client and server platforms to support submission, updates, and deletions of documents in the TDOCS repositories. Submission, update, and deletion transactions are accumulated for batch processing. Documents to be shared are copied to the output queue by the server to permit the remote system to access them for synchronization.

Documents stored in the output queue of the remote system are copied to the input queue of the local system during the day prior to the initiation of the batch processing. Because the synchronization design relies on manipulation of the output and input queues, the batch process makes no distinction between synchronizing and locally submitted transactions. Therefore, the existing batch processes are utilized for both local and synchronization processes.

The design described in this report has been successfully prototyped under Sun OS. The actual synchronization modules will be coded to operate under the Solaris operating system.

The synchronization code will be installed under the Solaris operating system when both the DWM and the CNWRA switch to that environment following NRC acquisition of database and text search software and installation of the TDO software at the NRC. This installation will require a final system test of the synchronization processes before putting them into production.

18/18

4 REFERENCES

Johnson, R.D., J.H. Cooper, C. Moehle, and E. Harloe. 1993. *Technical Reference Document Database System (TDOCS) Requirements Definition.* San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

DeWispelare, A.R., J.H. Cooper, and R.L. Marshall. 1995a. *Proposed Technical Reference Document Database System (TDOCS) Design.* San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

DeWispelare, A.R., J.H. Cooper, and R.L. Marshall. 1995b. *Initial Data Loading Plan.* San Antonio, TX: Center for Nuclear Waste Regulatory Analyses. To be published.