Spring 2012

Development of Database and Software Modules to Identify Case Relationships Using Unstructured Data

Alberto D. Lombardo
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

DEVELOPMENT OF DATABASE AND SOFTWARE MODULES TO IDENTIFY CASE RELATIONSHIPS USING UNSTRUCTURED DATA

A THESIS

SUBMITTED ON 30th OF JANUARY, 2012

TO THE DEPARTMENT OF INFORMATION SYSTEMS

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF

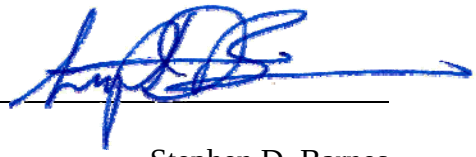
SCIENCE IN DATABASE TECHNOLOGIES

BY

Alberto D. Lombardo

APPROVALS

Robert T. Mason, Thesis Advisor



Stephen D. Barnes

Shari Plantz-Masters

Abstract

In this study, I intend to show how text based unstructured data, such as word processor documents and e-mails, can be systematically parsed for instances of data classes. Data classes can be any data type that can be fully described using the defined syntax of regular expressions. Examples of data classes can include SHA1, or MD5 hash values, Internet Protocol (IP) addresses, or any other data type whose format is well defined. Furthermore, possible correlations between datasets may be identified by grouping instances of equal or similar values within a data class. This approach of utilizing regular expressions to define the search criteria negates the need to predetermine what keyword to search for.

Acknowledgements

I offer my sincerest gratitude to my supervisor, Mr. Sigurd Murphy, and the government lead for the section I currently support, Special Agent Kevin Rivera, who have supported me with their patience, allowing me the ability work on this project. I would also like to acknowledge Mr. Michael Burtowski, the General Dynamics Advanced Information Systems contract lead, who saw fit to approve the funding of my masters program entirely through the company's employee higher education program. The completion of this degree program would not have been possible without their belief in me, nor without their monetary support. I would like to thank my two children, Jacob and Peter, who have come to expect their father to come home and start working on his school work. They have both learned to adjust to the limited time they have available with their father, but the success to how they have matured, goes entirely to my wife, Sasha. Without her support, both in encouragement and by taking care of the kids, this project would not have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

A database system can provide an organization with the means to manage large amounts of data on customers, suppliers and other pertinent data. This is due to a relational database system's ability to handle a tremendous amount of data with relative ease. The structured query language (SQL) is also very efficient in extracting information based on patterns, providing both detail data and *big picture* information to those querying. This approach works well for data that has already been structured to conform to the underlying database schema, more problematic is when the raw data is unstructured with little possibility of transforming it into something more meaningful. According to a recent survey cited by McKendrick in Unstructured data: the elephant in the Big Data room "unstructured data is growing at a faster clip than relational data." (McKendrick, 2011) Worse still, this survey reveals that "– 91% say unstructured information already lives in their organizations, but many aren't sure what to do about it" (McKendrick, 2011).

Unstructured data can take on many forms, including audio/video, graphics, e-mails, text documents, and social media messages among others (McKendrick, 2011). Great strides have already been made on search techniques for text based unstructured data. Microsoft's Windows operating system now offer indexing of the contents of the information contained on a system's hard drives. Google's Search Appliance can also parse the contents of files located anywhere on a corporate intranet. These search capabilities however, have one significant drawback; they all require the user to know exactly what they want to search for.

There is no systematic approach to search through text based unstructured data to extract classes of data, and then group cases with similarities between extracted instances of a data class to establish possible links between the different datasets.

For certain types of organizations with large volumes of text based unstructured data; it would be valuable to provide a means to search through their volumes of unstructured data to find information whose format matches a class of data rather than a specific search term. In this context, a data class may be e-mail addresses, phone numbers, IP addresses, or zip codes or other data class types that can be fully described using an established set of rules. Furthermore, once instances of the data class have been extracted, matching values between datasets may reveal previously undiscovered connections between those datasets. A dataset, in context for this research study, are the collection of unstructured text based data all related to an individual case. Rather than providing, for example, an individual e-mail address to search for, the data class search extracts all occurrences of everything whose format looks like an e-mail address contained within the unstructured data. Cases where the same values of e-mail address are extracted may in fact be connected. Furthermore, cases that share multiple data items may have a higher likelihood of being connected to each other. Additionally, two, or more, cases with no common shared information between them, but rather show common traits with a third case may also be connected, creating a web of connections not realized solely from the original source data.

As an investigative tool, many organizations could take advantage of this type of search capability to *connect the dots* between different datasets. Examples with a need for just such a capability include law enforcement agencies, insurance companies and organizations with a counter intelligence/counter terrorism mandate.

Local police precincts regularly process a large volume of text based unstructured data. These agencies often take witness statements, incident reports, confessions, etc., that contain specifics to individual cases. Currently, to find possible connections to other suspects or cases, law enforcement officers often utilize a system called CopLink. CopLink allows officers to

perform text searches such as “for a suspect known only by his first name” (Chen, Zeng, Atabakhsh, Wyzga, Schroeder, 2003). By entering a name, the search returns other suspects with similar names. This approach requires the officer to know what he is looking for. By culling class data from their reports, what may seem like unrelated cases, may turn out to have several similarities.

Insurance companies also often process text based unstructured data as well, in the form of claim reports, claimant statements, witness statements, etc. To minimize the possibility of fraud, they regularly investigate the validity of a claimant’s statement. “Increasingly, the industry is seeking technological solutions, via two main approaches. The first, data mining, utilizes artificial intelligence and database techniques to discover patterns and inconsistencies in client records and claims histories” (Ormerod, Morley, Ball, Langley, Spencer, 2003). Common traits identified by data class searches with other claims may help identify evidence of possible fraud.

Counter intelligence/counter terrorism agencies of the federal government must process large amounts of text based unstructured data. Counter Intelligence (CI) agencies often provide “Forensic examinations that involve cyber crimes and fraud” (Laboratory Services) as well as “Forensic examinations on cases that involve security violations, espionage, classified information, and support for the war on terrorism” (Laboratory Services). CI agencies can utilize this search approach to identify links between cases that may seem unrelated otherwise.

Chapter 2 – Review of Literature and Research

Parsing unstructured text based data to identify possible connections between documents is not a new endeavor. The literature on searching unstructured text presents various algorithms and techniques to perform searches and then identify methods of grouping like documents together.

Search Techniques and Data Correlations

The ability to *connect the dots* between ideas contained in different documents has shown to be a very valuable concept. One such method to process unstructured text involves the use of Natural Language Processing (Fan, Wallace, Rich, Zhang, 2006) which has led to the creation of technologies that allow computers to process natural language (Fan, Wallace, Rich, Zhang, 2006). In Tapping the Power of Text Mining, the authors discuss the case where a University of Chicago professor, while researching articles on migraines, identified the keywords that appeared at a significant frequency (Fan, Wallace, Rich, Zhang, 2006). One such significant keyword he discovered was *spreading depression* (Fan, Wallace, Rich, Zhang, 2006). While following the same process of culling the significant keywords for documents that contained spreading depression, he found *magnesium deficiency* (Fan, Wallace, Rich, Zhang, 2006) as a keyword. The professor theorized that there may be a connection between migraines and magnesium deficiencies. "The hypothesis was made only by linking related documents from migraines to those covering spreading depression to those covering magnesium deficiency. The direct link between magnesium deficiency and migraine headaches was later proved valid through scientific experiments" (Fan, Wallace, Rich, Zhang, 2006). In this case, the searches by the professor were performed by hand but simulated the same concept-linkage technology text-mining products provide. (Fan, Wallace, Rich, Zhang, 2006) This case is significant, in that it demonstrates the

ability to tie concepts contained within unrelated documents to give insight into new intelligence not previously known without the correlation. Another method of linking documents is through *clustering* (Fan, Wallace, Rich, Zhang, 2006). Through clustering, similar documents are clustered on the fly by creating "a vector of topics for each document and measures the weights of how the document fits into each cluster" (Fan, Wallace, Rich, Zhang, 2006). Given a keyword, such as "Saturn," clustering engines may create topic vectors such as "planet, photo, car, and performance" (Fan, Wallace, Rich, Zhang, 2006), with users narrowing the choices for which types of documents to present to the user. International Business Machines' (IBM) Text Navigator application uses clustering methods to group document together according to their contents and annotates each document with the two most frequently used words in the group of document clustered together (Feldman, 2006). To further enhance clustering results, parsing documents for terms will also increase the accuracy of how each document is clustered. Term extraction will need to be processed by an actual human reader that creates tokens of parts of speech. Examples may include terms such as "[wall] and [street]" (Feldman, 2006,). This process increases the need for human interaction to process each document to extract terms; however it reduces the number of insignificant results (Feldman, 2006).

Another novel approach to processing unstructured data is discussed by the authors of Efficiently Linking Text Documents with Structured Information. Their approach attempts to link the contents of unstructured data with the instances of data already within a structured database. As an example, they cite an organization with a network of multiple stores but have an inflow of complaints into a centralized location. Complaints can come from various sources, such as web-form, email, fax and voice-mail that are "typically a free-flow narrative text about one or more sales transactions" (Chakaravarthy, Gupta, Roy, Mohania, 2006). The actual complaint may

not contain specific data to identify a particular transaction, but may reveal information such as the store name, partial list of items bought, or the purchase dates that can assist in creating a list of possible transactions that can be acted upon to address the complaint. To accomplish this task, documents are viewed as a "sequence of sentences where a sentence is a bag of terms" (Chakaravarthy, Gupta, Roy, Mohania, 2006). Some of the terms are potentially useful since they also come up in the database as well, such as a store location. A part-of-speech parser is used to identify the nouns in sentences contained within the complain, and filters out the rest with the assumption being made that the nouns are more likely to appear as values in the database (Chakaravarthy, Gupta, Roy, Mohania, 2006). To optimize the process of linking terms to a possible entity, the approach utilizes a context cache that can be viewed as a set of relationships of the form *(e, t)* meaning that the term *t* is contained in the context of the entity *e* (Chakaravarthy, Gupta, Roy, Mohania, 2006). The two operations that actually accesses the database are "*GetEntities(t)*" and "*GetTerms(e)*" (Chakaravarthy, Gupta, Roy, Mohania, 2006). GetEntities(t) is defined as "given a term t appearing in D, this operation queries the database and returns the set of all entities that contain the term t in their context." GetTerms(e) is defined as "given an entity e, this operation queries the database and returns the set of all terms from D that are contained in the context of e" (Chakaravarthy, Gupta, Roy, Mohania, 2006). As a final entry on combining structured data with unstructured, the authors of Text-to-Query: Dynamically Building Structured Analytics to Illustrate Textual Content propose a framework for presenting a query to a database by simply using natural language statements. They propose a scenario in which a travel agency, which manages reservations for multiple resorts around the world that would like to prepare specific graphs. By typing "Sales analysis by region" (Thollot, Brauer, Barczynski, Aufaure, 2010) several appropriate sales chart are generated and made available to be selected. If the user

further adds a specific resort as a detail, additional charts showing the amount of sales in the specific resort are added. To accomplish this, the framework relies on three components which are described as "metadata stored in a data warehouse, a state-of-the-art entity extraction framework and pre-defined analysis categories" (Thollot, Brauer, Barczynski, Aufaure, 2010). The data warehouse is designed to hold "multi-dimensional OLAP cubes that can be defined over a structured database" (Thollot, Brauer, Barczynski, Aufaure, 2010). The cubes are defined to hold numerical facts and can be aggregated against different dimensions. Furthermore, a user can drill down, or filter on some specific value to restrict the scope of analysis. The entity extraction is defined over "57 predefined entity types such as person, organization or geographic location" (Thollot, Brauer, Barczynski, Aufaure, 2010). Additionally, this list can be further supplemented with extra dictionaries, "capturing canonical and alternative names to recognize custom entities" (Thollot, Brauer, Barczynski, Aufaure, 2010). Categories are generated from the data warehouse and each category is associated with the terminology that related to it. To actually extract the data from the warehouse database, a "large dictionary that contains reference entities to be extracted in text documents" (Thollot, Brauer, Barczynski, Aufaure, 2010) are automatically generated.

Chapter 3 – Methodology

As stated purpose of this research project is to develop a method to systematically search case-related, text based unstructured data to extract instances of data classes, and then identify possible associations among those instances based on similarities within the extracted data classes. To accomplish this goal, a series of software tools and applications were designed and developed. To act as the data repository, a database system to house basic case related data in conjunction with the documents that pose as the unstructured text was developed. Additionally, the database system is also responsible for managing the case indicators, which are the instances of the data classes extracted from the documents. A database schema was modeled to manage these data items, and was instantiated on a MySQL database server; the database creation script is included in Appendix A. The database was then populated with basic case data and representations of documents posing as the unstructured text based data associated with specific cases. With the database instance in place, a Windows based case viewer application used to insert and manipulate the data held within the database was created. The case viewer tool would serve as an organization's main portal to manipulate and view case related data, including the documents. To perform the regular expression searches, an additional Windows based application was developed that would systematically parse each document contained within the database against a series of regular expressions designed to extract the case indicators. In conjunction with the application, the actual regular expressions were developed and tested to ensure they extracted the data as expected. Additionally, since it would be reasonable to expect an organization to desire the scanning of their documents to be automated as they are added and modified within the repository a Windows based service application was developed. The service application runs in the background on an application server computer with minimal user

interaction. This is ideal for large organizations with constant additions and updates to their document set. The method each of these tools utilizes is detailed in the following section, along with an explanation of how each tool contributes toward the satisfying of this research project.

Database Schema

The design of the database schema was made to handle several data areas pertinent to this research project. The primary data area housed within the database is the data related to each case along with the documents associated with a case. As stated above, presumable, an organization would already have this portion of their database already established. The secondary area of the database schema is dedicated to providing support to hold the regular expressions that define the data classes and the case indicators extracted from the documents.

This schema subdivides all cases by year; this was desirable for this research project to simplify the display of case data within the case viewer application that was developed. An organization may decide not to follow this model if their data does not lend itself to being subdivided by year. Additionally, no additional data of importance is being maintained at year table level; however, an organization may choose to keep year specific data if they so choose. The case_file table is responsible for capturing essential facts about a particular case. For this research project, the data maintained at the case_file level is also limited due to it not being relevant toward this project. An organization would presumably keep more specific case data within or as a child entity off of this table. The case_document table holds the actual instances of the unstructured text relevant for this research project. An organization may choose to merely hold the file system path to the actual document, however; considering how volatile a file system directory structure can be, it was decided that to satisfy the requirements for this research project, the actual documents would be housed within the database table itself within a binary large object

field. The document_type table provides the means for the organization to classify each document within a subtype. The document types can be expanded as needed by an organization to accommodate new types as they are needed.

Figure 1 – Data Model of Basic Case Data Subject Area

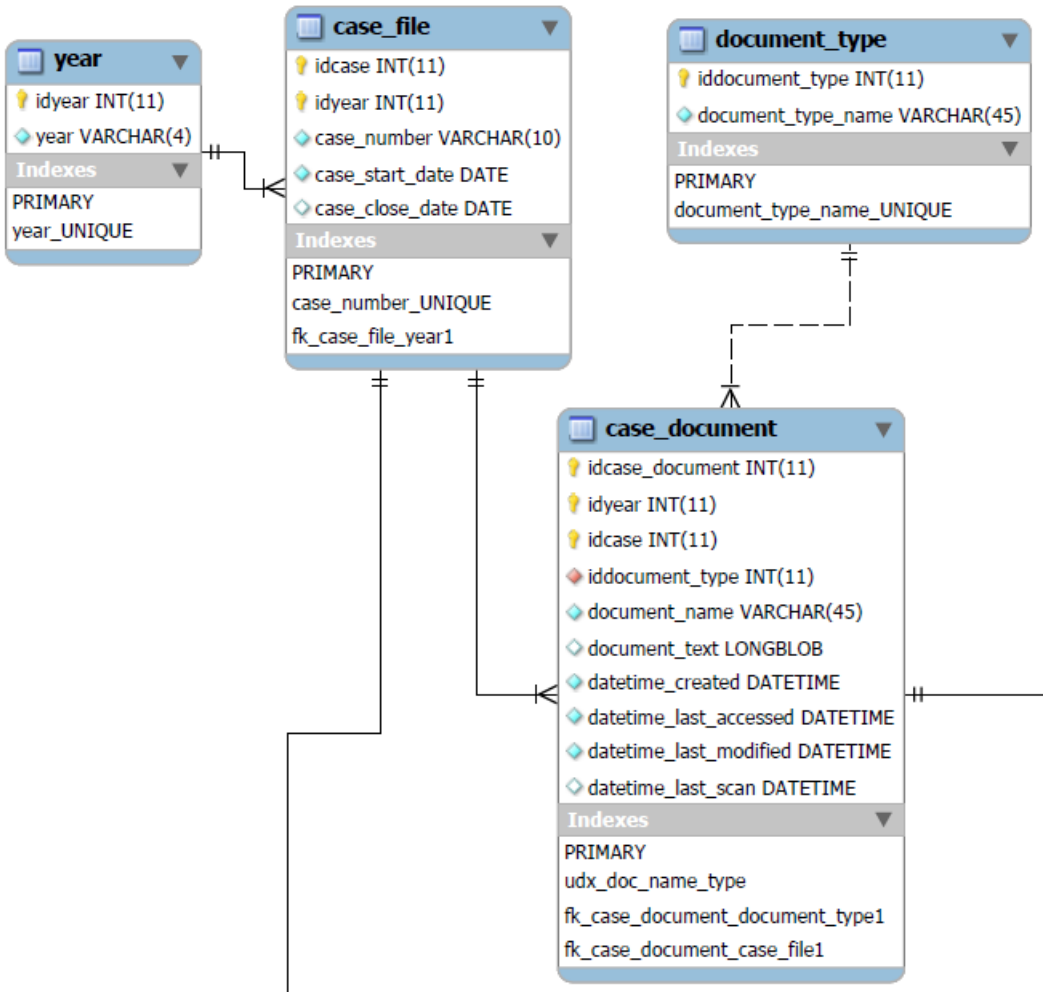


Figure 1. Main data areas associated with a typical organization maintaining case data with instances of the unstructured text based documents.

The secondary data area is responsible for maintaining the regular expressions and the case indicators extracted from the database.

The regex_pattern table is designed to hold the regular expressions, along with the expression options necessary to perform the searches. The post_process table is used to hold

structured query language statements that will be used, in essence, to filter invalid instances of the extracted indicators from the results of the regular expression parsing process. This process will be further detailed in the description of the search method. The searched_document and matched_text tables hold all the instances of case indicators extracted during the search process

Figure 2 – Data Model of Regular Expression/Case Indicator Subject Area

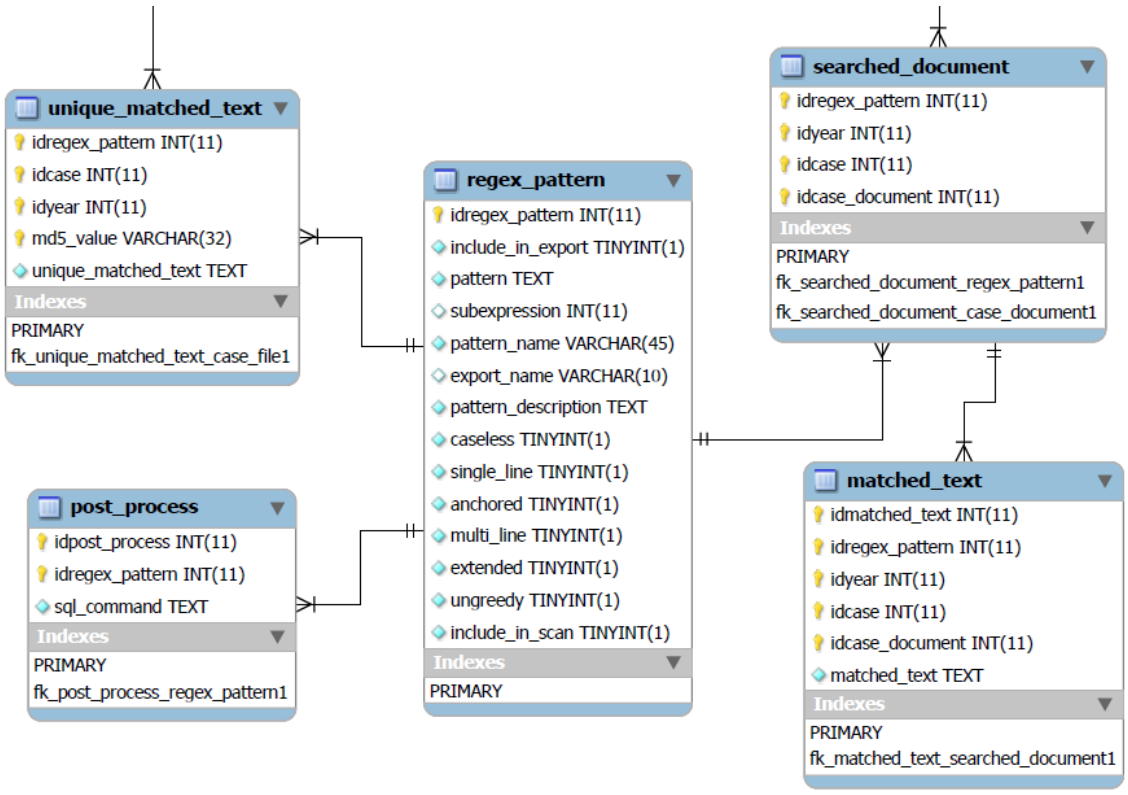


Figure 2. Main focus of this research project is captured by these tables. The regex_pattern fully describes a regular expression and the unique_matched_text table will ultimately hold the instances of case indicators (class data extracted from the case documentation).

and relate them to the document they were found in. An important aspect to realize is that a search may reproduce the same indicator for a given document and/or case. During the searching phase, reproduced indicators are maintained within the matched_text table but need to be condensed down to just one instance later. Optimally, these two tables, in a production environment, would be better suited as temporary tables only accessible to the current user,

however were included in the schema merely to highlight their function within the search process. As a centralized table, visible to all users could cause problems if more than one user were to conduct regular expression parsing at the same time. The unique_matched_text table is responsible for managing the condensed list of indicators and relates them to their associated case. An additional point to observe is that during the search phase, all indicators held within the matched_text table are associated to the document they were extracted from, while the condensed list of indicators held within the unique_matched_text table are associated directly to the case they are related to.

The following tables describe the database schema. Table one describes each database table, its overall purpose, and then describes the columns of each database table. Table two identifies the indexes created describing the business rules the index enforces within the schema.

Table 1

Database Schema, Table Description

Table Name (Note)	Table Purpose	Column Names	Column Purpose
year (For the purposes of this research project, no data is actually held within this table. Any such information an organization would care to hold within this table would not play a significant role for this project)	Allows the display of case data subdivided by year	idyear	Primary key column for table (auto increment field)
		year	Holds the year that cases will be related to

Table Name (Note)	Table Purpose	Column Names	Column Purpose
case_file (For the purposes of this research project, only basic case data are held. The actual data held in this table does not play a significant role for this project, however; in an organization, this information would be more detailed)	Holds basic data related to a case for an organization	idcase	Primary key column for table (auto increment field)
		idyear	Primary key column for table (foreign key from year table)
		case_number	unique number to identify a case
		case_start_date	date the case was started
		case_close_date	date the case was closed
document_type	Provides a list of possible categories that a document can be classified into.	iddocument_type	Primary key column for table (auto increment field)
		document_type_name	The category name that a document can be classified into
case_document	Holds the unstructured text (documents) that are associated with a particular case.	idcase_document	Primary key column for table (auto increment field)
		idyear	Primary key column for table (foreign key from case_file table)
		idcase	Primary key column for table (foreign key from case_file

Table Name (Note)	Table Purpose	Column Names	Column Purpose
			table)
		iddocument_type	Primary key column for table (foreign key from document_type table)
		document_name	The name of the document, the document type and document name must be unique for a particular case.
		document_text	The unstructured text of the document. The document text is not limited to solely text. It may also contain graphics or other embedded objects, however; only the text will be searched for instances of data classes.
		datetime_created	date time stamp of when the document was first inserted within the table.
		datetime_last_accessed	date time stamp of when the document was last accessed by a user. Accessing the document for the purposes of searching, is not registered.
		datetime_last_modified	date time stamp of when the document was last modified.
		datetime_last_scan	date time stamp of when the document was last scanned by the search process.
regex_pattern	Is the list of regular expressions that define a data class, and is used by the search process to extract the case indicators	idregex_pattern	Primary key column for table (auto increment field)
		include_in_export	Flag to indicate if the case indicators found to match a particular regular expression should be included in the XML export report.

Table Name (Note)	Table Purpose	Column Names	Column Purpose
		pattern	The regular expression that defines a data class.
		subexpression	For regular expressions that use a particular token to find a data class, it may not be desirable to actually return the token as part of the data found. Subexpressions allow the regular expression to be broken into subgroups (within parenthesizes). If subexpression has a value other than zero (0), then only the nth group is returned as the extracted data class.
		pattern_name	A short name used to identify the regular expression,
		export_name	The name used for a particular pattern within the XML export report.
		pattern_description	A full description of the regular expression.
		include_in_scan	Flag to indicate if the regular expression pattern should be used in the search.
		caseless	Regular expression flag option – performs the search where comparisons disregard character case.
		single_line	Regular expression flag option – Allows lines that span more than one line within the text to be treated as a single line.
		anchored	Regular expression flag option – Regulates if the next match starts right after the first character the previous match, or at the end of the previous match.
		multi_line	Regular expression flag option – Treats one string with multiple lines as multiple strings.

Table Name (Note)	Table Purpose	Column Names	Column Purpose
		extended	Regular expression flag option – Allows the regular expression pattern to contain extra white space, newlines, and perl-style comments.
		ungreedy	Regular expression flag option – Used to match as many characters as possible
searched_document	Holds a list of all documents scanned by a particular regular expression pattern during the current search	idregex_pattern	Primary key column for table (foreign key from regex_pattern table)
		idyear	Primary key column for table (foreign key from case_document table)
		idcase	Primary key column for table (foreign key from case_document table)
		idcase_document	Primary key column for table (foreign key from case_document table)
post_process (Table holds actual update and/or delete SQL statements that are used to filter out invalid indicators. For example, for the IPv4 data class, IP ranges such in 10.x.x.x or 192.168.x.x are of no particular intelligence value since they indicate internal network IP	Provides a means to filter out invalid instances of class data found within the documents	idpost_process	Primary key column for table (auto increment field)

Table Name (Note)	Table Purpose	Column Names	Column Purpose
ranges. These IP addresses can be easily deleted)			
		idregex_pattern	Primary key column for table (foreign key from regex_pattern table)
		sql_command	update or delete SQL statement that will be used to cleanse the indicator data.
matched_text (Instances of the matched text are related to the document they were extracted from. This table may hold many instances of the same matched text for each document and case.)	Holds all the instances of case indicators extracted from the case documents.	idmatched_text	Primary key column for table (auto increment field)
		idregex_pattern	Primary key column for table (foreign key from searched_document table)
		idyear	Primary key column for table (foreign key from searched_document table)
		idcase	Primary key column for table (foreign key from searched_document table)
		idcase_document	Primary key column for table (foreign key from searched_document table)
		matched_text	Holds the class data instance found and associated to the searched document.
unique_matched_text (Instances within this table have been cleansed by the filtering process and	Instances of unique indicators extracted from the documents.	idregex_pattern	Primary key column for table (foreign key from regex_pattern table)

Table Name (Note)	Table Purpose	Column Names	Column Purpose
condensed to just one instance of the matched text; however it is related to the case rather than the document.)			
		idcase	Primary key column for table (foreign key from case_file table)
		idyear	Primary key column for table (foreign key from case_file table)
		md5_value	Primary key column for table To assist in condensing multiple instances of matched text to just one instance, the matched text should also be part of the primary key, however; MySQL disallows a binary large object (BLOB) to be part of the primary key. To circumvent this limitation, the MD5 hash value of the matched text is calculated and saved. The composite key is now possible using the MD5 hash value in lieu of the actual matched text.
		unique_matched_text	The instance of the case indicator extracted from the documents and associated with its case.

Table 2

Database Schema, Table Indexes

Table Name	Index Name (Index Type)	Column(s)	Index Purpose
year	year_UNIQUE (Unique Index)	year	Ensures year value is not repeated.
case_file	case_number_UNIQUE (Unique Index)	case_number	Ensures case number value is not repeated.
case_document	udx_doc_name_type (Unique Index)	idcase iddocument_type document_name	Ensures unique document name/document type within a case
document_type	document_type_name_UNIQUE (Unique Index)	document_type_name	Ensures document type value is not repeated

Case Viewer Application

The case viewer is developed as a Windows application that accesses the MySQL database and provides the capability to create, delete, view and manipulate basic case related data and the associated case documents. Additionally, the case viewer also provides the capability to view case indicators that have been extracted from the case documentation. The viewer application organizes case files by year in descending order, and lists all the available cases beneath the year nodes. Furthermore, case documents are organized as nodes beneath the case. The source code for the case viewer application can be found in Appendix C.

To expand or collapse any node, a user may double click on the text of a node or simply click on the diamond prefacing the desired node. By clicking the right mouse button over a node, a context sensitive menu pops up showing the possible options allowed for that particular node.

By selecting the root node, a user can add a new year. Selecting a year node allows the addition of a new case, and by selecting the case node, a user can either view case related data or add a new document. Finally, by selecting a document node allows the deletion or editing of the selected document. Figure 3 shows the basic structure of the application's graphical user interface described.

Figure 3 – Case Viewer Application (Main Graphical User Interface)

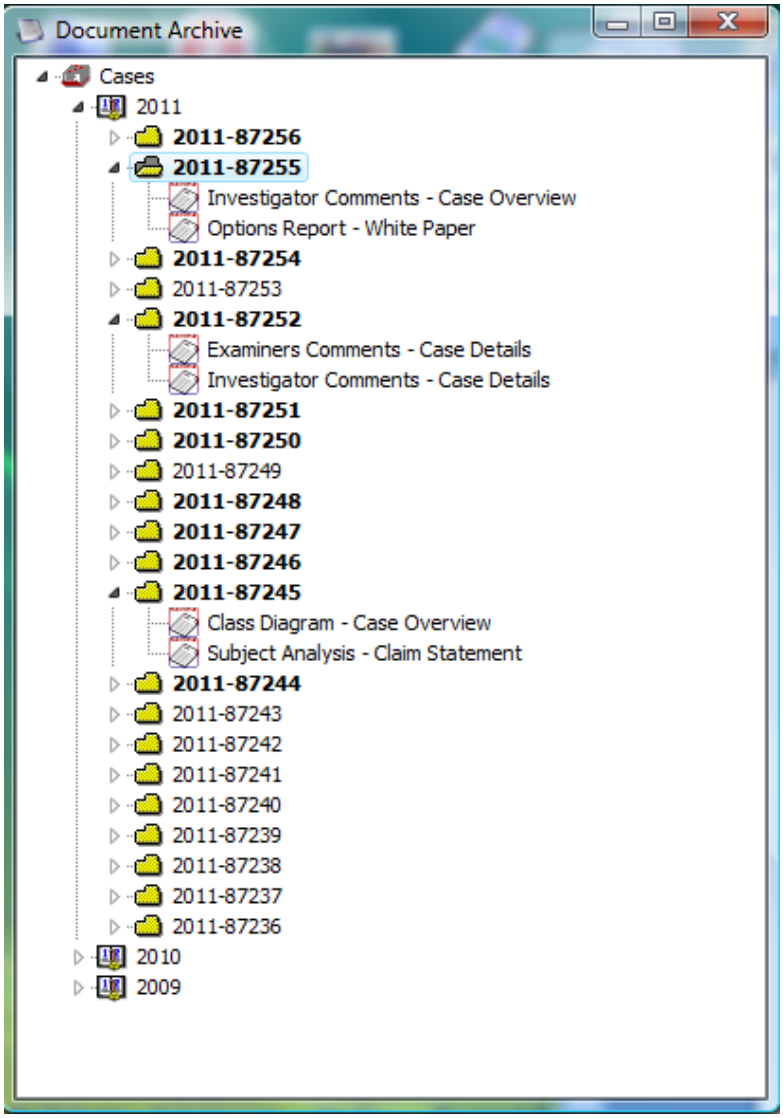


Figure 3. Main graphical user interface for the case viewer application. Each node is populated directly from the database tables. Users can use the right mouse button to cause a context sensitive menu to pop up providing options on how to view and/or manipulate the data.

Case Maintenance

To add a new case, a user need only right click on the appropriate year node, and select *New Case*. Once selected, the new case dialog box pops up already populated with the next sequential case number and the current date as the case start date. Upon acceptance, the new case is added to the list of cases under the selected year.

Figure 4 – Adding a New Case

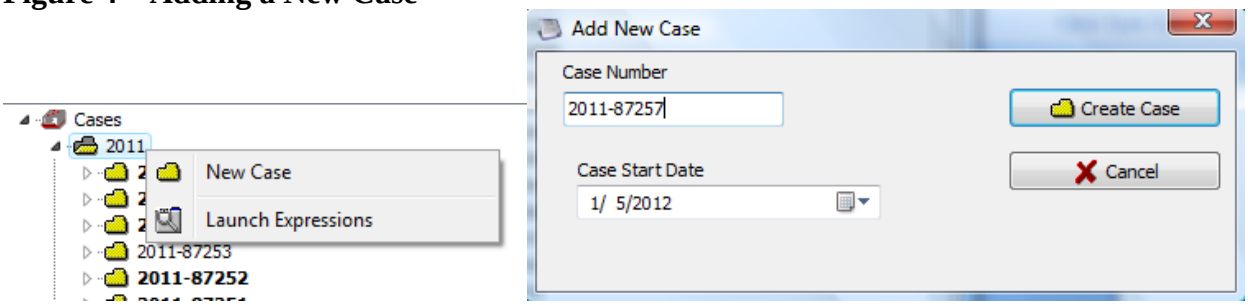


Figure 4. Cases are added as needed by an organization; each case can have multiple documents associated with it.

Document Maintenance

Documents can be imported, or created within the case viewer application. The application has been designed to allow documents to be imported if they are formatted as plain text (.txt), rich text format (.rtf), Microsoft Works documents (.wps), Microsoft Word documents (.doc or .docx) or finally, WordPerfect documents (.wpd). This allows an organization the freedom to use the tool of their choice to generate their documentation, and still maintain compatibility with the case viewer. To import a new document, simply right click on the appropriate case, and select *New Document* from the popup menu. From the import document dialog, enter a *document name*, and select a *document type*. If *Import Now* is checked, clicking the *Create Document* button allows a user to select an import file from the file system. If *Import Now* is not checked, a blank document is created and opened, ready to be edited by the user. The word processor control features many of the same advanced features available in many

professional word processors, such as Microsoft Word, however, the individual capabilities are not detailed here since it has little impact on the capabilities associated with this research project.

Figure 5 – Adding a New Document

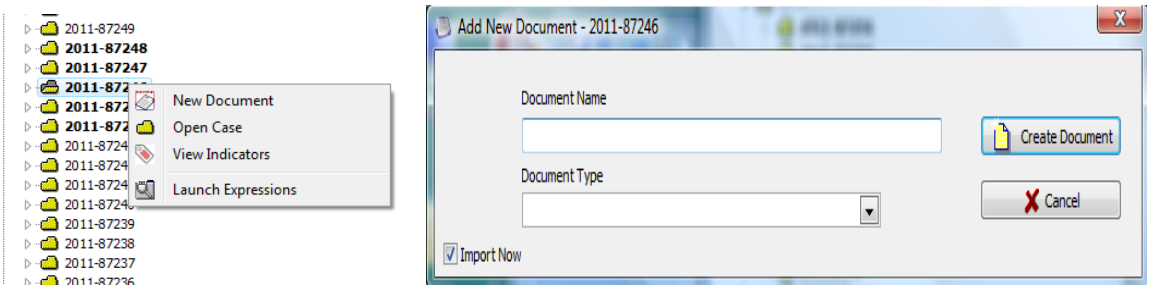


Figure 5. As an organization gains new documentation for a given case, it should be added to the database to allow access organization wide, and to allow it to be scanned by the parser.

To open a document for viewing or editing, right click on the desired document node and select *Open Document*. The application now extracts the document from the database, and loads it into an embedded word processor control on a secondary form. Changes to the document are also written directly to the database from the application.

Figure 6 – Viewing/Editing a Document

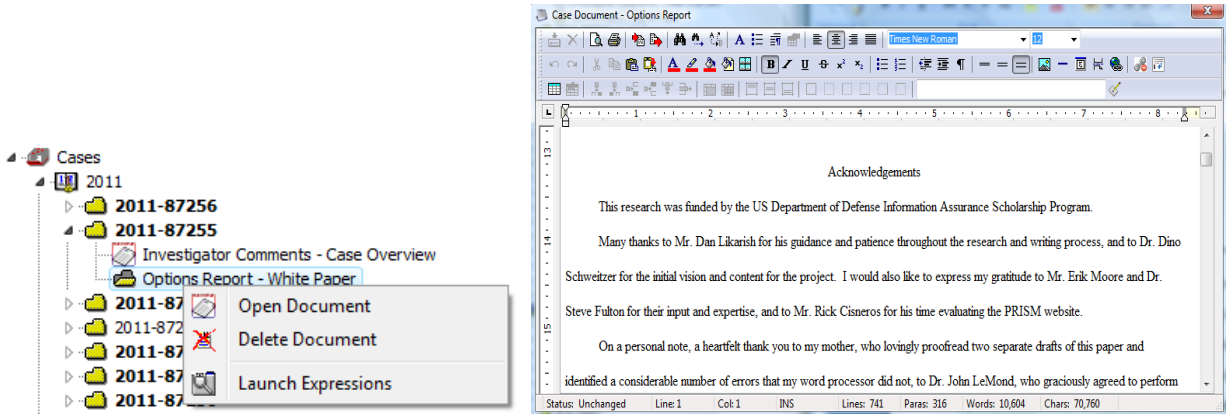


Figure 6. The application provides a full featured word processor which is important for an organization that needs to document more than is possible with a text editor like Notepad.

The database has been designed to track when a particular document is first added to the database, and then any subsequent changes made to it, to include being scanned for case

indicators. In concert with the database, the case viewer application ensures these data items are maintained accurately. When a document is created, either directly within the application or via document import, it automatically enters the current date and time into the datetime_created, datetime_last_accessed and datetime_last_modified fields of the case_document table. When a document is accessed via the case viewer application, the datetime_last_accessed is again updated with the current date and time. If modifications are made and saved to the document, the datetime_last_modified is then updated with the current date and time. These date time stamps will be used by the Windows service application, discussed later, to determine which documents need to be scanned. Once scanned, either by the regular expression parser application, or the Windows Service application, the datetime_last_scan field is updated with the current date and time of the scan.

Figure 7 – Date Time Status of Document Import

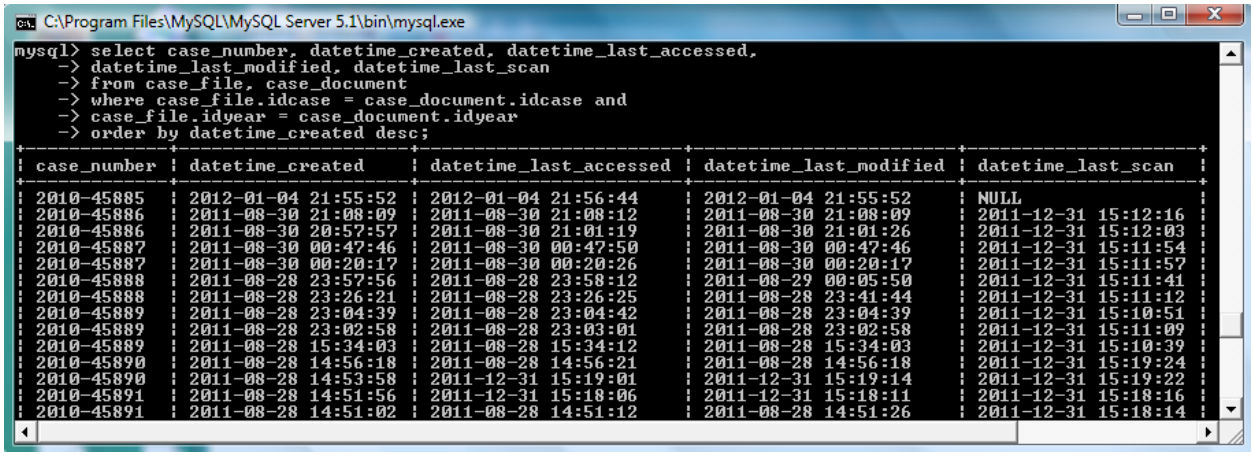


Figure 7. For imported document associated with case 2010-45885, the date time stamps for datetime_created, datetime_last_accessed and datetime_last_modified are shown to reflect the date and time the document was imported. Additionally, the document was subsequently accessed moments later, shown with an updated datetime_last_accessed. The datetime_last_scan value remains NULL until the document is actually scanned.

Viewing Case Indicators

Once the series of documents associated with a case has been scanned by the regular expression parser or regular expression service application to extract case indicators, the unique case indicators are populated into the unique_matched_text table, and available to be viewed within the case viewer application. There are two distinct case indicator views available to users of the application. Case indicators can be viewed either from the perspective of an individual case or all case indicators regardless of case.

Case related indicators can be brought up by right clicking the desired case and selecting *View Indicators*.

Figure 8 – Case Specific Indicator View

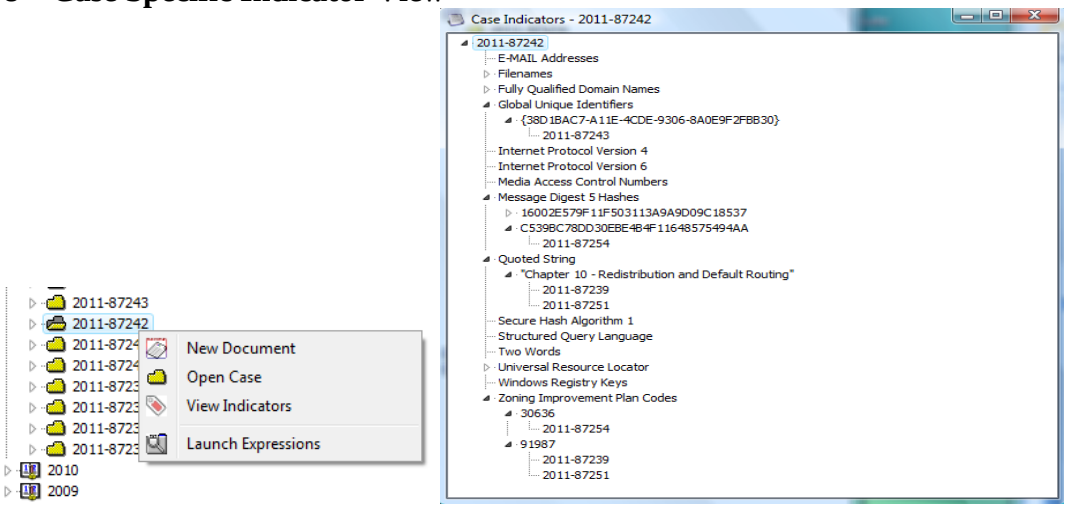
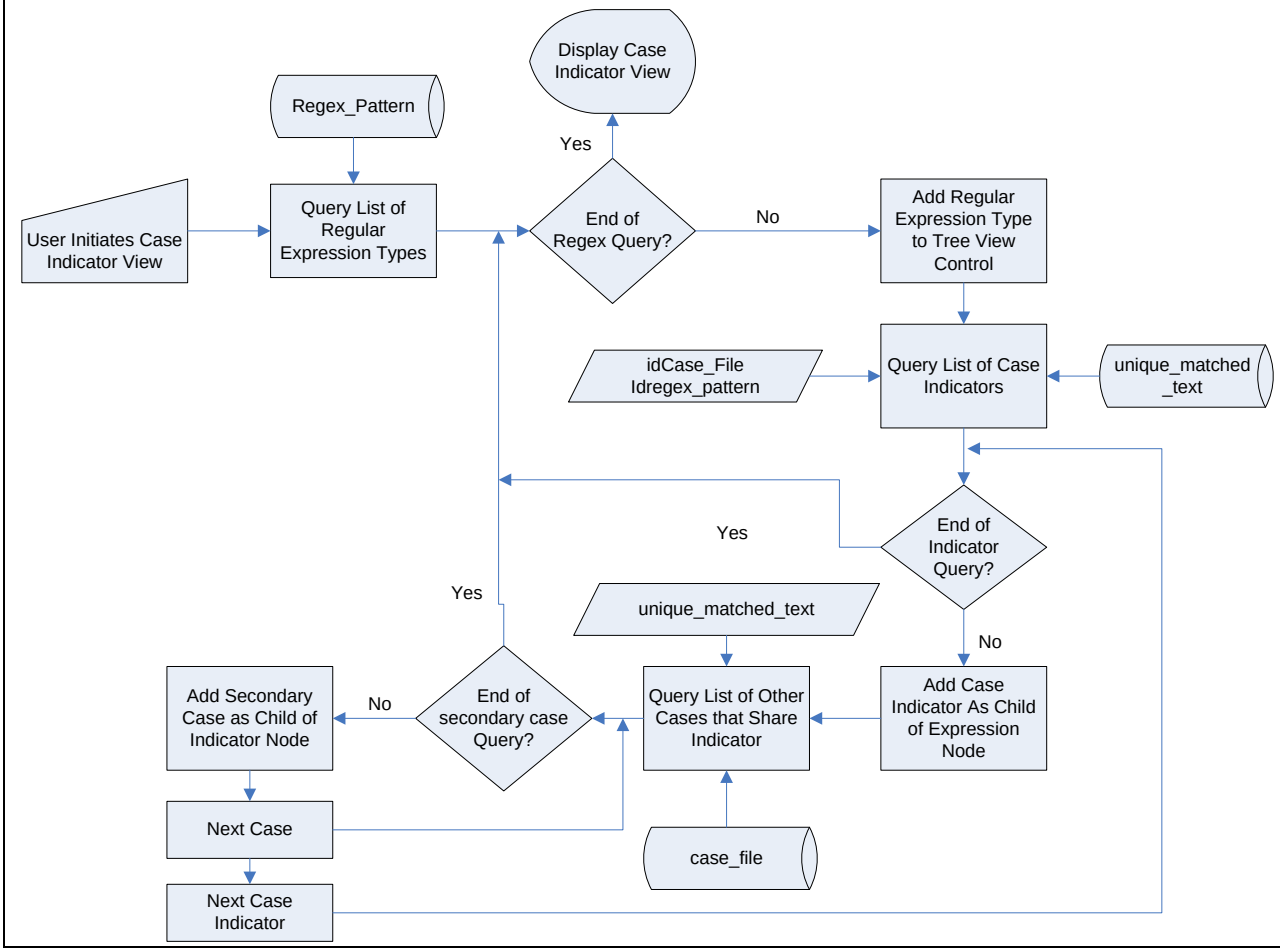


Figure 8. The Case Indicator view allows a user to view all the indicators, organized by type, for a selected case. If there are other cases that share that same indicator, it is shown as child nodes to the indicator. Clicking on the case opens a new case indicator view for the related case in a new window.

To populate the *Case Indicator* display, three distinct queries are necessary. The initial query result set is the list of pattern types from the regex_pattern table. Iterating through the list of patterns, the database is then queried for all case indicators associated with the case node selected in the case viewer and the pattern. A final step queries each individual case indicator to

find other cases that may share this piece of data. By clicking on the associated case node opens the same case indicator view for the selected case in a new window, allowing direct comparison to other case indicators. Figure 9 details the process used to create the Case Specific Indicator view.

Figure 9 – Case Specific Indicator View Process



To view case indicators from the perspective of all available indicators regardless of case association a user would right click on the root node and select *View All Indicators*.

Populating the *All Indicator* display presents some difficulties due to the sheer number of case indicators to manage. If the entire hierarchical tree structure were to be populated when the view is first opened, a user would have to wait a significant amount of time before it is displayed.

To circumvent this, the form merely populates the tree structure to the regular expression pattern level, and places a dummy node as a child to it. This ensures that the pattern node has the diamond indicator signifying that there are children nodes. When a user expands a particular pattern, the secondary query returns all the case indicators associated with the selected pattern. Once again a dummy node is inserted as a child to each node. By expanding any case indicator node, a follow-on query returns all cases that share that indicator. Finally, by clicking on the case node, opens the *Case Indicator* view in a new window.

Figure 10 – All Indicator View

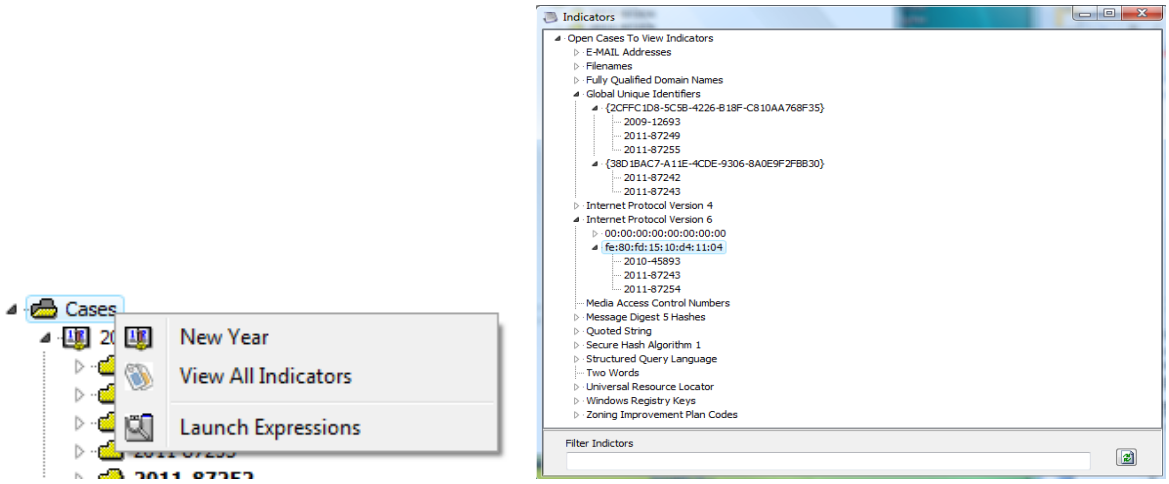
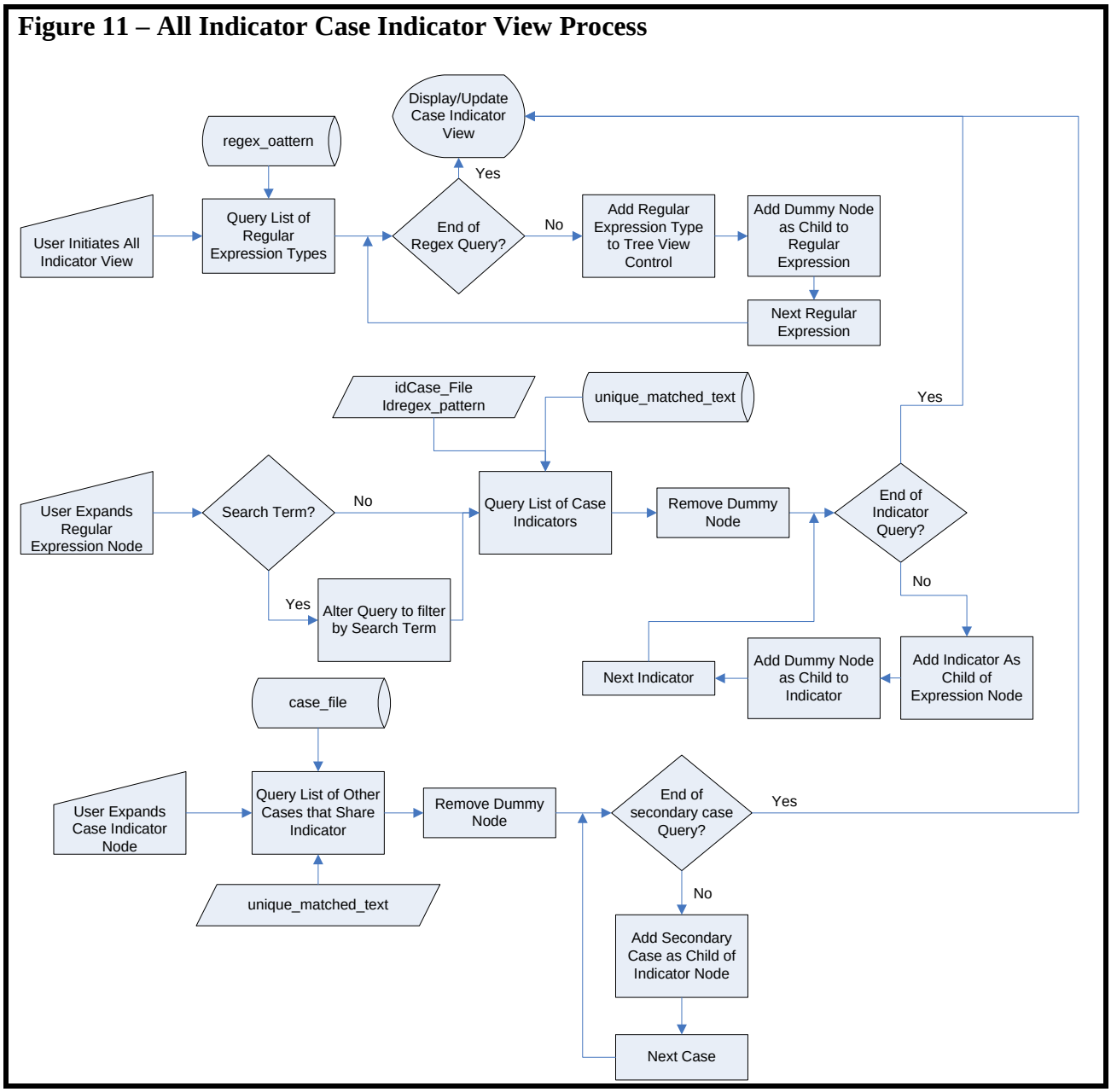


Figure 10. The All Indicator view allows a user to view all indicators, organized by type. Related cases to each case indicator are listed as child nodes, and by clicking on the case, it opens a Case Indicator View for the selected case in a new window. Additionally, the application can filter the indicators displayed with a filter.

By populating each level dynamically allows for the possibility to search for specific case indicators as the indicators are queries from the database. To accomplish this, the query that returns case indicators is designed to take a parameter that is applied using an SQL *Like* against the text of the indicator. Additionally, if the parameter does not contain the '%' character, used as a wildcard match, the application automatically pads both the front and back of the parameter with the percent sign. If however, the parameter already contains this character, then the

parameter is not padded. This allows the user several options. By not including the percent sign, the query returns any indicator with the text of the search term anywhere in the indicator. By starting the parameter with a percent, only indicators that end with the search term are returned, and finally, ending with the percent returns indicators that start with the search term. Figure 11 details the process used to create the All Indicator view.

Figure 11 – All Indicator Case Indicator View Process



Connection with Regular Expression Parser

Finally, the case viewer is dependent on the Regular Expression Parser or the Regular Expression service application to extract the case indicators from the documents maintained within the database. While the regular expression parser is a stand-alone application, and can be launched directly, it can also be launched from the case viewer application. Users can launch the parser by right clicking anywhere within the client area of the case viewer application and selecting *Launch Expressions*. As stated, the Regular Expression Parser (Expressions application) is responsible for scanning the documents for the case indicators and will be discussed in detail below.

As a precaution to not have both the regular expression parser and the regular expression service application running at the same time, which could cause issues with simultaneous access to the matched_text table, each application has been made to first confirm if either are already active. The case viewer application is also aware if the service application is active, and if it is, the *Launch Expressions* popup menu may not be visible.

Figure 12 – Launching Regular Expression Parser

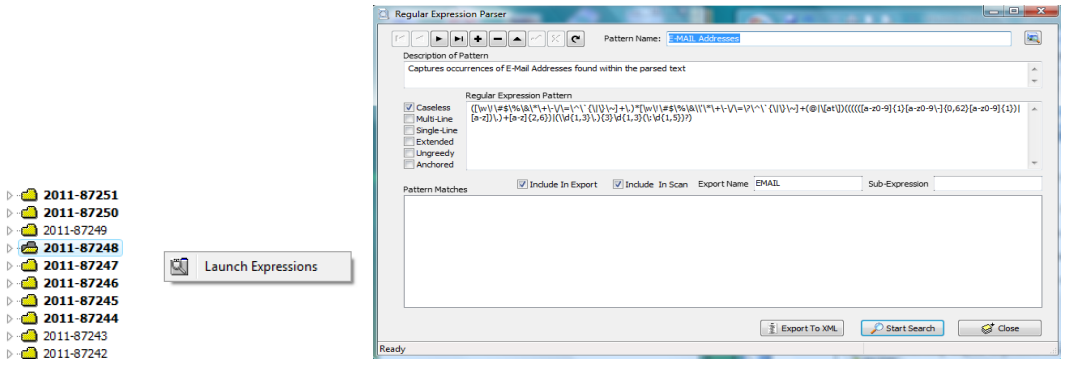


Figure 12. The regular expression parser is the heart of this research project, providing the ability to scan the documents and extract case indicators.

Regular Expressions

The actual mechanism to perform the searches utilizes regular expressions. Valid regular expressions had to be devised to extract each type of data class proposed by this research project. Specifically, regular expressions were designed that can effectively search for e-mail addresses, computer file names, fully qualified domain names, global unique identifiers, internet protocol version 4 and 6, media access control numbers, message digest five hash values, secure hash algorithm one digests, quoted strings, structured query language statements, universal resource locators, Microsoft Windows registry keys and zoning improvement plan codes were developed and tested.

It is worth noting that each regular expression was a compromise in what a particular data class allows, format wise, and what is possible to search for. Some examples that highlight the difficulty in designing a regular expression include that of searching for a file name. The file name regular expression uses a list of possible file extensions as a *token* to key the identification of a file name. The list of file extensions included to search for is quite extensive, but not exhaustive. Any file using a non-standard extension will not be flagged as a file. Additionally, both Microsoft Windows and MAC OS allow spaces as valid characters in a file name. If the regular expression were to also allow for spaces however, every character before an identified file extension could also presumably be part of the file name. E-mail address formats were also just as difficult to quantify. The user name portion of e-mail addresses allow for characters not normally associated with e-mail addresses. Even the @ character is allowed in the user name portion if encapsulated within quotes. The domain portion has several standard possibilities, such as .net or .com, however, if the regular expression is to also allow for international e-mail addresses, it must also allow for domains that include every country of the world, such as .ru for

Russia, or .us for United States. Forgoing a list of all possible domains, and strictly keying off of a period (.) followed by two or three characters in the range of a to z, again such as .ru or .com, restricts the possibility of longer domains, such as .museum, however, allowing for six characters, to allow for .museum, increases the chance for invalid matches.

Below is the list of regular expressions used by the regular expression parser application. The list also identifies some of the strengths and a few of the weaknesses associated with the regular expression, and also shows an example of a valid match.

Table 3

Regular Expressions

Data Class	Regular Expression Strength	Weakness Example Match
E-Mail Address	([\w\!\#$\%\&*\+\-\/\=\^\`{\|\}\~]+\.)*[\w\!\#$\%\&\\'*\+\-\/\=\?\^\`{|\}\~]+(@|\[at\])(((((([a-z0-9]{1}[a-z0-9\-]{0,62}[a-z0-9]{1})|[a-z])\.)+[a-z]{2,6})|(\\d{1,3}\.){3}\d{1,3}(\:\d{1,5})?)	
	Allows a wide range of valid characters for the user name portion. xxxxxxxx@gmail.com	1) Domain portion may allow more than just e-mail addresses register as a match. 2) Legal domains include IPs which are not matched.
File Names	([a-z0-9~\[\]_\-]+[a-z]?+[0-9a-z\(\)\[\]\=\/\^\%\!\{\}_\-\+~#@]?+\.(acl|asf|asm|asp|ba?k|bas|bat|bat\.tar\.gz|bcc|bcp|bd|bin|blt|bmp?|bpr|bup|cab|cad|cap|cb|cdr|cer|cfg|cgi|chm|chn|cla|cmd|cpl|cpp|csv|dat|dbf?|dcr|dcu|d1l|dll|dl_|docx?|drv|ds|dvr|ebs|egg|eml|enc|epj|evt|exe?|exe\.vir|exe\.vir\.[0-9]|exe.reg|ex_|gg|gif|gpl|gz|g?zip|h\+\+|hex|hke|hlp|hsc|hsl|ht|ico|idx|ime|inc|index\.dat\.txt|inf|ini|jar|java?|jpe?g?|jse?|jsfl|jsp|key|kog|lang|levels|lib|lnk|logs?|lpr|lrs|lzh?|mai|make?|map|mdb|mht|midi?|mpe?g|msi|nfo|oab|obj|oca|ocx|old|ora|pag|pas|pcap?|pd?f|pe|perl|pfx?|pgm|php|p?html?|pl|pm|png|ppk|pps|pps\.lnk|pptx?|pst|python|rar|raw|reg|res|rtf|sav|scr|session|\$?swf|sys|tar|tar\.gz|tcl|te?mp|tlb|txt|txt\.xls|uid|uue|vbe|vbx?|wav|xif|xlsx?|xml|xsd|wav|wma|wmv|wri|xwd|x11|00+[0-9]|7z))	

Data Class	Regular Expression Strength Example Match	Weakness
	Extensive list of file extensions	1) Not an exhaustive list of file extensions. 2) Spaces not allowed in file name. 3) Limits the possible first character against actual specifications for a file name
	digital-camera-sensor-size.ht	
Fully Qualified Domain Names	(([a-z0-9-%])+\.)([a-z0-9-%]+\.)([a-z0-9-%]+\.)?+((aero|arpa)|(biz)|(cat|com|coop)|(edu)|(gov)|(info|int)|(jobs)|(mil|mobi|museum)|(name|net)|(om|org)|(pro)|(qa)|(ru)|(travel))	
	Most domains are matched	1) List of possible domains not exhaustive. 2) New domains have been proposed
	www.lexisnexis.com	
Global Unique Identifiers	[({](0x)?[0-9A-F]{8}([-,](0x)?[0-9A-F]{4}){2}((-[0-9A-F]{4}-?[0-9A-F]{12})|(,\{0x[0-9A-F]{2}(,0x[0-9A-F]{2}){7}\}))[)}]	
	No significant strength	Validly formatted GUIDs can also look like valid MD5 values
	{2CFFC1D8-5C5B-4226-B18F-C810AA768F35}	
Internet Protocol v.4	(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9][0-9]|[1-9])\.(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9][0-9]|[1-9])\.(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9][0-9]|[1-9])\.(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9][0-9]|[1-9])	
	Ensures each octet is within the valid range.	No significant weakness
	5.1.2.1	

Data Class	Regular Expression Strength	Weakness Example Match
Internet Protocol v.6	((([0-9a-f]{1,4}:){7}[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){6}:[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){5}:([0-9a-f]{1,4}:)?[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){4}:([0-9a-f]{1,4}:){0,2}[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){3}:([0-9a-f]{1,4}:){0,3}[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){2}:([0-9a-f]{1,4}:){0,4}[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){6}((\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|(([0-9a-f]{1,4}:){0,5}:((\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|(::([0-9a-f]{1,4}:){0,5}((\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|([0-9a-f]{1,4}::([0-9a-f]{1,4}:){0,5}[0-9a-f]{1,4})|(::([0-9a-f]{1,4}:){0,6}[0-9a-f]{1,4})|(([0-9a-f]{1,4}:){1,7}:))	
	Identifies valid values even if value is collapsed.	No significant weakness
	fe:80:fd:15:10:d4:11:04	
Media Access Control Number	\s([0-9a-f][0-9a-f](:|\-)){5}([0-9a-f][0-9a-f])\s	
	No Significant strength.	Does not guarantee value is an actual MAC address in use.
	00:1F:16:72:74:51	
Message Digest 5 Hashes	\b[a-f0-9]{32}\b	
	No significant strength.	Validly formatted MD5 values can also be GUIDs
	6334afdc7d53ea45bf0fbd722e415961	
Secure Hash Algorithm 1	(\b(0x)?([a-f0-9]{40})\b)	
	No significant strength.	No significant weakness.
	2fd4e1c67a2d28fced849ee1bb76e7391b93eb12	

Data Class	Regular Expression Strength Example Match	Weakness
Structured Query Language	(SELECT\s(*|\w|((\w\s?\,\s?)+\w))\sFROM\s[\w]+)|(UPDATE\s[\w]+\sSET\s[\w\,\'\=]+)|(INSERT\sINTO\s[\d\w]+[\s\w\d\)\(\,]*\sVALUES\s\([\d\w\'\,\)]+)|(DELETE\sFROM\s[\d\w\'\=]+)	
	No significant strength. SELECT * FROM patients	Does not always match the entire SQL statement
Universal Resource Locator	((https?|hxxps?|ftp|gopher|telnet|file|notes|ms-help|www):((//)|(\\\\))[\w\d:#!%/;$~?\-=\\\.&]*)	
	No significant strength. http://www.powerhomebiz.com/062005/intelligence.htm	No significant weakness.
Windows Registry Keys	(HKEY|HKCU|HKCR|HKLM|HKU|HKCC|HK_CU|HK_CR|HK_LM|H_KU|HK_CC)\w+?(\\(\w+?))+?(\s|\r|\n|\t)	
	No significant strength. HKEY_CURRENT_CONFIG\Software\Microsoft\windows\CurrentVersion\InternetSettings	No significant weakness.
Zoning Improvement Plan Code	\b\d{5}(-\d{4})?\b	
	Allows for optional 4 digit extension to ZIP Code. 45221	Does not guarantee match is an actual ZIP Code in use by the Postal Service.

Regular Expression Parser Application

The regular expression parser application is the heart of this research project. It has a multipurpose role in that it manages the list of regular expressions, provides an avenue to test regular expressions as they are being developed, conducts the actual scan against the case documentation with the regular expressions and finally, can export the list of case indicators that

it has found in an extensible markup language (.xml) formatted export document. The application is designed as a Windows based application that can be executed directly from the Windows Start menu, or by double clicking the application icon, or by selecting *Launch Expressions* from the case viewer application popup menu. The source code for the Regular Expressio Parser can be found in Appendix D. The main graphical user interface allows a user to add, delete or modify regular expressions, and to set the options necessary to use the expression. It also allows the ability to turn on or off individual regular expressions for a particular scan as well as whether the case indicators for a particular data class should be included in the export .xml report. Below is a view of the application's main graphical user interface.

Figure 13 – Regular Expression Parser (Main Graphical User Interface)

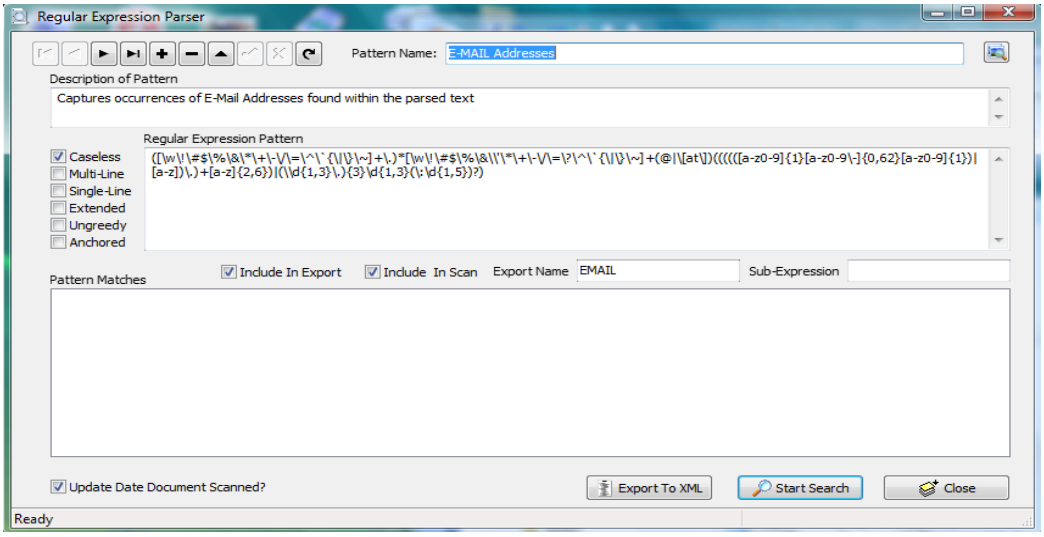


Figure 13. The application interface is divided into two main sections. The top half is dedicated towards the management of the regular expressions, while the bottom is geared toward displaying the results of the document scans.

Management of Regular Expressions

The regular expressions are a critical aspect of this project. Ensuring they are adequately maintained thus becomes an important requirement. To ensure this, they have been allocated a portion of the data model to maintain the data items associated with each regular expression.

Furthermore, the regular expression parser can access these tables to add, delete or modify each expression.

Adding a new regular expression can be done by clicking the plus (+) sign in the database navigator control in the upper left corner of the application. A *pattern name* must be entered, which is a shorthand notation to identify the type of data class the regular expression will attempt to match. The *description of pattern* is a more detailed explanation of the regular expression, and is saved within a binary large object, allowing for the possibility of maintaining detailed instructions for a particular expression. The *Include in Export* flag identifies if the case indicators it found associated with the regular expression should be included in the extensible markup language (.xml) export report, checked signifies that it should. The *Include in Scan* flag identifies if the regular expression should be included in a scan of the documents, again, checked signifies that it should. The *export name* field will be used in the .xml file to signify the data class type of the case indicators. The *sub-expression* field is used to return a substring of the entire matched text. This is important if a *token* is used to help find a match, however, the token itself should not be part of the case indicator. As an example, if it is known that names of individuals can always be found after the token '*Name:* ', then it would be reasonable to use this token to match and capture the name that follows it, however; the token itself would not be desired in the actual text of the case indicator. In this case, each sub-portion of the regular expression can be wrapped within parenthesis, and the order number of the relevant portion of the matched text is entered into the *sub-expression* field to limit the returned text. Additional aspects of a regular expression's meta-data include the selection of the options that include: *Caseless*, *Multi-Line*, *Single Line*, *Extended*, *Ungreedy* and *Anchored* which are all described in table one above. To cleanse the pattern matches, standard update and delete structured query language statements are

maintained in the post_process database table. Any regular expression can have as many queries as needed to cleanse the data as needed. These queries are also accessible from the regular expression parser application by clicking the control button located in the upper right corner of the application's graphical user interface. The queries are applied to the matched_text table before they are committed to the unique_matched_text table. This minimizes the process necessary to transfer data from the matched_text table to the unique_matched_text table, which is explained below.

Figure 14 – Example SQL Filter for Internet Protocol v.4 Pattern Matches

```
Delete From matched_text
Where (matched_text like "172.16.%" or matched_text like "172.016.%" or
        matched_text like "172.17.%" or matched_text like "172.017.%" or
        matched_text like "172.18.%" or matched_text like "172.018.%" or
        matched_text like "172.19.%" or matched_text like "172.019.%" or
        matched_text like "172.20.%" or matched_text like "172.020.%" or
        matched_text like "172.21.%" or matched_text like "172.021.%" or
        matched_text like "172.22.%" or matched_text like "172.022.%" or
        matched_text like "172.23.%" or matched_text like "172.023.%" or
        matched_text like "172.24.%" or matched_text like "172.024.%" or
        matched_text like "172.25.%" or matched_text like "172.025.%" or
        matched_text like "172.26.%" or matched_text like "172.026.%" or
        matched_text like "172.27.%" or matched_text like "172.027.%" or
        matched_text like "172.28.%" or matched_text like "172.028.%" or
        matched_text like "172.29.%" or matched_text like "172.029.%" or
        matched_text like "172.30.%" or matched_text like "172.030.%" or
        matched_text like "172.31.%" or matched_text like "172.031.%" or
        matched_text like "10.%" or matched_text like "192.168.%") and
        idregex_pattern =  (Select idregex_pattern From regex_pattern
                            Where pattern_name = 'Internet Protocol Version 4')
```

Figure 14. This query deletes IPs that would only match an organization's internal network. Connection with another cases based on these values would be superficial.

Finally, the regular expression itself is entered within the *regular expression pattern* field. This too is associated with a binary large object, allowing for arbitrarily long regular expressions.

Deleting a regular expression can be done by clicking the minus (-) sign in the database navigator in the upper left corner of the application. When clicked, the currently displayed regular expression is deleted from the database. Other operations possible from the database navigator include, go to first regular expression, go to previous, go to next and go to last. Additionally, buttons that initiate edits on the regex_pattern table, save edits to the current regular expression, cancels unsaved edits and finally, refresh the view of the regex_pattern table are available.

Testing Regular Expressions

Testing a regular expression involves the use of the *Include in Scan* flag associated with each regular expression. By simply turning off this flag for every regular expression except the one being tested, the application can parse the documents for text that matches the pattern. It is not inconceivable though, that there is in fact no match to the pattern being tested located in the database of documents. In a production environment, simply altering the documents to include text to match a pattern is not an option. It is recommended that, for a real-world solution, a test environment should be established that can be utilized to test regular expressions as they are being designed.

Scanning Procedure

The regular expression parser scans all the documents against all the regular expressions that are marked with the *Include in Scan* flag set. To start the scanning process, ensure all the regular expressions that should be included in the scan are selected and then simply click the *Start* button in the lower right corner of the application. The *pattern matches* display window is used by the application to show the actual instances of the text that is being matched while

simultaneously populating the matched_text table. The output is organized hierarchically by year, then by case and document, then by regular expression culminating in the matched text.

Figure 15 – Regular Expression Parser Output Display

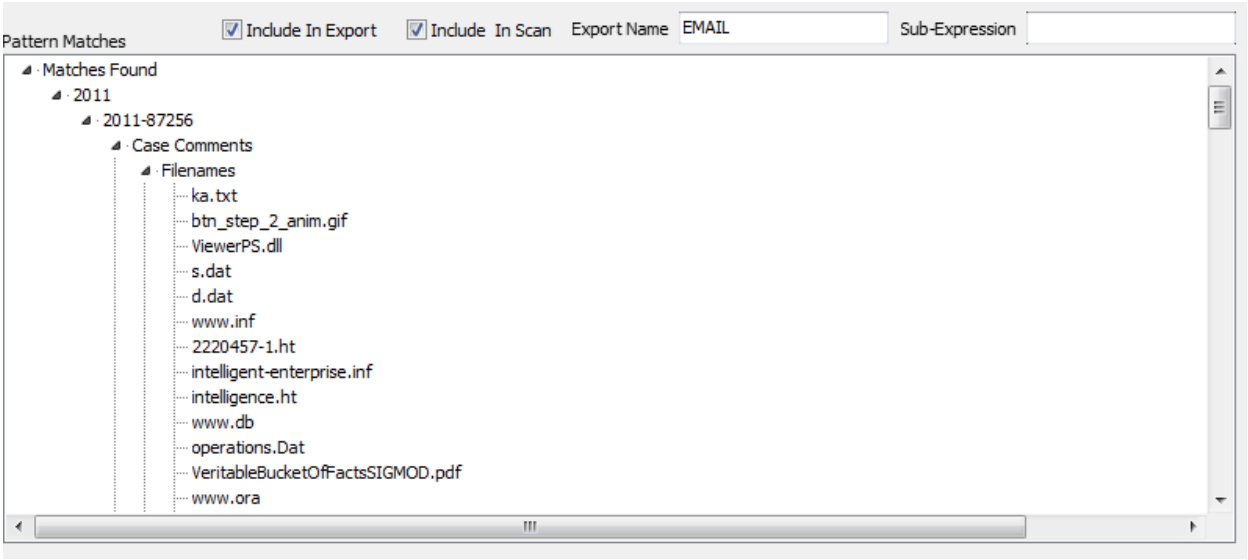


Figure 15. Every match to the regular expressions are output to the display, to include duplicates, in a hierarchical tree.

Once scanning is initiated, the application starts by querying the documents, order by year, then by case number, then by document name. By cycling through each document, it loads the document into the word processor control, which is purposely made to not be displayed on the application interface. Each document is scanned in turn, by each regular expression that is selected.

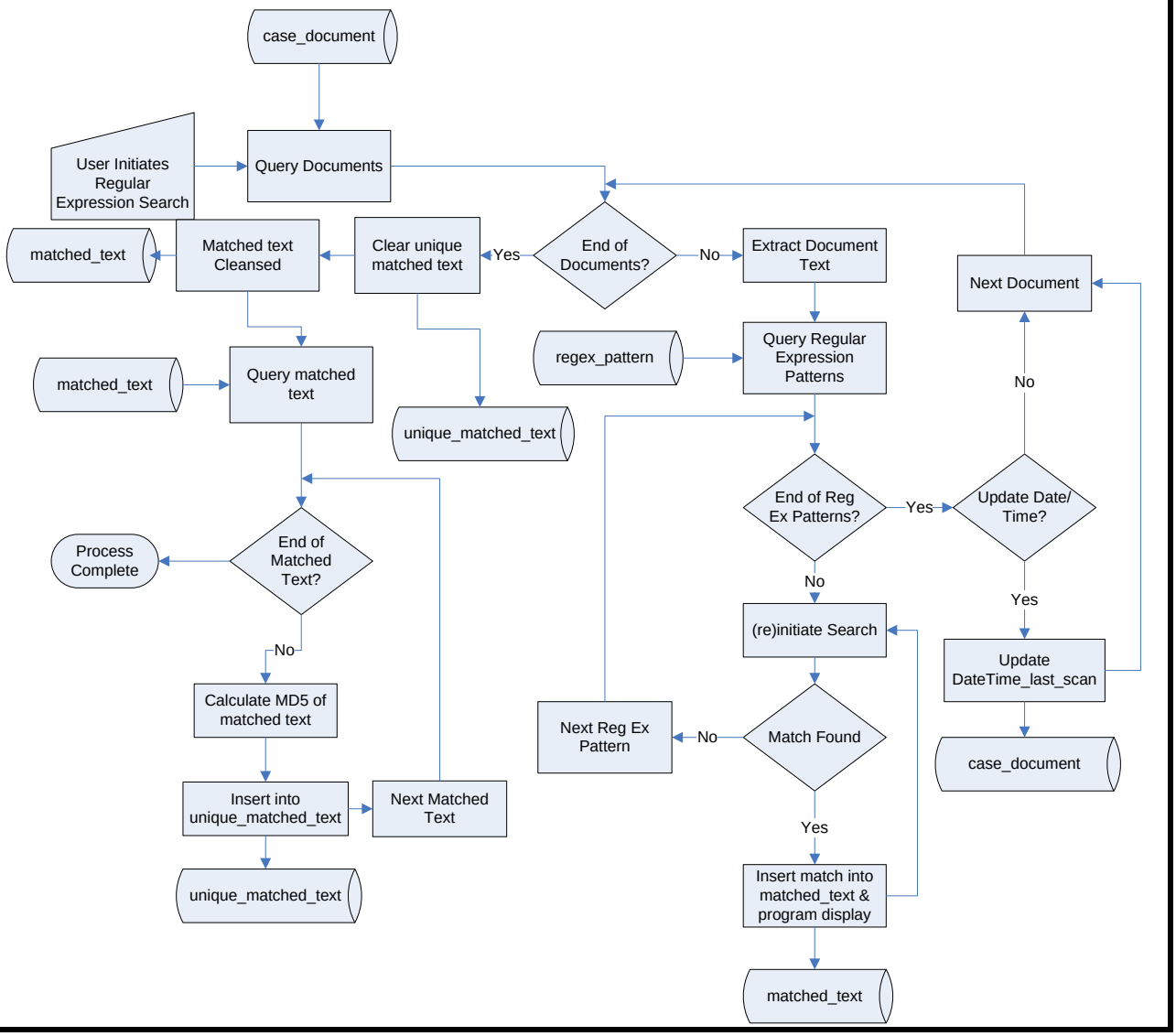
Unfortunately, documents loaded into the word processor control cannot be directly parsed as they currently are, since the document may have graphics or other embedded objects that could interfere with the scan. The word processor control has the capability to programmatically extract just the text of a document, either in ANSI or UNICODE format. For this research project, UNICODE text was extracted to be scanned. Once extracted it is assigned to the search text property of the regular expression software object within the application.

Additionally, each regular expression is also assigned to the expression property of the regular expression object individually. With both properties of the software object set, the application then initiate the search. As matches are captured, the value of the sub-expression is examined. If there are no values in the sub-expression field, a sub-expression value of zero is assumed, which defaults to the entire matched string, otherwise, only the selected substring is returned and inserted into the matched_text database table. If the *Update Date Document Scanned* flag is set, which is in the lower left corner of the application interface, the datetime_last_scan field of the case_document table is updated with the current date and time for each document scanned. Finally, when all cases have been completed, the matched text table will be populated with every string that matched the regular expression patterns. It is at this point that the filtering process executes the SQL associated with each regular expression to remove invalid instances of matched text.

The second phase of the parsing process can now commence, which is to remove duplicate instances of matched strings from the matched_text table. Starting from the first and then cycling through the entire contents of the matched_text table, the MD5 hash value of the matched text is calculated. The application then attempts to insert the value into the unique_matched_text table. Given the inclusion of the matched text's MD5 value being part of the primary key of the unique_matched_text table, a key violation error will enforce the uniqueness of matched strings in the table for a given case. The application is specifically programmed to react accordingly when the key violation occurs. This ensures that no error message is displayed to the user. Once complete, the contents of the matched_text table are emptied and all case indicators have been successfully inserted into the unique_matched_text table.

Below is a high level representation of the regular expression scanning process:

Figure 16 – Regular Expression Scanning Process



Exporting Case Indicators

An organization may have the need to share the case indicators found during the scanning process. Extensible markup language (.xml) format is an ideal format to export this type of data, since it can be easily viewed by any text editor or web browser. Pressing the *export to XML* button in the lower right corner of the application opens the export form in a new window. By pressing the *Create XML* button, the application opens the unique_matched_text table, and

produces the export report. The export report only reports on the case indicators associated with regular expressions that have the *Include in Export* flag set.

Figure 17 – XML Export Report

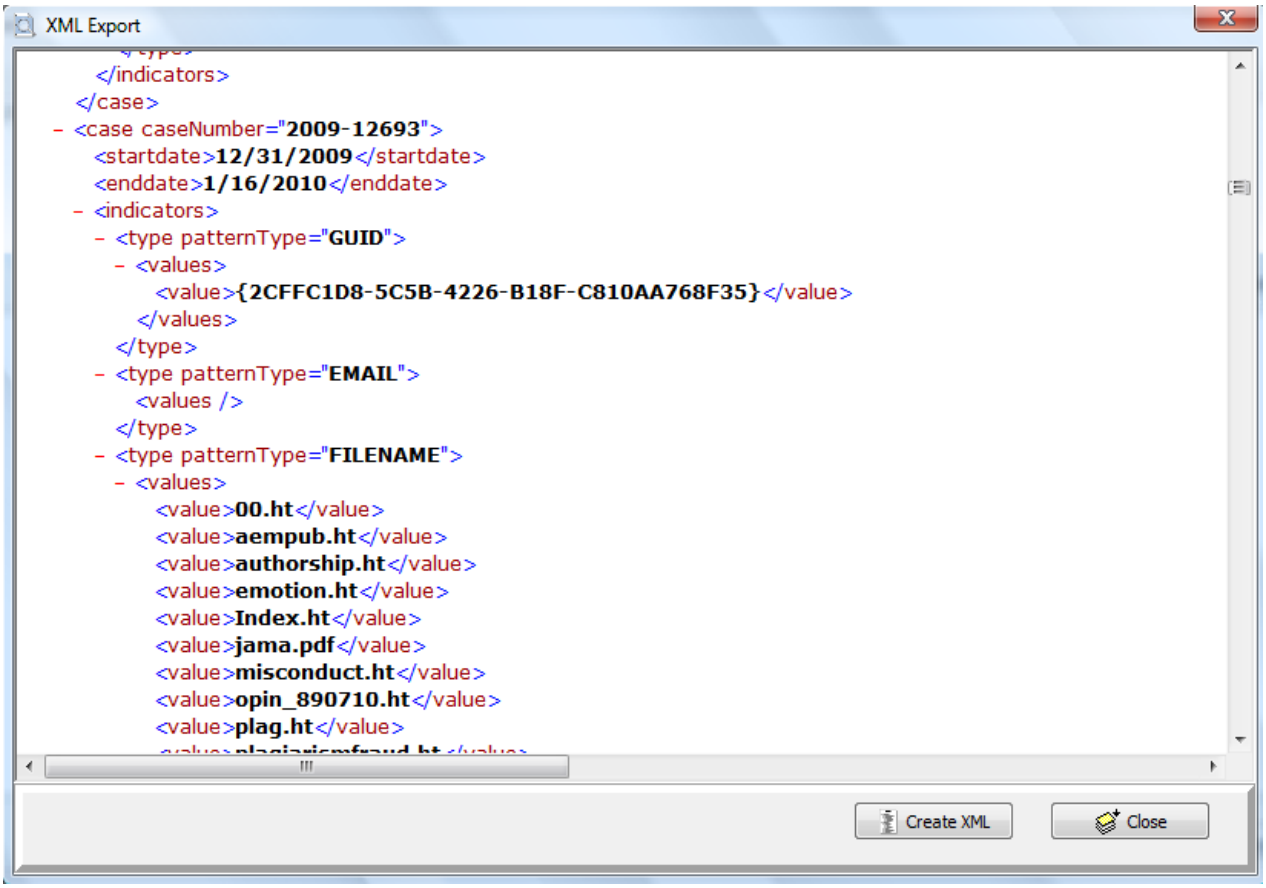


Figure 17. Export report identifies all the case indicators found associated with a case in an XML formatted document.

Regular Expression Service Application

The Regular Expression Service application is intended to replace the scanning capabilities of the regular expression parser application. For an organization that continually makes changes to their repository of documents, a manually launched scanner is a less than optimal solution. The results of a scan can be invalidated within minutes of a scan if a document were to be modified to remove text that had matched one of the regular expression patterns.

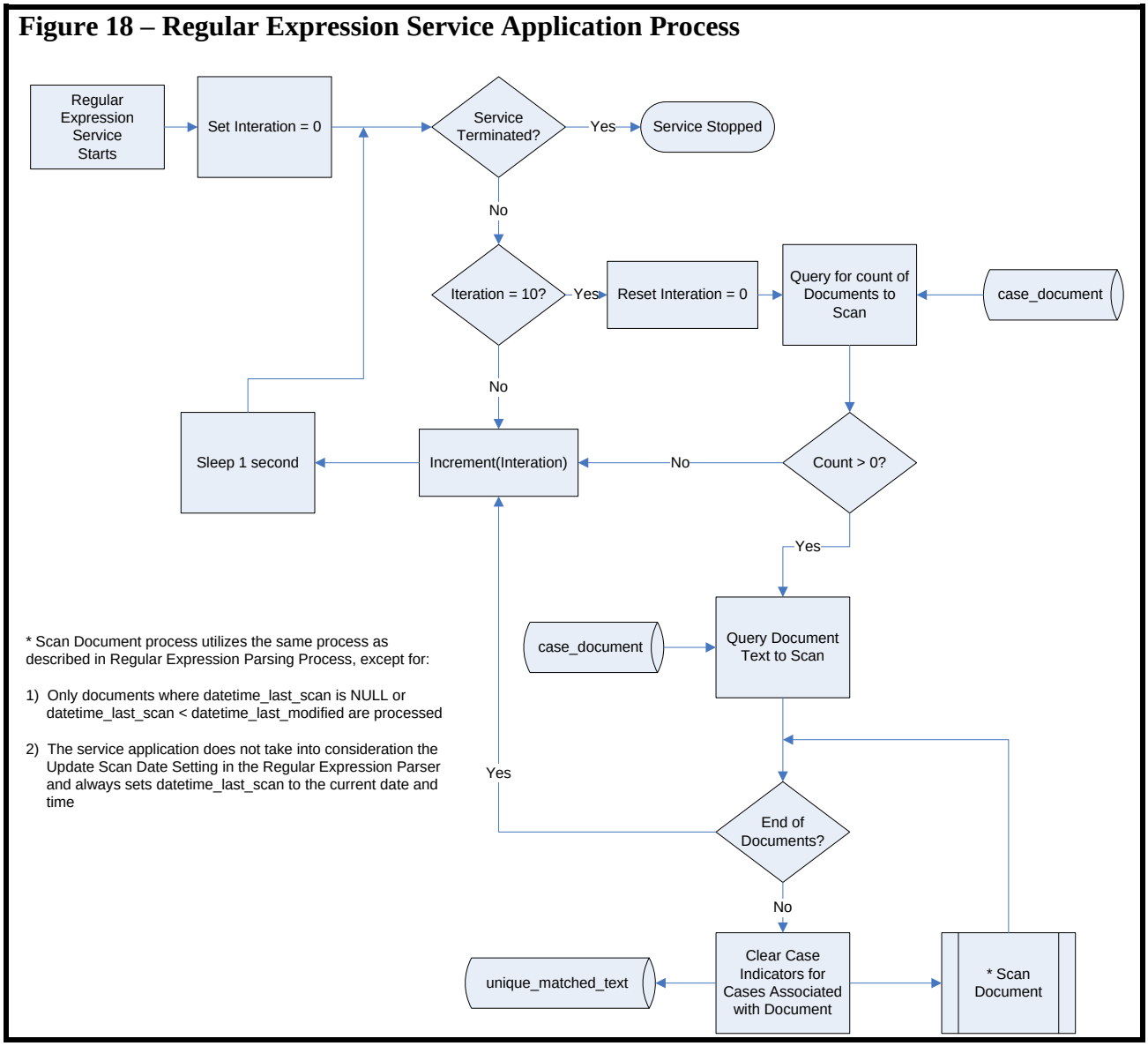
Similarly, new documents could possibly need to wait days to be scanned at all. Due to these limitations, a real-world solution would be an application that would be able to react to changes to the database as they happened. A Windows service application can provide these benefits. The source code for the service application can be found in Appendix E.

Since it is designed as a service application, it has no user interface and continually runs in the background; however, the actual method employed to conduct the scan is equivalent to the regular expression parser application. A new requirement brought about by the way the service operates is to develop a mechanism to identify which documents need to be scanned. This can easily be accomplished by selecting only those documents where the datetime_last_scan field is either NULL or has a date time stamp earlier than the datetime_last_modified. This is done by polling the database, although the process doing so is intricate. It is typical for a service application to accomplish a task, and then sleep until it needs to accomplish the task again. Having the service poll the database continually would put an undue strain on the server and the database; however, times between polling durations should not be too long causing changes to the database to go unprocessed for an extended period. For this research project, the time between polling was selected to be every ten seconds; however having the service sleep for ten seconds could make the service seem unresponsive, especially if an administrator wanted the service to be stopped, such as during a reboot. To fix this, the service actually only sleeps for one second increments, wakes and tests to see if it has been requested to terminate, and if not sleeps again, unless ten seconds have elapse, which prompts the application to poll the database once again. Another complication also arises on how new or modified documents are scanned. The validity of the case indicators held within the unique_matched_text table comes into question if a modified document has been altered to remove text that has been selected as a match to one of the

regular expressions. Due to this, when a document is added or modified within a case, all case indicators for that case are deleted from the unique_matched_text table, and every document associated with that case is rescanned. With the completion of the scan, the datetime_last_scan field for every scanned document is once again updated with the current date and time.

Below is a high level representation of the regular expression scanning process of the service application:

Figure 18 – Regular Expression Service Application Process



To utilize the service application, a server computer should be dedicated, which has access to the MySQL database instance. The application must be installed as a service on the computer and will automatically restart itself when the server is rebooted. A DOS window should be opened as an administrator to install the service. Navigating to the directory where the service is located and issuing the command: *ThesisSvc.exe /install* installs the service.

Figure 19 – Installing the Service Application

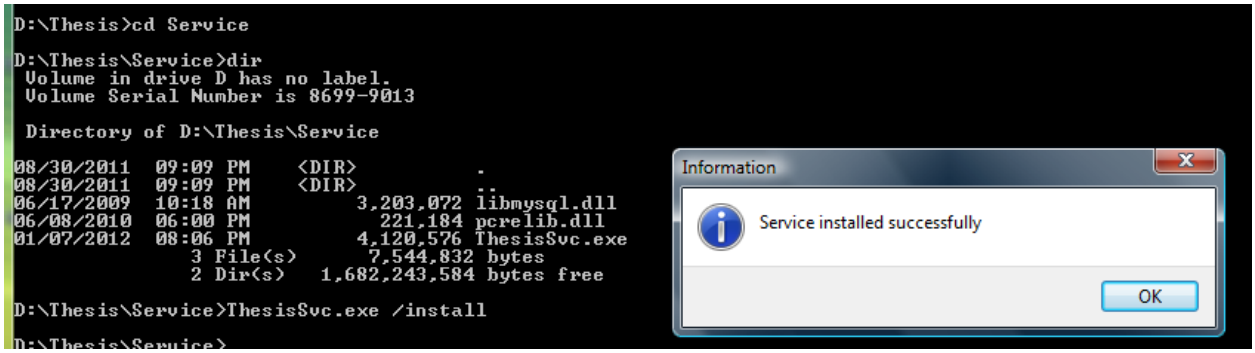


Figure 19. Installing the service makes it visible as a service in the Computer Management Window.

To uninstall a service, similarly, from a DOS windows issue the command: *ThesisSvc.exe /uninstall*.

Figure 20 – Uninstalling the Service Application

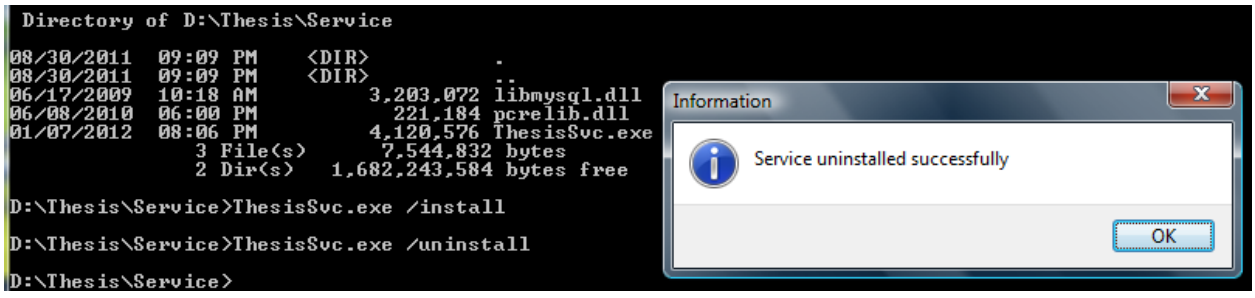


Figure 20. Uninstalling the service removes the service from the Computer Management Window.

The Windows Computer Management window can provide the status of the service application.

Figure 21 – Obtaining Service Status

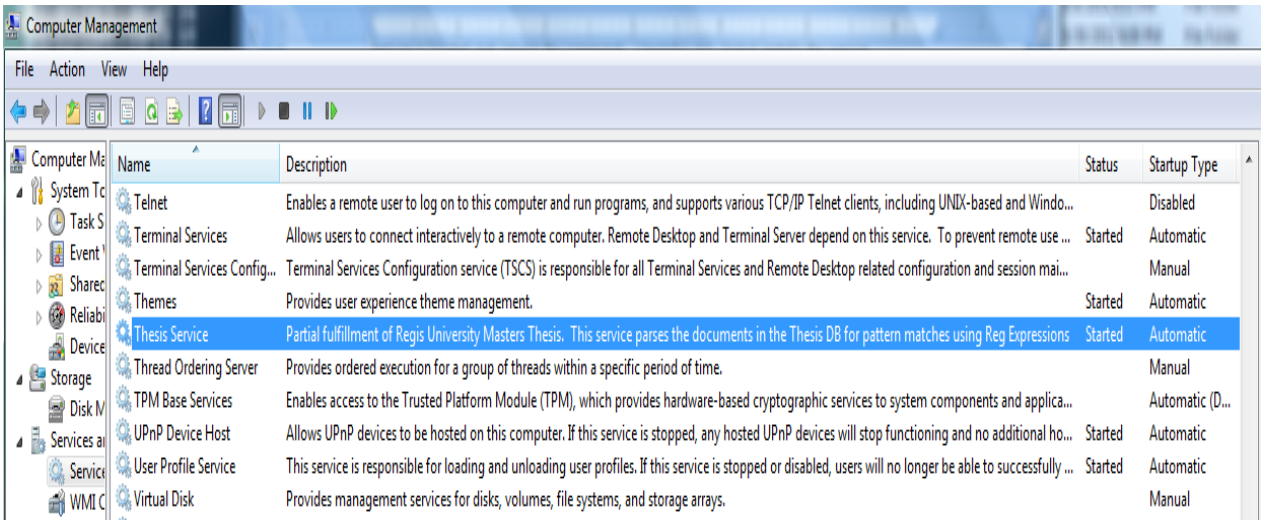


Figure 21. The service will be listed in the Computer Management window. The status should be Started, and the Startup Type should be Automatic.

A network administrator can use this view to monitor the status of the service application to ensure that it is started and that the startup type is automatic, which ensures that the service automatically restarts when the server is restarted.

Chapter 4 – Project Analysis and Results

According to the article, Data Mining Techniques, data mining efforts can be classified as generalized, characterization, association, classification, clustering or pattern matching (Han 1996). This research project attempts to utilize pattern matching, via regular expressions, to create associations among cases. The measure of success for this research project is whether the proposed process does in fact find correlations between cases where they exist, while simultaneously minimizing the possibility of connecting cases that are not related. Evaluating this research project's proposed process will involve identifying the process's strengths and weaknesses, coupled with reviewing the actual results of the document scan to see if the connections between the cases emerged naturally.

Process Fortes

An important aspect of the proposed process permits the cases and documents themselves to reveal any relevant connections without a priori knowledge of what should be searched for. This allows the relationships between cases to be discovered without any human biases a reviewer may bring with them. This not only saves an organization the man-hours involved in document reviews, but also eliminates any prior prejudices that may affect how the case documents are interpreted. Additionally, while no method can be as thorough as manually reading the documents, having an automated process, as this is, does pose a significant benefits, not gained by manually reading. Documents, once scanned, are still available for rescanning. Reprocessing can be prompted for many reasons, such as the modification of the document itself, but also possible are the refinement of the regular expression to better find its intended data class, or the definition of a new regular expression designed to find a new data classes not previously scanned for. Rescanning the entire contents of an organization's document repository is easily accomplished

via the regular expression parser, while manually re-reading the documents would be a major undertaking.

By classifying the case indicators by the regular expression (data class) that was used to identify it, gives the case indicators greater context than by simply collecting an uncategorized list of indicators. Categorization of the case indicators is accomplished within, the unique_matched_text table by having relationships with both the case_file table and the regex_pattern table. This allows for searches within the case indicators to be limited to one or more data classes. Furthermore, confidence on a correlation between different cases can be heightened by the existence of additional shared case indicators, both within or outside the scope of a single data class.

More interesting is the possibility of finding secondary correlations, through common case indicators, which results in a web of connections between cases. The results of the regular expression parsing process show that it is capable of identifying direct connections where they exist, and that it is also capable of also finding the secondary correlations between cases. Furthermore, in an actual production environment version of this research project's proposal, far beyond secondary level connections have been identified throughout multiple years of datasets. Analysts are now exploring these connections to validate if these connections are more than superficial associations.

Results Verifying Case Correlations

As described in Chapter 3, the case viewer application has two specific views that illustrate the case indicators categories by the regular expression used to find them. The first view is case centric, while the secondary takes on a more global view, providing all indicators, solely categorized by the regular expression used to find it. This next section will attempt to illustrate

different levels of connections using those views, followed by snapshots of relationship graphs generated by visualization software on actual datasets that show the web of connection that can be attained through the process proposed in this research project.

The case indicator view is used to view indicators, related to one case. By expanding any node that has children (indicators), it then shows the actual text found with the case that matched the regular expression.

Figure 22 – Case Related Indicators Showing Relationship to Secondary Case

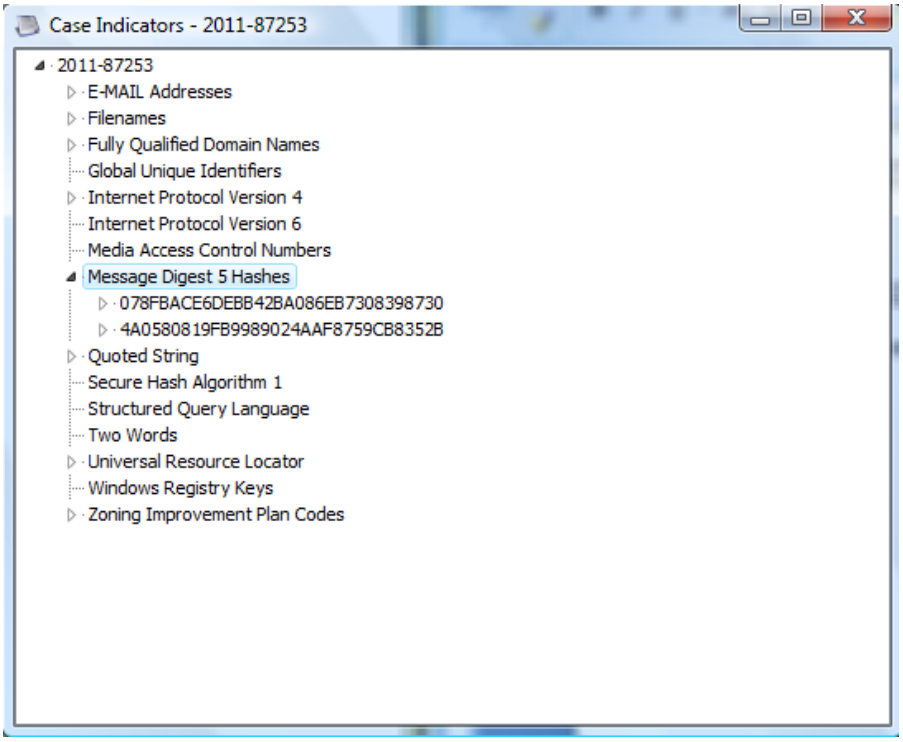


Figure 22. This view shows the indicators associated with case 2011-87253. The MD5 node has been expanded to illustrate actual matches extracted during the scan.

Each node that has related sub-nodes, are identified with diamonds. Expanding the nodes associated with the MD5 case indicators will show other cases that share that same indicator.

Figure 23 – Related Secondary Case

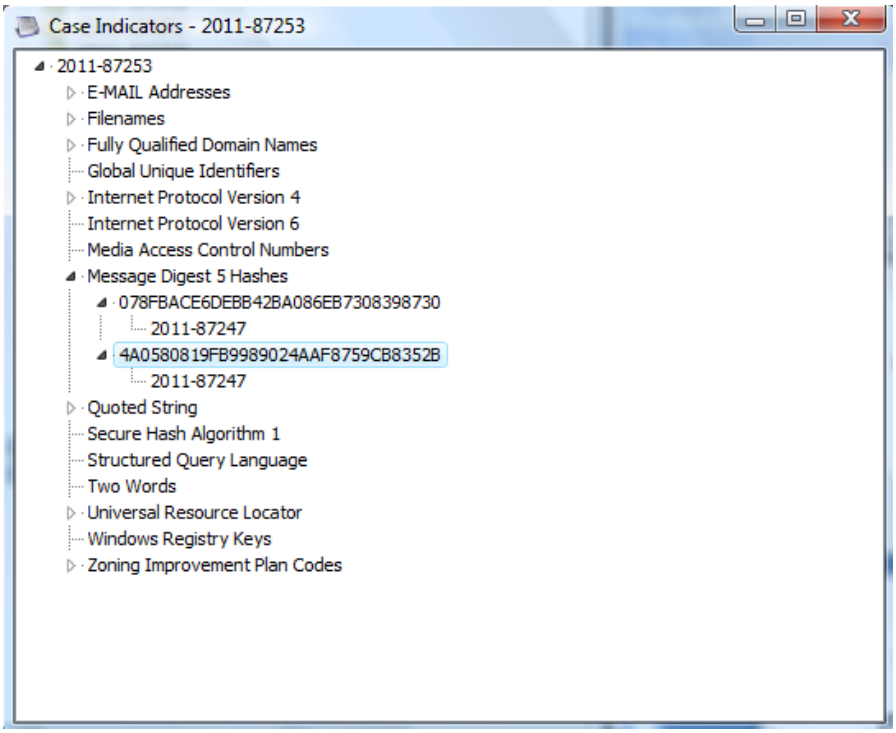


Figure 23. This view shows that case 2011-87247 share two instances of MD5 values with case 2011-87253.

By opening the same case indicator view for the related case, shows the same association to the original case using MD5 values as the link.

Figure 24 – Secondary Case, Confirmation of Association

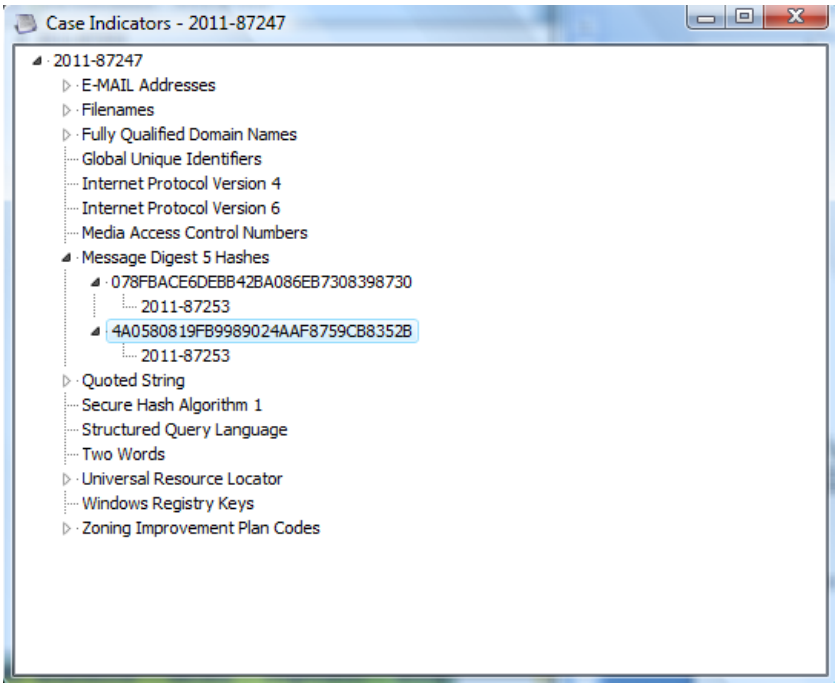


Figure 24. This view confirms that the relationship was established in both directions.

Illustrating secondary relationships are also possible using this view. Interestingly, the next examples, shows two cases that do not directly share any indicators, but are linked via a chain of indicators. In this example, the first two cases share a GUID, followed by the second case sharing an MD5 value with a third case, which chains to a final case using that same MD5 value.

Figure 25 – Demonstration of Relationship Chain

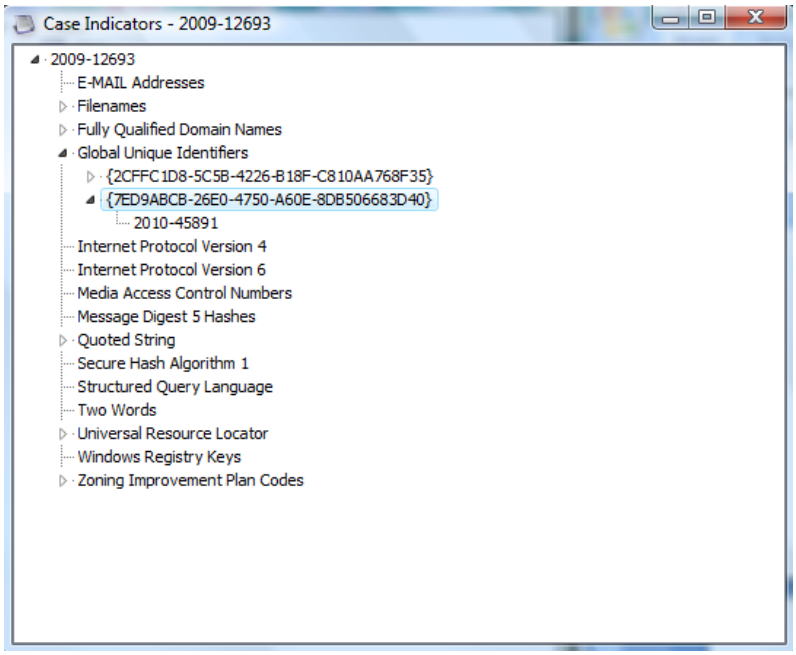


Figure 25. This view shows case 2009-12693 shares a GUID with case 2010-45891.

Figure 26 – Relationship Chain, Secondary Case

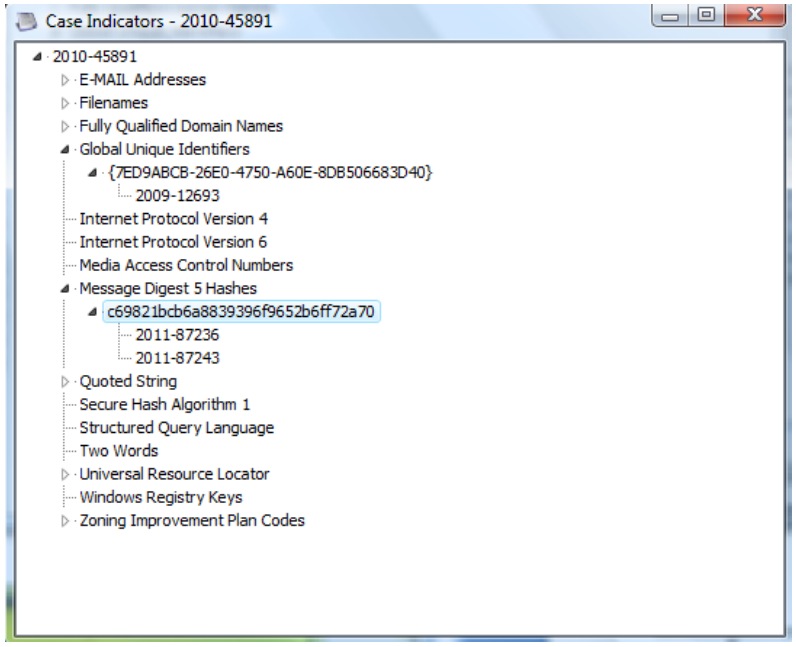


Figure 26. This view shows related case 2010-45891 that is also related to 2011-87342 via MD5.

Figure 27 – Relationship Chain, Related Case with no Direct Link to Original

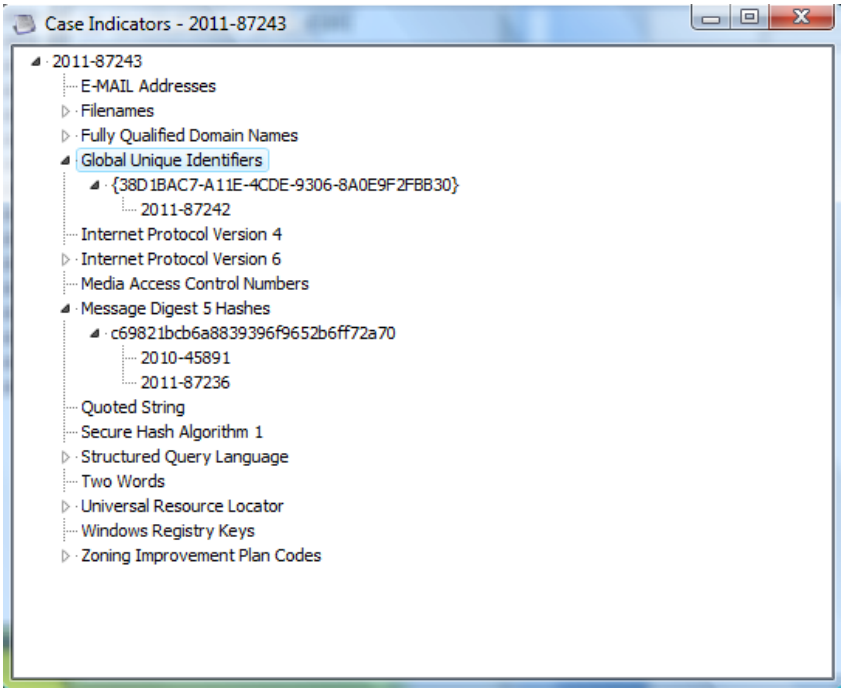


Figure 27. Finally, case 2011-87243 shows the same relationship with 2010-45891, but does not share GUID with original case. Following the link to 2011-87242 furthers the chain of relationships developing a web of relationships.

Finally, using visualization software, a true picture of the types of relationship webs that are extracted, can be fully illustrated. These relationship graphs detailed below were created using live data sets derived from the Department of Defense Cyber Crime Center's Defense Computer Forensics Laboratory, located in Linthicum Maryland. The Department of Defense Cyber Crime Center provides digital evidence processing, supporting fraud investigations and counter intelligence activities (Department of Defense Cyber Crime Center, 2012). The case indicators are related to cases of malware and network intrusions identified by Department of Defense contractors, which can pose a serious national security risk. Forensic examiners are responsible for determining the extent of the intrusion, or damage assessment of malware attacks, and create document artifacts that detail the findings of their examinations. These documents are scanned using a similar process proposed by this research project.

Figure 28 – MD5 Relationship Graph

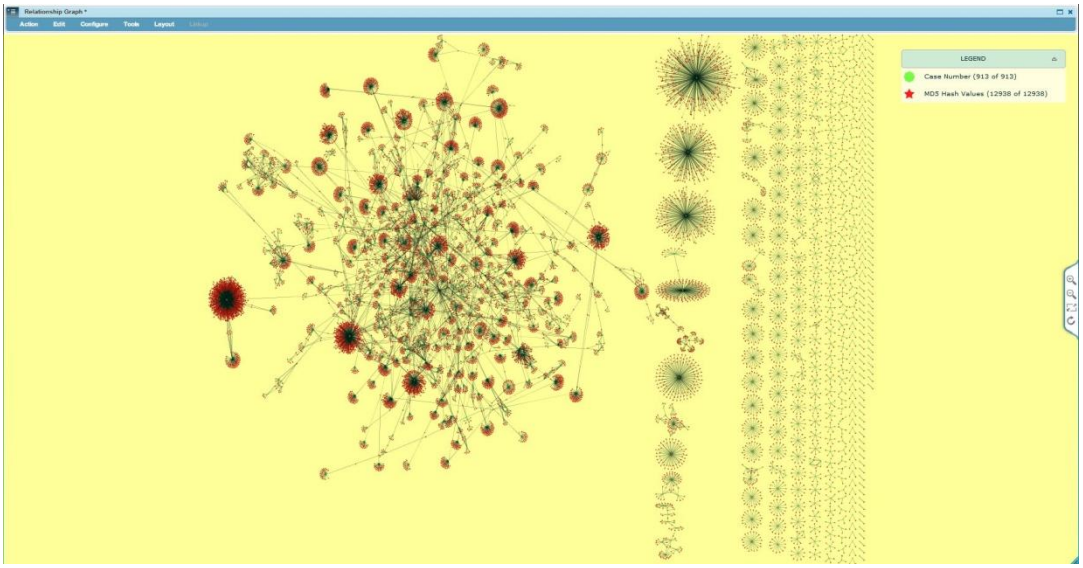


Figure 28. Green nodes represent cases, red nodes represent MD5 values. MD5 values are a major data point for forensic examinations. They can be associated with snippets of code, a file or series of files.

Figure 29 – URL Relationship Graph

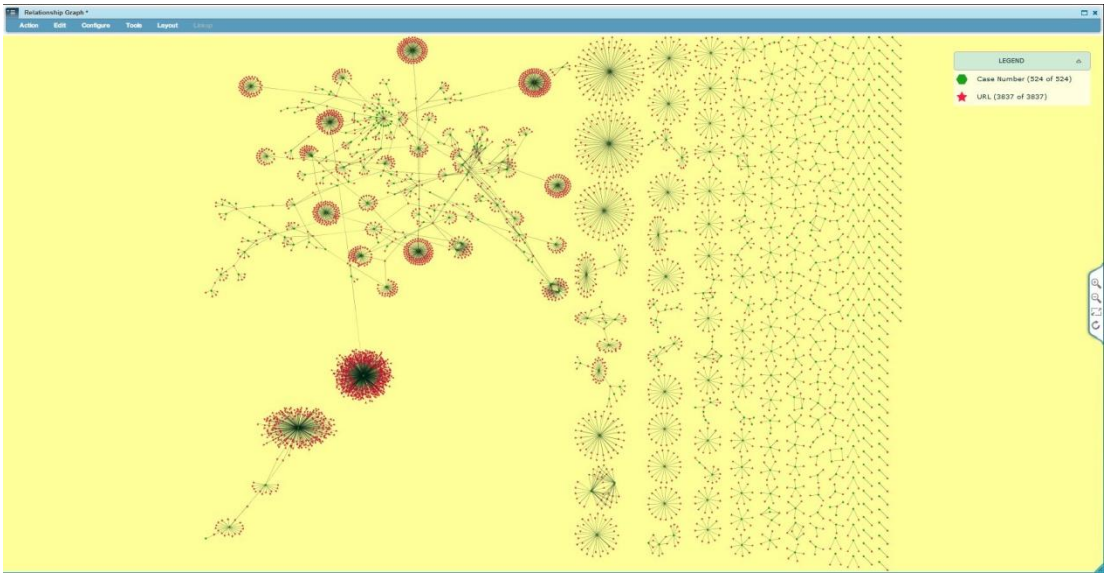


Figure 29. Greed nodes represent cases, red nodes represent URL values. URLs are often indicators of where network data is ex-filtrated to by malware and network intruders.

Process Weaknesses

The process proposed by this research project is not a fool proof method to find all related cases, nor does it guarantee that the relationship it does find is based on a true connection. For all their strengths, finding case indicators via regular expressions are susceptible to the idiosyncrasies of the underlying documents they search. Context of a particular term within a document can easily change the meaning of a relationship. An example that has been experience at the forensic laboratory is cases that identify a particular case indicator, such as URL. While the URL was easily found in the text, the context with the document would actually negate any relationship. Often, cases are forwarded to the laboratory for examination with a belief that it is related to other cases by the investigator. The indicators associated with the other suspected related case is routinely searched for by the examiner, and the outcome of their search is often part of the case documentation, even if the results of the search are negative.

The data class may also be of little intelligence value given the context for an organization. For an organization where most of their business is constrained within a particular geographic area, little, if any, intelligence is gained from links established based on a ZIP code data class. Any such connections could easily be explained by the limited number of values serviced by the organization. Additionally, certain values of a particular data class may also be of minimal importance. Examples of IPv.4 addresses with minimal intelligence value were already described in chapter three; however, these values are easily understood by IP domain experts. Another, not so obvious example may be the MD5 or SHA1 hash value for an empty file, which will always evaluate to D41D8CD98F00B204E9800998ECF8427E for MD5 or DA39A3EE5E6B4B0D3255BFEF95601890AFD80709 for the SHA. Any relationship based on these values is superficial, at best.

Chapter 5 – Conclusions

This research project has shown a proof of concept for utilizing regular expression to search unstructured text for data classes, followed with a method to use the indicators found as a gauge towards determining the likelihood of correlations between cases. As stated, the Department of Defense Computer Forensic Laboratory is already utilizing this method, and has disseminated to their customers, the indicator specific to the cases they have submitted. Defense contractors have used the information provided via .xml export files to harden their internal networks against future attack.

Future Work

Further refinements to this process may increase its success rate at finding case relationships or reveal new aspects of a case or series of cases. The following are specific areas that can be explored in that effort.

Case Tags

Introducing tags associated with cases may increase the confidence of possible relationship between cases. Case tags would be defined as key words discovered directly via human interaction as the document is created and entered into a central repository of tags. Any case could be associated with numerous tags, with tags also capable of being related to numerous cases. To exploit this new data point, case indicators realized through the regular expression search process could also be cross compared against the tags, adding the possibility of relationships discovered through the case indicator to tag connection.

Time Lines

Many of the relationships discovered at the forensics lab point to the evolution of malware through time. Many of these malware applications retain the core of their code, but incorporate

new functionality to circumvent countermeasures or simply to improve upon its operation. As a result of this natural evolution, it is not uncommon to have several versions of the same root malware. By plotting each related case chronologically as they happened with visualization software, a timeline graph may indicate how a particular malware has progressed.

References

Barton, D. (n.d.). Opensource:dcpcrypt. Retrieved Aug 30, 2011, from City In The Sky: http://www.cityinthesky.co.uk/opensource/dcpcrypt

Chakaravarthy, V., Gupta, H., Roy, P., Mohania, M., (September 2006), Efficiently Linking Text Documents with Relevant Structured Information, *Proceedings of the 32nd International Conference on Very Large Data Bases*, pp.667-678, Retrieved From http://delivery.acm.org.dml.regis.edu/10.1145/1170000/1164185/p667-chakaravarthy.pdf?ip=207.93.211.102&CFID=37082803&CFTOKEN=93523660&__acm__=1311727767_840035895e5ca1a26c725e25e20104e7

Chen, H., Zeng D., Atabakhsh H., Wyzga W., Schroeder J., (January 2003), CopLink, Managing Law Enforcement Data and Knowledge, *Communications of the ACM*, pp.28-34. doi:10.1145/602421.602441

Department of Defense Cyber Crime Center. (n.d.). Retrieved January 13, 2012, from Defense Computer Forensic Laborator Mission: http://www.dc3.mil/dcfl/dcflMission.php

Fan, W., Wallace, L., Rich, S., Zhang, Z., (September 2006), Tapping the Power of Text Mining, *Communications of the ACM - Privacy and security in highly dynamic systems*, pp. 77-82, doi: 10.1145/1151030.1151032

Feldman, R., (September 2006), Mining Unstructured Data, *Proceeding KDD '99 Tutorial notes of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, doi: 10.1145/312179.312192

Gajic, Z. (n.d.). Free Delphi Components, With Source Code! Retrieved Aug 30, 2011, from About.Com Delphi: http://delphi.about.com/library/bluc/ucvcl.htm

Han, j. (1996, June). Data Mining Techniques. SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data , 545, doi: 10.1145/235968.280351

Laboratory Services. (n.d.). Retrieved July 11, 2011, from Department of Defense Cyber Crime Center: http://www.dc3.mil/dcfl/dcflServices.php

McKendrick, J., (June 8, 2011). Unstructured Data: The Elephant in the Big Data Room. Retrieved June 8, 2011, from ZDNet: http://www.zdnet.com/blog/service-oriented/unstructured-data-the-elephant-in-the-big-data-room/7116

Ormerod, T., Morley, N., Ball, L., Langley, C., Spenser, C., (April 2003) Using Ethnography to Design a Mass Detection Tool (MDT) For The Early Discovery of Insurance Fraud, Proceeding CHI 2003 Extended Abstracts on Human Factors in Computing Systems, pp. 650-651. doi: 10.1145/765891.765910

Park, B., Song, I., (March 2011), Toward Total Business Intelligence Incorporating Structured and Unstructured Data, *Proceeding, BEWEB '11 Proceedings of the 2nd International Workshop on Business intelligence and the WEB*, pp. 12-19, doi: 10.1145/1966883.1966890

Regular-Expression.Info. (n.d.). Retrieved Aug 31, 2011, from http://www.regular-expressions.info/delphi.html

Thollot, R., Brauer, F., Barczynski, W. Aufaure, M., (March 2010), Text-to-Query: Dynamically Building Structured Analytics to Illustrate Textual Content, *Proceedings of the 2010 EDBT/ICDT Workshops*, pp.1-4, doi: 10.1145/1754239.1754255

What is TRichView. (n.d.). Retrieved Aug 30, 2011, from TRichView: http://www.trichview.com/

ZeosLib Portal. (n.d.). Retrieved Aug 30, 2011, from ZeosLib open-source tools for your database solutions: http://zeos.firmos.at/

Appendix A – Database Creation Script

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
        FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `thesis` ;
CREATE SCHEMA IF NOT EXISTS `thesis` DEFAULT CHARACTER SET ucs2 COLLATE
        ucs2_unicode_ci ;
USE `thesis` ;

-- -----------------------------------------------------
-- Table `thesis`.`year`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`year` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`year` (
  `idyear` INT(11) NOT NULL AUTO_INCREMENT ,
  `year` VARCHAR(4) CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT NULL ,
  PRIMARY KEY (`idyear`) ,
  UNIQUE INDEX `year_UNIQUE` (`year` ASC) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`case_file`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`case_file` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`case_file` (
  `idcase` INT(11) NOT NULL AUTO_INCREMENT ,
  `idyear` INT(11) NOT NULL ,
  `case_number` VARCHAR(10) CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT
        NULL ,
  `case_start_date` DATE NOT NULL ,
  `case_close_date` DATE NULL DEFAULT NULL ,
  PRIMARY KEY (`idcase`, `idyear`) ,
  UNIQUE INDEX `case_number_UNIQUE` (`case_number` ASC) ,
  INDEX `fk_case_file_year1` (`idyear` ASC) ,
  CONSTRAINT `fk_case_file_year1`
    FOREIGN KEY (`idyear` )
    REFERENCES `thesis`.`year` (`idyear` )
    ON DELETE NO ACTION
```

```
   ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`document_type`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`document_type` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`document_type` (
  `iddocument_type` INT(11) NOT NULL AUTO_INCREMENT ,
  `document_type_name` VARCHAR(45) CHARACTER SET 'ucs2' COLLATE
            'ucs2_unicode_ci' NOT NULL ,
  PRIMARY KEY (`iddocument_type`) ,
  UNIQUE INDEX `document_type_name_UNIQUE` (`document_type_name` ASC) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`case_document`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`case_document` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`case_document` (
  `idcase_document` INT(11) NOT NULL AUTO_INCREMENT ,
  `idyear` INT(11) NOT NULL ,
  `idcase` INT(11) NOT NULL ,
  `iddocument_type` INT(11) NOT NULL ,
  `document_name` VARCHAR(45) CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci'
            NOT NULL ,
  `document_text` LONGBLOB NULL DEFAULT NULL ,
  `datetime_created` DATETIME NOT NULL ,
  `datetime_last_accessed` DATETIME NOT NULL ,
  `datetime_last_modified` DATETIME NOT NULL ,
  `datetime_last_scan` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`idcase_document`, `idyear`, `idcase`) ,
  UNIQUE INDEX `udx_doc_name_type` (`idcase` ASC, `iddocument_type` ASC,
            `document_name` ASC) ,
  INDEX `fk_case_document_document_type1` (`iddocument_type` ASC) ,
  INDEX `fk_case_document_case_file1` (`idcase` ASC, `idyear` ASC) ,
  CONSTRAINT `fk_case_document_case_file1`
    FOREIGN KEY (`idcase` , `idyear` )
```

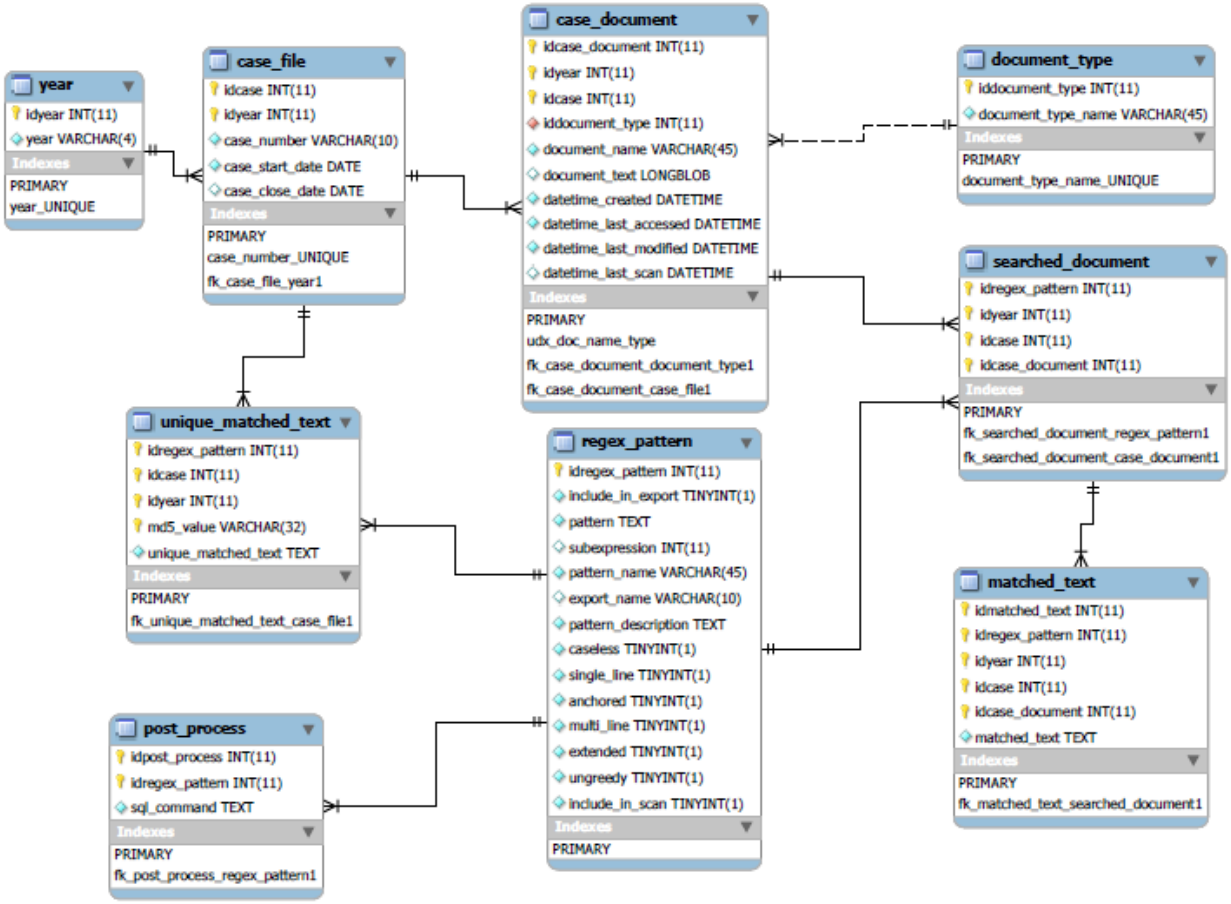
```
    REFERENCES `thesis`.`case_file` (`idcase` , `idyear` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_case_document_document_type1`
    FOREIGN KEY (`iddocument_type` )
    REFERENCES `thesis`.`document_type` (`iddocument_type` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`regex_pattern`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`regex_pattern` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`regex_pattern` (
  `idregex_pattern` INT(11) NOT NULL AUTO_INCREMENT ,
  `include_in_export` TINYINT(1) NOT NULL DEFAULT '1' ,
  `pattern` TEXT CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT NULL ,
  `subexpression` INT(11) NULL DEFAULT NULL ,
  `pattern_name` VARCHAR(45) CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT
        NULL ,
  `export_name` VARCHAR(10) CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NULL
        DEFAULT NULL ,
  `pattern_description` TEXT CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT
        NULL ,
  `caseless` TINYINT(1) NOT NULL DEFAULT '1' ,
  `single_line` TINYINT(1) NOT NULL DEFAULT '0' ,
  `anchored` TINYINT(1) NOT NULL DEFAULT '0' ,
  `multi_line` TINYINT(1) NOT NULL DEFAULT '0' ,
  `extended` TINYINT(1) NOT NULL DEFAULT '0' ,
  `ungreedy` TINYINT(1) NOT NULL DEFAULT '0' ,
  `include_in_scan` TINYINT(1) NOT NULL DEFAULT '1' ,
  PRIMARY KEY (`idregex_pattern`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`searched_document`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`searched_document` ;
```

```
CREATE  TABLE IF NOT EXISTS `thesis`.`searched_document` (
  `idregex_pattern` INT(11) NOT NULL ,
  `idyear` INT(11) NOT NULL ,
  `idcase` INT(11) NOT NULL ,
  `idcase_document` INT(11) NOT NULL ,
  PRIMARY KEY (`idregex_pattern`, `idyear`, `idcase`, `idcase_document`) ,
  INDEX `fk_searched_document_regex_pattern1` (`idregex_pattern` ASC) ,
  INDEX `fk_searched_document_case_document1` (`idcase_document` ASC, `idyear` ASC,
              `idcase` ASC) ,
  CONSTRAINT `fk_searched_document_case_document1`
    FOREIGN KEY (`idcase_document` , `idyear` , `idcase` )
    REFERENCES `thesis`.`case_document` (`idcase_document` , `idyear` , `idcase` )
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_searched_document_regex_pattern1`
    FOREIGN KEY (`idregex_pattern` )
    REFERENCES `thesis`.`regex_pattern` (`idregex_pattern` )
    ON DELETE CASCADE
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`matched_text`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`matched_text` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`matched_text` (
  `idmatched_text` INT(11) NOT NULL AUTO_INCREMENT ,
  `idregex_pattern` INT(11) NOT NULL ,
  `idyear` INT(11) NOT NULL ,
  `idcase` INT(11) NOT NULL ,
  `idcase_document` INT(11) NOT NULL ,
  `matched_text` TEXT CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT NULL ,
  PRIMARY KEY (`idmatched_text`, `idregex_pattern`, `idyear`, `idcase`, `idcase_document`) ,
  INDEX `fk_matched_text_searched_document1` (`idregex_pattern` ASC, `idyear` ASC,
              `idcase` ASC, `idcase_document` ASC) ,
  CONSTRAINT `fk_matched_text_searched_document1`
    FOREIGN KEY (`idregex_pattern` , `idyear` , `idcase` , `idcase_document` )
    REFERENCES `thesis`.`searched_document` (`idregex_pattern` , `idyear` , `idcase` ,
              `idcase_document` )
    ON DELETE CASCADE
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`post_process`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`post_process` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`post_process` (
  `idpost_process` INT(11) NOT NULL AUTO_INCREMENT ,
  `idregex_pattern` INT(11) NOT NULL ,
  `sql_command` TEXT CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT NULL ,
  PRIMARY KEY (`idpost_process`, `idregex_pattern`) ,
  INDEX `fk_post_process_regex_pattern1` (`idregex_pattern` ASC) ,
  CONSTRAINT `fk_post_process_regex_pattern1`
    FOREIGN KEY (`idregex_pattern` )
    REFERENCES `thesis`.`regex_pattern` (`idregex_pattern` )
    ON DELETE CASCADE
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;


-- -----------------------------------------------------
-- Table `thesis`.`unique_matched_text`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `thesis`.`unique_matched_text` ;

CREATE  TABLE IF NOT EXISTS `thesis`.`unique_matched_text` (
  `idregex_pattern` INT(11) NOT NULL ,
  `idcase` INT(11) NOT NULL ,
  `idyear` INT(11) NOT NULL ,
  `md5_value` VARCHAR(32) CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT
              NULL ,
  `unique_matched_text` TEXT CHARACTER SET 'ucs2' COLLATE 'ucs2_unicode_ci' NOT
              NULL ,
  PRIMARY KEY (`idregex_pattern`, `idcase`, `idyear`, `md5_value`) ,
  INDEX `fk_unique_matched_text_case_file1` (`idcase` ASC, `idyear` ASC) ,
  CONSTRAINT `fk_unique_matched_text_case_file1`
    FOREIGN KEY (`idcase` , `idyear` )
    REFERENCES `thesis`.`case_file` (`idcase` , `idyear` )
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_unique_matched_text_regex_pattern1`
    FOREIGN KEY (`idregex_pattern` )
    REFERENCES `thesis`.`regex_pattern` (`idregex_pattern` )
    ON DELETE CASCADE
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = ucs2
COLLATE = ucs2_unicode_ci;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Appendix B – Database Schema



Appendix C – Case Viewer Application Source Code

Regexthesis.dpr

```
program regexthesis;

uses
  Forms,
  Windows,
  Thesis_Main in 'Thesis_Main.pas' {Scanner_Main_Form},
  Document in 'Document.pas' {Case_Doc_Form},
  NewDocument in 'NewDocument.pas' {New_Doc_Form},
  CaseFile in 'CaseFile.pas' {CaseFile_Form},
  RVWordEnum in 'RVWordEnum.pas',
  NewCase in 'NewCase.pas' {New_Case_Form},
  NewYear in 'NewYear.pas' {New_Year_Form},
  AllIndicator in 'AllIndicator.pas' {Indicator_Form},
  CaseIndicator in 'CaseIndicator.pas' {CaseIndicatorForm};

{$R *.res}

begin
   CreateMutex(nil, false, '{2CFFC1D8-5C5B-4226-B18F-C810AA768F35}');
   if GetLastError = ERROR_ALREADY_EXISTS then
      SendMessage(HWND_BROADCAST,RegisterWindowMessage(
                  '{2CFFC1D8-5C5B-4226-B18F-C810AA768F35}'),0,0)
   else
   begin
      Application.Initialize;
      Application.MainFormOnTaskbar := True;
      Application.Title := 'Document Archive';
      Application.CreateForm(TScanner_Main_Form, Scanner_Main_Form);
      Application.Run;
   end;
end.
```

Thesis_Main.pas

```
unit Thesis_Main;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ZConnection, ZAbstractRODataset,
ZAbstractDataset, ZDataset, ImgList, ComCtrls, CommCtrl, ShellApi,
Menus;

type
  TScanner_Main_Form = class(TForm)
    PopupMenu1:                         TPopupMenu;
```

```
    NewYear1:                          TMenuItem;
    NewCase1:                          TMenuItem;
    NewDocument1:                      TMenuItem;
    OpenDocument1:                     TMenuItem;
    OpenCase1:                         TMenuItem;
    DeleteYear1:                       TMenuItem;
    DeleteCase1:                       TMenuItem;
    DeleteDocument1:                   TMenuItem;
    ViewIndicators1:                   TMenuItem;
    ViewAllIndicators1:                TMenuItem;
    TreeView1:                         TTreeView;
    ImageList1:                        TImageList;
    ImageList2:                        TImageList;

    ZConnection1:                      TZConnection;

    YearDS:                            TDataSource;
    YearQuery:                         TZQuery;
    YearQueryyear:                     TWideStringField;
    YearQueryidyear:                   TIntegerField;

    CaseQueryDS:                       TDataSource;
    CaseQuery:                         TZQuery;
    CaseQueryidcase:                   TIntegerField;
    CaseQuerycase_number:              TWideStringField;
    CaseQueryidyear:                   TIntegerField;
    CaseQuerycase_start_date:          TDateField;
    CaseQuerycase_close_date:          TDateField;
    DocumentQuery:                     TZQuery;
    DocumentQueryidcase_document:      TIntegerField;
    DocumentQueryidyear:               TIntegerField;
    DocumentQueryidcase:               TIntegerField;
    DocumentQuerydocument_name:        TWideStringField;
    DocumentQuerydocument_type_name:   TWideStringField;

    DeleteYearQuery:                   TZQuery;
    DeleteCaseQuery:                   TZQuery;
    DeleteDocumentQuery:               TZQuery;
    DeleteUniqMatchTextQuery:          TZQuery;
    NullDateScannedQuery:              TZQuery;
    ReScanCaseDocuments1:              TMenuItem;
    N1:                                TMenuItem;
    LaunchExpressions1:                TMenuItem;

    procedure FormCreate              (Sender: TObject);
    procedure FormDestroy             (Sender: TObject);
    procedure NewYear1Click           (Sender: TObject);
    procedure NewCase1Click           (Sender: TObject);
    procedure NewDocument1Click       (Sender: TObject);
    procedure OpenCase1Click          (Sender: TObject);
    procedure OpenDocument1Click      (Sender: TObject);
    procedure TreeView1Compare        (Sender: TObject;
```

```
                                        Node1, Node2: TTreeNode;
                                        Data: Integer;
                                        var Compare: Integer);
    procedure TreeView1MouseDown      (Sender: TObject;
                                        Button: TMouseButton;
                                        Shift: TShiftState;
                                        X, Y: Integer);
    procedure DeleteYear1Click        (Sender: TObject);
    procedure DeleteCase1Click        (Sender: TObject);
    procedure DeleteDocument1Click    (Sender: TObject);
    procedure ViewAllIndicators1Click(Sender: TObject);
    procedure ViewIndicators1Click    (Sender: TObject);
    procedure ReScanCaseDocuments1Click(Sender: TObject);
    procedure LaunchExpressions1Click(Sender: TObject);
  private
    { Private declarations }
    procedure Load_Treeview;
  public
    { Public declarations }
    procedure OpenDocument            (DID: Integer);
    procedure AddNewYearNode          (YearID:  Integer;
                                        NodeText: String);
    procedure AddNewCaseNode          (Year: String;
                                        CaseFileID: Integer;
                                        NodeText: String);
    procedure AddNewDocumentNode      (Year, CaseFile: String;
                                        DocumentID: Integer;
                                        NodeText: String);
    procedure AddNode                 (ParentNode:
                                        TTreeNode; Index,
                                        ID: Integer;
                                        Text:  String);
    procedure DeleteNode              (tn:  TTreeNode);
    procedure BoldOpenCaseNodes       (Node: TTreeNode;
                                        Bold: Boolean);
    procedure BoldCaseNodes           (NodeText: String;
                                        Bold: Boolean);
    procedure WMSysCommand            (var Msg: TWMSysCommand);
                                        message WM_SYSCOMMAND;
  end;

  function NewWindowProc              (TheWindow: HWND;
                                        Msg: Integer;
                                        wParam: WPARAM;
                                        lParam: LPARAM): Longint;
                                        stdcall;

var
  Scanner_Main_Form: TScanner_Main_Form;
  OldWindowProc:     TFNWndProc;
  MyMsg:             LongInt;
```

```
implementation

{$R *.dfm}

Uses NewDocument,NewCase,NewYear,Document,CaseFile,AllIndicator,
     CaseIndicator;

function NewWindowProc(TheWindow: HWND; Msg: Integer;
                       wParam: WPARAM; lParam: LPARAM): Longint;
                       stdcall;

begin
   if Msg = MyMsg then
   begin
      SendMessage(Application.Handle, WM_SYSCOMMAND,
                  SC_RESTORE, 0);
      SetForegroundWindow(Application.Handle);
      Result := 0;
      exit;
   end;
   Result:=CallWindowProc(OldWindowProc, TheWindow, Msg,
                          wParam, lParam);
end;

procedure TScanner_Main_Form.AddNewCaseNode(Year: String;
                                            CaseFileID: Integer;
                                            NodeText: String);
var
   tn: TTreeNode;
begin
   tn := Treeview1.Items.GetFirstNode;
   tn := tn.getFirstChild;
   if tn <> nil then
   begin
      while tn.Text <> Year do
         tn := tn.getNextSibling;
      AddNode(tn, 3, CaseFileID, NodeText);
   end;
end;

procedure TScanner_Main_Form.AddNewDocumentNode(Year,
                                                CaseFile:String;
                                                DocumentID:
                                                Integer;
                                                NodeText:String);
var
   tn: TTreeNode;
begin
   tn := Treeview1.Items.GetFirstNode;
   tn := tn.getFirstChild;
   if tn <> nil then
   begin
```

```
      while tn.Text <> Year do
         tn := tn.getNextSibling;
      tn := tn.getFirstChild;
      while tn.Text <> CaseFile do
         tn := tn.getNextSibling;
      AddNode(tn, 4, DocumentID, NodeText);
   end;
end;

procedure TScanner_Main_Form.AddNewYearNode(YearID: Integer;
                                            NodeText: String);
var
   tn: TTreeNode;
begin
   tn := Treeview1.Items.GetFirstNode;
   AddNode(tn, 2, YearID, NodeText);
end;

procedure TScanner_Main_Form.AddNode(ParentNode: TTreeNode;
                                     Index, ID: Integer;
                                     Text: String);
var
   Insertedtn:  TTreeNode;
begin
   Treeview1.Items.BeginUpdate;
   Insertedtn            := Treeview1.Items.AddChild(ParentNode,
                                                     Text);
   Insertedtn.Data       := pointer(ID);
   Insertedtn.ImageIndex := Index;
   ParentNode.AlphaSort  (true);
   Insertedtn.MakeVisible;
   Treeview1.Items.EndUpdate;
end;


procedure TScanner_Main_Form.BoldCaseNodes(NodeText: String;
                                           Bold: Boolean);
var
   tn:  TTreeNode;
begin
   tn := Treeview1.Items.GetFirstNode;
   tn := tn.getFirstChild;
   while tn.Text <> copy(NodeText,1,4) do
      tn := tn.GetNextSibling;
   if tn <> nil then
      tn := tn.getFirstChild;
   while tn.Text <> NodeText do
      tn := tn.getNextSibling;
   if tn <> nil then
      BoldOpenCaseNodes(tn, Bold);
end;
```

```
procedure TScanner_Main_Form.BoldOpenCaseNodes(Node:  TTreeNode;
                                                Bold: Boolean);
var
   A:  TTVItem;
begin
   Treeview1.Items.BeginUpdate;
   A.hItem := Node.ItemId;
   A.stateMask := TVIS_BOLD;
   A.mask := TVIF_HANDLE or TVIF_STATE;
   if Bold then
      A.state := TVIS_BOLD
   else
      A.state := 0;
   Treeview_SetItem(Node.Handle, A);
   Treeview1.Items.EndUpdate;
end;

procedure TScanner_Main_Form.DeleteCase1Click(Sender: TObject);
begin
   DeleteCaseQuery.ParamByName('caseid').Value :=
                               integer(TreeView1.Selected.Data);
   DeleteCaseQuery.ExecSQL;
   DeleteNode(Treeview1.Selected);
end;

procedure TScanner_Main_Form.DeleteDocument1Click(Sender:
                                                    TObject);
begin
   DeleteDocumentQuery.ParamByName('casedocid').Value :=
                                integer(TreeView1.Selected.Data);
   DeleteDocumentQuery.ExecSQL;
   DeleteUniqMatchTextQuery.ParamByName('case').Value :=
                          integer(TreeView1.Selected.Parent.Data);
   DeleteUniqMatchTextQuery.ExecSQL;
   if Treeview1.Selected.Parent.HasChildren then
   begin
      NullDateScannedQuery.ParamByName('case').Value :=
                          integer(TreeView1.Selected.Parent.Data);
      NullDateScannedQuery.ExecSQL;
   end;
   DeleteNode(Treeview1.Selected);
end;

procedure TScanner_Main_Form.DeleteNode(tn: TTreeNode);
var
   ParentNode:  TTreeNode;
begin
   ParentNode := tn.Parent;
   tn.Delete;
   Treeview1.Selected := ParentNode;
end;
```

```
procedure TScanner_Main_Form.DeleteYear1Click(Sender: TObject);
begin
   DeleteYearQuery.ParamByName('yearid').Value :=
                                integer(TreeView1.Selected.Data);
   DeleteYearQuery.ExecSQL;
   DeleteNode(Treeview1.Selected);
end;

procedure TScanner_Main_Form.FormCreate(Sender: TObject);
begin
   MyMsg := RegisterWindowMessage(
                      '{2CFFC1D8-5C5B-4226-B18F-C810AA768F35}');
   OldWindowProc :=
              TFNWndProc(SetWindowLong(Scanner_Main_Form.Handle,
              GWL_WNDPROC, Longint(@NewWindowProc)));
   ZConnection1.Connected := True;
   Load_TreeView;
   Left := 0;
   Top  := 0;
end;

procedure TScanner_Main_Form.FormDestroy(Sender: TObject);
begin
   ZConnection1.Connected := False;
   SetWindowLong(Scanner_Main_Form.Handle, GWL_WNDPROC,
Longint(OldWindowProc));
end;

procedure TScanner_Main_Form.LaunchExpressions1Click(Sender:
                                                     TObject);

   function BackPos(Substr:  string; S:  String):  Integer;
   var
      loop:  Integer;
      ch:  string;
   begin
      BackPos := 0;
      for loop := Length(s) downto 1 do
      begin
         ch := copy(s, loop, length(substr));
         if ch = substr then
         begin
            backpos := loop;
            exit;
         end;
      end;
   end;

var
   zAppName  :  array[0..512] of char;
   path      :  String;
begin
```

```
   Path := copy(ParamStr(0), 1, backpos('\',ParamStr(0)));
   FillChar(zAppName, sizeof(zAppName), #0);
   Path := '"'+Path+'EXPRESSIONS.EXE'+'"';
   StrPCopy(zAppName, Path);
   ShellExecute(0, 'open', zAppName, nil, nil, SW_SHOW);
end;

procedure TScanner_Main_Form.Load_Treeview;
var
   RootNode, YearNode, CaseNode, DocNode:  TTreeNode;
begin
   RootNode := Treeview1.Items.Add(nil, 'Cases');
   RootNode.ImageIndex := 1;
   YearQuery.Open;
   try
      while Not YearQuery.Eof do
      begin
         YearNode            := TreeView1.Items.AddChild(
                                 RootNode, YearQueryYear.AsString);
         YearNode.ImageIndex := 2;
         YearNode.Data       :=
                                pointer(YearQueryidyear.Asinteger);
         CaseQuery.ParamByName('yearparam').Value :=
                                           YearQueryidYear.Value;
         CaseQuery.Open;
         try
            while Not CaseQuery.Eof do
            begin
               CaseNode := TreeView1.Items.AddChild(YearNode,
                                 CaseQueryCase_Number.AsString);
               CaseNode.ImageIndex := 3;
               CaseNode.Data :=
                               pointer(CaseQueryidCase.Asinteger);
               if CaseQuerycase_close_date.AsString = '' then
                  BoldOpenCaseNodes(CaseNode, True);
               DocumentQuery.ParamByName('idyearparam').Value :=
                                           YearQueryidYear.Value;
               DocumentQuery.ParamByName('idcaseparam').Value :=
                                           CaseQueryidCase.Value;
               DocumentQuery.Open;
               try
                  while Not DocumentQuery.Eof do
                  begin
                     DocNode :=
                               TreeView1.Items.AddChild(CaseNode,
                       DocumentQueryDocument_Name.AsString+
                       ' - '+
                       DocumentQuerydocument_type_name.AsString);
                     DocNode.ImageIndex := 4;
                     DocNode.Data := pointer(
                         DocumentQueryidCase_Document.AsInteger);
                     DocumentQuery.Next;
```

```
                end;
             finally
                DocumentQuery.Close;
             end;
             CaseQuery.Next;
          end;
       finally
          CaseQuery.Close;
       end;
       YearQuery.Next;
    end;
 finally
    YearQuery.Close;
 end;
 Treeview1.Selected := Treeview1.Items.GetFirstNode;
 Treeview1.Selected.Expand(False);
end;

procedure TScanner_Main_Form.NewCase1Click(Sender: TObject);
begin
 TNew_Case_Form.openNewCase(integer(TreeView1.Selected.data));
end;

procedure TScanner_Main_Form.NewDocument1Click(Sender: TObject);
var
 ChildForm:  TNew_Doc_Form;
begin
 ChildForm := TNew_Doc_Form.Create(Application);
 ChildForm.Show;
end;

procedure TScanner_Main_Form.NewYear1Click(Sender: TObject);
begin
 TNew_Year_Form.openNewYear(integer(TreeView1.Selected.data));
end;

procedure TScanner_Main_Form.OpenCase1Click(Sender: TObject);
begin
 TCaseFile_Form.OpenCaseForm(integer(TreeView1.Selected.data));
end;

procedure TScanner_Main_Form.OpenDocument(DID: Integer);
begin
 TCase_Doc_Form.OpenDocForm(DID);
end;

procedure TScanner_Main_Form.OpenDocument1Click(Sender: TObject);
begin
 screen.Cursor := crSQLWait;
 try
    TCase_Doc_Form.OpenDocForm(integer(
                               TreeView1.Selected.data));
```

```
   finally
      screen.Cursor := crDefault;
   end;
end;

procedure TScanner_Main_Form.ReScanCaseDocuments1Click(Sender:
TObject);
begin
      NullDateScannedQuery.ParamByName('case').Value := integer(
                                          TreeView1.Selected.Data);
      NullDateScannedQuery.ExecSQL;
end;

procedure TScanner_Main_Form.TreeView1Compare(Sender: TObject;
                                              Node1, Node2:
                                              TTreeNode; Data:
                                              Integer;
                                              var Compare:
                                              Integer);
begin
   if Node1.Level < 3 then
   begin
      if UpperCase(Node1.Text) > UpperCase(Node2.Text) then
         Compare := -1
      else if UpperCase(Node1.Text) < UpperCase(Node2.Text) then
         Compare := 1
      else
         Compare := 0
   end
   else
   begin
       if UpperCase(Node1.Text) < UpperCase(Node2.Text) then
         Compare := -1
      else if UpperCase(Node1.Text) > UpperCase(Node2.Text) then
         Compare := 1
      else
         Compare := 0;
   end;
end;

procedure TScanner_Main_Form.TreeView1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var
   hts:  THitTests;
   hMutex:  THandle;
begin
   hts                := TreeView1.GetHitTestInfoAt(X, Y);
   Treeview1.Selected := Treeview1.GetNodeAt(X, Y);
   if Button = mbMiddle then
   begin
      if (htOnItem in hts) then
         case Treeview1.Selected.Level of
```

```
         2: OpenCase1Click(Sender);
         3: OpenDocument1Click(Sender);
      end;
end
else if Button = mbRight then
begin
   NewYear1.Visible                                := False;
   NewCase1.Visible                                := False;
   NewDocument1.Visible                            := False;
   OpenCase1.Visible                               := False;
   OpenDocument1.Visible                           := False;
   DeleteYear1.Visible                             := False;
   DeleteCase1.Visible                             := False;
   DeleteDocument1.Visible                         := False;
   ViewIndicators1.Visible                         := False;
   ViewAllIndicators1.Visible                      := False;
   ReScanCaseDocuments1.Visible                    := False;
   hMutex := OpenMutex(MUTEX_ALL_ACCESS, False,
             'Global\901D5C9B-A3A2-4DE9-AF3B-9CEFF5290F5B');
   if hMutex = 0 then
   begin
      N1.Visible                                   := True;
      LaunchExpressions1.Visible                   := True;
   end
   else
   begin
      N1.Visible                                   := False;
      LaunchExpressions1.Visible                   := False;
   end;
   releasemutex(hMutex);
   closehandle(hMutex);
   if (htOnItem in hts) then
   begin
      case Treeview1.Selected.Level of
         0:  begin
                NewYear1.Visible                   := True;
                ViewAllIndicators1.Visible         := True;
             end;
         1:  begin
                NewCase1.Visible                   := True;
                if Not Treeview1.Selected.HasChildren then
                   DeleteYear1.Visible             := True;
             end;
         2:  begin
                OpenCase1.Visible                  := True;
                NewDocument1.Visible               := True;
                if Treeview1.Selected.HasChildren then
                begin
                   hMutex := OpenMutex(MUTEX_ALL_ACCESS,
                             False,
               'Global\901D5C9B-A3A2-4DE9-AF3B-9CEFF5290F5B');
                   if hMutex <> 0 then
```

```
                        ReScanCaseDocuments1.Visible := True;
                     releasemutex(hMutex);
                     closehandle(hMutex);
                  end;
                  if TreeView1.Selected.HasChildren then
                     ViewIndicators1.Visible         := True;
                  if Not Treeview1.Selected.HasChildren then
                     DeleteCase1.Visible             := True;
               end;
            3:  begin
                  OpenDocument1.Visible              := True;
                  if Not Treeview1.Selected.HasChildren then
                     DeleteDocument1.Visible         := True;
               end;
         end;
      end;
   end;
end;

procedure TScanner_Main_Form.ViewAllIndicators1Click(Sender:
                                                     TObject);
var
   ChildForm:  TIndicator_Form;
begin
   screen.Cursor := crSQLWait;
   try
      ChildForm := TIndicator_Form.Create(Application);
      ChildForm.Show;
   finally
      screen.Cursor := crDefault;
   end;
end;

procedure TScanner_Main_Form.ViewIndicators1Click(Sender: TObject);
begin
   screen.Cursor := crSQLWait;
   try
      TCaseIndicatorForm.OpenIndicatorWithID(
                              Integer(TreeView1.Selected.data));
   finally
      screen.Cursor := crDefault;
   end;
end;

procedure TScanner_Main_Form.WMSysCommand(var Msg: TWMSysCommand);
begin
   if Msg.CmdType = SC_CLOSE then
   begin
      if TCase_Doc_Form.DocInEditMode then
         MessageDlg('Finish Document Edits Before Exiting!',
                     mtError,[mbOK],0)
      else if TCaseFile_Form.CaseInEditMode then
```

```
         MessageDlg('Finish Case Edits Before Exiting!',
                    mtError,[mbOK],0)
      else
         Inherited;
   end
   else
      Inherited;
end;

begin
   ShowWindow(Application.Handle, SW_HIDE);
end.
```

Thesis_Main.dfm

```
object Scanner_Main_Form: TScanner_Main_Form
  Left = 0
  Top = 0
  ActiveControl = TreeView1
  Caption = 'Document Archive'
  ClientHeight = 549
  ClientWidth = 397
  Color = clBtnFace
  Constraints.MinHeight = 520
  Constraints.MinWidth = 315
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnCreate = FormCreate
  OnDestroy = FormDestroy
  PixelsPerInch = 96
  TextHeight = 13
  object TreeView1: TTreeView
    Left = 0
    Top = 0
    Width = 397
    Height = 549
    Align = alClient
    HotTrack = True
    Images = ImageList1
    Indent = 19
    PopupMenu = PopupMenu1
    ReadOnly = True
    RowSelect = True
    SortType = stText
    TabOrder = 0
    OnCompare = TreeView1Compare
    OnMouseDown = TreeView1MouseDown
  end
```

```
  object ZConnection1: TZConnection
    Protocol = 'mysql-5'
    HostName = '127.0.0.1'
    Port = 3306
    Database = 'thesis'
    User = 'thesis'
    Password = 'pass123'
    TransactIsolationLevel = tiReadCommitted
    Connected = True
    Left = 24
    Top = 8
  end
  object CaseQueryDS: TDataSource
    DataSet = CaseQuery
    Left = 152
    Top = 8
  end
  object CaseQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'select * '
      'from case_file'
      'where idyear = :yearparam'
      'order by case_number desc')
    Params = <
      item
        DataType = ftUnknown
        Name = 'yearparam'
        ParamType = ptUnknown
      end>
    Left = 88
    Top = 72
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'yearparam'
        ParamType = ptUnknown
      end>
    object CaseQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object CaseQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
    object CaseQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object CaseQuerycase_start_date: TDateField
```

```
      FieldName = 'case_start_date'
      Required = True
    end
    object CaseQuerycase_close_date: TDateField
      FieldName = 'case_close_date'
    end
  end
  object ImageList1: TImageList
    Left = 240
    Top = 72
    Bitmap = {
      494C010105000800040010001000FFFFFFFFFF10FFFFFFFFFFFFFFFF424D3600
      0000000000003600000028000000400000002000000001002000000000000020
      0000000000000000000000000000000000000000000000BF7F7F00BF7F7F00BF7F
      7F00BF7F7F00BF7F7F00BF7F7F0080404000400000009F5F5F00BF7F7F00BF7F
      7F00BF7F7F00BF7F7F00BF7F7F0000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      000000000000000000000000000000000000BF7F9F00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00BFBFBF00404040007F7F7F00EFEFEF0050505000DFDFDF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00BF7F9F0000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      000000000000000000000000000000000000BF7F9F00FFFFFF00FFFFFF00BFBF
      BF00404040007F7F7F00FFFFFF00DFDFDF00DFDFDF00EFEFEF0050505000DFDF
      DF00FFFFFF00FFFFFF00FFFFFF00BF7F9F0000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      000000000000000000000000000000000000BF7F9F00BFBFBF00404040007F7F
      7F00FFFFFF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00DFDFDF00EFEFEF005050
      5000DFDFDF00FFFFFF00FFFFFF00BF7F9F0000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000006020600007F7F7F00FFFFFF00DFDF
      DF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00DFDFDF00EFEF
      EF0050505000DFDFDF00FFFFFF00BF7FBF0000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
```

```
000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000007F3F7F009F9F9F00DFDFDF00DFDF
DF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00DFDF
DF00EFEFEF0050505000DFDFDF00BF7FBF0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000BF7FBF00BFBFBF009F9F9F00DFDF
DF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDF
DF00DFDFDF00EFEFEF00505050009F5F9F0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000009F7FDF00FFFFFF00BFBFBF009F9F
9F00DFDFDF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDF
DF00FFFFFF00EFEFEF006F6F6F0020005F0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000009F7FDF00FFFFFF00FFFFFF00BFBF
BF009F9F9F00DFDFDF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00FFFFFF00FFFF
FF00FFFFFF00BFBFBF00BFBFBF0020005F0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000007F7FFF00FFFFFF00FFFFFF00FFFF
FF00BFBFBF009F9F9F00DFDFDF00DFDFDF00FFFFFF00FFFFFF009FDF9F00FFFF
FF00FFFFFF00FFFFFF003F3F3F003F3FBF0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000007F7FFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00BFBFBF009F9F9F00FFFFFF00DF9FDF00BF5F5F009FBF9F00DFDF
DF007F7F7F007F7F7F00FFFFFF007F7FFF0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000007F7FFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00BFBFBF009F9F9F00BFBFFF00DFDFDF007F7F7F007F7F
7F00FFFFFF00FFFFFF00FFFFFF007F7FFF0000000000000000000000000000000000
```

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000007F7FDF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00BFBFBF006F6F6F007F7F7F00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF007F7FDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000007F7FDF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF007F7FDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000007F7FDF007F7FDF007F7FDF007F7F
DF003F3FDF00BFBFDF007F7FDF00BFBFDF007F7FDF007F7FDF007F7FDF00BFBF
FF007F7FDF000000BF00BFBFFF007F7FDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000007F7FBF003F3F9F007F7FBF00BFBF
DF003F3F9F00BFBFDF007F7FBF007F7FBF007F7FBF007F7FBF003F3F9F007F7F
BF00BFBFDF003F3F9F007F7FBF00BFBFDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000EFEFEF00AFAFAF00A0A0A000CFCFCF000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000DFDFDF00DFDFDF00DFDF
DF00DFDFDF00DFDFDF009F9F9F00306F6F00003F3F00003F3F00007F7F006060
6000DFDFDF00DFDFDF00DFDFDF00DFDFDF000000000000000000000000000000

```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000008080
8000808080008080800080808000808080008080800080808000808080008080
8000808080008080800080808000000000000000000000000009F9F9F008080
8000808080008080800080808000808080008080800080808000808080008080
8000DFDFDF0000000000000000000000005F5F5F0040404000404040004040
4000404040004040400000 3F3F0000BFBF0000BFBF00007F7F0000FFFF00007F
7F0020202000404040004040400 0A0A0000000000000000000000000008080
8000808080008080800080808000808080008080800080808000808080008080
8000808080008080800080808000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000080808000808080000000000003F3FDF000000BF000000
BF000000BF000000BF000000BF000000BF000000BF000000BF000000BF006060
8000808080000DFDFDF0000000000000000000000003F3F3F003F3F3F007F7F
7F007F7F7F007F7F7F003F7F7F0000BFBF00007F7F00007F7F00007F7F0000BF
BF003F3F3F003F3F3F0000000000A0A0A0000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000080808000808080000000000000000000000BFBF00FFFF
FF0000BFBF008080800000BFBF008080800000BFBF008080800000BFBF008080
800000BFBF0080808000000000000808080007F7FBF0000007F0000007F000000
7F0000007F0000007F0000007F0000007F0000003F003F3F9F007F7F7F003F3F
9F006060800080808000DFDFDF00000000000000007F7F7F000000000000000
0000000000000000000000000000003F3F0060DFDF0060DFDF0030AFAF00BFBF
BF00000000007F7F7F0000000000A0A0A0000000000000000000BFBF008080
800000BFBF008080800000BFBF008080800000BFBF008080800000BFBF008080
800000BFBF0080808000000000000808080000000000000000000000000000BF
BF00FFFFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF0000000000808080 00BFBFDF003F3F7F007F7FBF007F7F
BF007F7FBF007F7FBF007F7FBF007F7FBF007F7F9F00000000007F7F7F007F7F
7F003F3F9F006060800080808000DFDFDF00000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F007F7F7F00003F3F00007F7F007F7F7F007F7F
7F00BFBFBF007F7F7F0000000000A0A0A000000000000000000FFFFFF0000BF
BF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF0000000000808080000000000000000808080000000
000000BFBF00FFFFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF0000FFFF0000BFBF00000000000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F0000000000000000007F7F7F007F7F
7F007F7F7F003F3F9F0060608000BFBFBF00000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F0000000000000000007F7FBF00404080007F7F7F007F7F
7F00BFBFBF007F7F7F0000000000A0A0A000000000000000000BFBF0000FF
FF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF0080808000000000008080800000000000000000808080008080
80000000000000BFBF00FFFFFF0000BFBF00FFFFFF0000BFBF00FFFFFF0000BF
BF00FFFFFF0000BFBF00FFFFFF00000000000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F007F7F7F0000000000000000007F7F7F007F7F
7F007F7F7F007F7F7F003F3F9F00BFBFBF00000000007F7F7F0000000000FFBF
BF00FF7F7F00FFBFBF00000000000000007F7FBF0040408000FF7F7F00FF7F
7F00000000007F7F7F0000000000A0A0A000000000000000000FFFFFF0000BF
BF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF0000000000808080000000000000000808080008080
8000808080000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000007F7F7F00BFBFBF007F7F
```

7F007F7F7F007F7F7F007F7F7F00BFBFBF00000000000000000007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF000000000007F7F7F0000000000FFBF
BF00FF000000FFBFBF00000000000000000007F7FBF0040408000FF3F3F00FF7F
7F000000000007F7F7F0000000000A0A0A0000000000000000000000BFBF0000FF
FF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF0080808000000000008080800000000000000000008080800080
8000808080008080800080808000808080008080800080808000808080008080
8000808080008080800000000000000000000000000007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF000000000007F7F7F000000000000000
0000FF0000000000000000000000000000007F7FBF0040408000FF000000FFBF
BF000000000007F7F7F0000000000A0A0A0000000000000000000FFFFFF0000BF
BF00FFFFFF0000BFBF00FFFFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF0000000000008080800000000000000000000000000000000000
0000000000000000000000000000008080800080808000808080008080800080
8000808080000000000000000000000000000000000000000007F7F7F007F7F
BF000000BF003F3FDF007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF000000000007F7F7F0000000000FFBF
BF00FF00000000000000BFBFBF000000000003F3F7F0040408000FF7F7F00FF00
0000000000007F7F7F0000000000A0A0A0000000000000000000000000000000
0000000000000000000000000000FFFFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF0000000000008080800000000000000000000000000000000000
0000000000000000000000000000008080800080808000808080008080800080
8000808080000000000000000000000000000000000000000007F7F
7F007F7F7F007F7F7FBF000000BF003F3FDF007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF000000000007F7F7F0000000000FFBF
BF00FF00000000000000BFBFBF007F7F7F005F5F7F0060608000FF000000FF3F
3F000000000007F7F7F0000000000A0A0A0000000000000000000000000000000
0000000000000000000000000000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0080808000000000008080800000000000000000000000000000000000000000
0000000000000000000000000000008080800080808000808080008080800080
8000808080000000000000000000000000000000000000000000000000000000
00007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7FBF000000BF003F3F
DF007F7F7F007F7F7F007F7F7F00BFBFBF000000000007F7F7F00BFBFBF000000
0000000000000000000000000000000000000000000000000000000000000000
0000BFBFBF007F7F7F0000000000A0A0A0000000000000000000000000000000
0000000000000000000000000000FFFFFF0000BFBF00FFFFFF0000BFBF00FFFF
FF0000BFBF0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000007F7F7F007F7F7F003F3FDF000000BF007F7FBF007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000DFDFDF0000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000

```
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000424D3E00000000000003E000000
    2800000040000000200000000100010000000000010000000000000000000000
    0000000000000000000000FFFFFF00800100000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000
    00000000000000000000000000000000FFFFFFFFFE1FFFFFFFFFFFF8000FFFF
    E001C0070000E001C00080030000C000800000013E088000800004000008000
    800081C002008000800080C022088000800080C022088000800180003608800 0
    C003C0002408C001FC03E0002408FC01FC03F0001EF0FC03FE07F8010000FE07
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000000000000000000000000000000000
    000000000000}
end
object YearDS: TDataSource
  DataSet = YearQuery
  Left = 88
  Top = 8
end
object YearQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'select idyear, year'
    'from year'
    'order by year desc')
  Params = <>
  Left = 32
  Top = 72
  object YearQueryidyear: TIntegerField
    FieldName = 'idyear'
    Required = True
  end
  object YearQueryyear: TWideStringField
    FieldName = 'year'
    Required = True
    Size = 4
  end
end
object DocumentQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'select '
    '   idcase_document,'
    '   idyear,'
```

```
    '   idcase,'
    '   document_name,'
    '   document_type_name'
    'from'
    '   case_document,'
    '   document_type'
    'where'
      '   case_document.iddocument_type =
          document_type.iddocument_type' +
      ' and'
    '   idyear = :idyearparam and'
    '   idcase = :idcaseparam'
    'order by'
    '   document_name')
  Params = <
    item
      DataType = ftUnknown
      Name = 'idyearparam'
      ParamType = ptUnknown
    end
    item
      DataType = ftUnknown
      Name = 'idcaseparam'
      ParamType = ptUnknown
    end>
  Left = 240
  Top = 8
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'idyearparam'
      ParamType = ptUnknown
    end
    item
      DataType = ftUnknown
      Name = 'idcaseparam'
      ParamType = ptUnknown
    end>
  object DocumentQueryidcase_document: TIntegerField
    FieldName = 'idcase_document'
    Required = True
  end
  object DocumentQueryidyear: TIntegerField
    FieldName = 'idyear'
    Required = True
  end
  object DocumentQueryidcase: TIntegerField
    FieldName = 'idcase'
    Required = True
  end
  object DocumentQuerydocument_name: TWideStringField
    FieldName = 'document_name'
```

```
      Required = True
      Size = 45
    end
    object DocumentQuerydocument_type_name: TWideStringField
      FieldName = 'document_type_name'
      Required = True
      Size = 45
    end
  end
  object PopupMenu1: TPopupMenu
    Images = ImageList2
    Left = 312
    Top = 8
    object NewYear1: TMenuItem
      Caption = 'New &Year'
      ImageIndex = 2
      OnClick = NewYear1Click
    end
    object NewCase1: TMenuItem
      Caption = 'New &Case'
      ImageIndex = 3
      OnClick = NewCase1Click
    end
    object NewDocument1: TMenuItem
      Caption = 'New &Document'
      ImageIndex = 4
      OnClick = NewDocument1Click
    end
    object OpenCase1: TMenuItem
      Caption = 'Open Case'
      ImageIndex = 3
      OnClick = OpenCase1Click
    end
    object OpenDocument1: TMenuItem
      Caption = '&Open Document'
      ImageIndex = 4
      OnClick = OpenDocument1Click
    end
    object DeleteYear1: TMenuItem
      Caption = 'Delete Year'
      ImageIndex = 5
      OnClick = DeleteYear1Click
    end
    object DeleteCase1: TMenuItem
      Caption = 'Delete Case'
      ImageIndex = 5
      OnClick = DeleteCase1Click
    end
    object DeleteDocument1: TMenuItem
      Caption = 'Delete Document'
      ImageIndex = 5
      OnClick = DeleteDocument1Click
```

```
    end
    object ViewAllIndicators1: TMenuItem
      Caption = 'View All Indicators'
      ImageIndex = 7
      OnClick = ViewAllIndicators1Click
    end
    object ViewIndicators1: TMenuItem
      Caption = 'View Indicators'
      ImageIndex = 6
      OnClick = ViewIndicators1Click
    end
    object ReScanCaseDocuments1: TMenuItem
      Caption = 're-Scan Case Documents'
      ImageIndex = 8
      Visible = False
      OnClick = ReScanCaseDocuments1Click
    end
    object N1: TMenuItem
      Caption = '-'
      Visible = False
    end
    object LaunchExpressions1: TMenuItem
      Caption = 'Launch Expressions'
      ImageIndex = 9
      Visible = False
      OnClick = LaunchExpressions1Click
    end
  end
  object ImageList2: TImageList
    Left = 312
    Top = 72
    Bitmap = {
      494C01010A001000040010001000FFFFFFFFFF10FFFFFFFFFFFFFFFF424D3600
      0000000000003600000028000000400000003000000001002000000000000030
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000200000007F3F
      00003F000000000000000000000000000000000404040004040004040
      4000404040004040400040404000404040004040400040404000404040004040
      4000404040004040400040404000202020000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000005050
      5000606060006060600060606000606060006060600050303000DF3F0000FFBF
      00007F000000606060006060600000000007F7F7F00BFBFBF00A0A0A000A0A0
      A000A0A0A000A0A0A000A0A0A000A0A0A000A0A0A000A0A0A000A0A0A0009090
      9000404040006060600080808000404040000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000BFBF
      BF0000000000000000000000000000000DFBFBF00DF3F0000FFBF0000BF00
```

00007F7F7F0000000000DFDFDF00000000007F7F7F00DFDFDF00C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000B0B0B0004040
4000AFAFAF00505050009090900040404000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000BFBF
BF0000000000DFDFDF00DFDFDF00CFAFAF00DF3F0000FFBF0000BF0000007070
7000EFEFEF0000000000DFDFDF00000000007F7F7F00DFDFDF00C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000B0B0B00040404000BFBF
BF00BFBFBF006F6F6F00A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000BFBF
BF0000000000DFDFDF00CFCFCF00DF3F0000FFBF0000BF0000070707000DFDF
DF00EFEFEF0000000000DFDFDF00000000007F7F7F00DFDFDF00C0C0C000C0C0
C000B0B0B000A0A0A000A0A0A000B0B0B000B0B0B00040404000BFBFBF00BFBF
BF006F6F6F00C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000005F5F
5F007F7F7F009F9F9F00BFBFBF00CFCF8F00BF0000070707000DFDFDF00DFDF
DF00EFEFEF0000000000DFDFDF00000000007F7F7F00DFDFDF00C0C0C0009090
900040404000404040004040400080808000404040 00BFBFBF00BFBFBF006F6F
6F00C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000505050008F8F3000AFAF
3000AFAF30007F7F0000505030006060600070707000DFDFDF00DFDFDF00DFDF
DF00EFEFEF0000000000DFDFDF00000000007F7F7F00DFDFDF00707070006060
6000909090006060600090909000606060007070700 0CFCFCF006F6F6F00C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000040404000EFEF6F00DFDF9F00DFDF
6000BFBF8000AFAF70009F9F000040404000DFDFDF00DFDFDF00DFDFDF00DFDF
DF00EFEFEF0000000000DFDFDF00000000007F7F7F009F9F9F00909090006F6F
6F009F9F9F00DFDFDF00AFAFAF0060606000B0B0B00020202000B0B0B000C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000BFBF7F00FFFF7F00DFDF9F00DFDF
6000BFBF8000AFAF7000AFAF7000404000000AFAFAF00DFDFDF00DFDFDF00DFDF
DF00EFEFEF0000000000DFDFDF00000000007F7F7F006F6F6F0060606000DFDF
DF00DFDFDF00DFDFDF00DFDFDF00DFDFDF00606060005050500 0A0A0A000C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000

```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000BFBF7F00FFFF7F00DFDFDF00DFDF
6000BFBF8000AFAF7000AFAF700040400000AFAFAF00DFDFDF00DFDFDF00DFDF
DF00EFEFEF0000000000DFDFDF00000000007F7F7F00606060006F6F6F00DFDF
DF00DFDFDF00DFDFDF00DFDFDF00DFDFDF006F6F6F0060606000A0A0A000C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000007F7F4000FFFF7F00DFDFDF00DFDF
6000DFDF6000DFDF6000DFDF600040402000CFCFCF0000000000000000000000
00000000000000000000DFDFDF00000000007F7F7F00606060006F6F6F00EFEF
EF00EFEFEF00EFEFEF00EFEFEF00EFEFEF007F7F7F0060606000A0A0A000C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000020202000DFDF9F00DFDF9F00FFFF
7F00DFDF6000DFDF6000707030007070700 0DFDFDF00DFDFDF00000000000000
00000000000000000000DFDFDF00000000007F7F7F00606060006F6F6F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF006F6F6F0060606000C0C0C000C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000020202000707070000BFBF
7F00BFBF7F005F5F5F008F8F8F0000000000000000000000000000000000BFBF
BF007F7F7F006F6F6F0020202000000000007F7F7F007F7F7F00909090004040
4000404040004040400040404000404040009090900060606000C0C0C000C0C0
C000C0C0C000C0C0C000A0A0A000404040000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000BFBF
BF0000000000000000000000000000000000000000000000000000000000BFBF
BF00EFEFEF007070700000000000000000003F3F3F007F7F7F00000000009090
9000000000000000000000000000909090000000000007F7F7F00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00EFEFEF0050505000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000BFBF
BF0000000000000000000000000000000000000000000000000000000000BFBF
BF007070700000000000000000000000FFFFFF00FFFFFF00FFFFFF000000
0000606060006060600060606000000000FFFFFF0000000000FF000000FF00
0000FF000000FF0000007F0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000009F9F
9F00BFBFBF00BFBFBF00BFBFBF00BFBFBF00BFBFBF00BFBFBF00BFBFBF009F9F
```

9F000000000000000000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00000000000000
00000000000000000000000000FFFFFF0000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000BF7F7F00BF7F7F00BF7F
7F00BF7F7F00BF7F7F00BF7F7F00804040004000000009F5F5F00BF7F7F00BF7F
7F00BF7F7F00BF7F7F00BF7F7F00000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FF00FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01E6E6E62DBABAB87FF3F3F315FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFF
FF01C4C4C2759A9A97FF898986FFB9B9B78BF1F1F11B959591FFB2B2AFFFAAAA
A7C9D6D6D461FFFFFF01FFFFFF01FFFFFF01BF7F9F00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00BFBFBF00404040007F7F7F00EFEFEF0050505000DFDFDF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00BF7F9F000000000000000FF000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000FF0000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01B9B9B7818E8E8BFFB9B9B7FFADADAB97EEEE
ED1FFFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FDFD
FD0380807DFFEDEDECFFF7F7F7FFBFBFBDFF868683FF9D9D99FFF9F9F8FFE7E7
E6FFADADAAFFB3B3B0B5DFDFDE4BFFFFFF01BF7F9F00FFFFFF00FFFFFF00BFBF
BF00404040007F7F7F00FFFFFF00DFDFDF00DFDFDF00EFEFEF0050505000DFDF
DF00FFFFFF00FFFFFF00FFFFFF00BF7F9F000000000000000FF000000FF000000
00007B7B7B007B7B7B007B7B7B007B7B7B007B7B7B007B7B7B007B7B7B007B7B
7B007B7B7B000000FF0000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01B9B9B7818E8E8BFFF2F2F2FFFDFDFDFFBDBDBCFFADAD
AB97EEEEED1FFFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01C2C2
C079B0B0ADFFFDFDFDFFE8D9C8FFFDFDFDFFF4F4F4FFB6B6B4FF838380FFA9B6
BCFFFDFDFDFFDEDEDDFFA8A8A5FFBEBEBC99BF7F9F00BFBFBF00404040007F7F
7F00FFFFFF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00DFDFDF00EFEFEF005050
5000DFDFDF00FFFFFF00FFFFFF00BF7F9F000000000000000000000000FF000000
FF0000FFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFF
FF000000FF0000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01B9B9B7818E8E8BFFF2F2F2FFFDFDFDFFEBEAF5FFFDFDFDFFBDBD
BCFFADADAB97EEEEED1FFFFFFF01FFFFFF01FFFFFF01FFFFFF01FBFBFB078282
7EFFEEEEEEFFDDC5ACFFC5A07BFFC19566FFDBC2A8FFFCFBFBFFEEEEEEFF9B9B
98FF76AED0FFE5F1F9FFFDFDFDFFD3D3D1FF602060007F7F7F00FFFFFF00DFDF
DF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00DFDFDF00EFEF
EF0050505000DFDFDF00FFFFFF00BF7FBF000000000000000000000000000000
FF00FFFFFF00FF000000FF000000FF000000FF000000FF000000FF000000FFFF
FF007B7B7B000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01B9B9B7818E8E8BFFF2F2F2FFF6F6FAFF8B88D0FF7D7ACBFFB5B3E0FFFDFD
FDFFBDBDBCFFADADAB97EEEEED1FFFFFFF01FFFFFF01FFFFFF01C1C1BF7BB3B3
B1FFFDFDFDFFC3996DFFC6A88FFFC6A890FFC5A07CFFD0B08FFFF7F7F7FF8B8B
87FF66BBE4FF56AEE4FFE5F1F9FFCECECCFF7F3F7F009F9F9F00DFDFDF00DFDF
DF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00DFDF
DF00EFEFEF0050505000DFDFDF00BF7FBF000000000000000000000000000000
FF000000FF00FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF000000
FF007B7B7B000000000000000000000000FFFFFF01FFFFFF01FFFFFF01B9B9
B7818E8E8BFFF2F2F2FFF6F6FAFF8D8AD1FF7675D8FF7072ECFF7E7BCDFFB5B3

E0FFFDFDFDFFBDBDBCFFADADAB97FFFFFF01FFFFFF01FBFBFB0782827EFFF0F0
EFFFDBC2A8FFC5A07CFFC6A992FFC6A992FFC5A07CFFDBC2A8FFC3C3C1FF76A2
B3FF6AC9F3FF51AAE2FFF8F8F8FF9D9D99FFBF7FBF00BFBFBF009F9F9F00DFDF
DF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDF
DF00DFDFDF00EFEFEF00505050009F5F9F00000000000000000000000000000000
00000000FF000000FF00FF000000FF000000FF0000000000FF000000FF000000
FF007B7B7B00000000000000000000000000FFFFFF01FFFFFF01B9B9B7818E8E
8BFFF2F2F2FFF6F6FAFF8D8AD1FF7675D8FF6F72EFFF6F72EFFF7072ECFF7E7B
CEFFE3E2F2FFFDFDFDFFA7A7A5FFCDCDCB5DFFFFFF01C1C1BF7BB6B6B4FFFDFD
FCFFC3986BFFC6A890FFC6A992FFC6A88FFFC3996DFFF7F7F7FF888884FF6EC1
E4FF5EB7E9FF9BCEEEFFCCCCCAFFC4C4C28B9F7FDF00FFFFFF00BFBFBF009F9F
9F00DFDFDF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00DFDFDF00BFBFBF00DFDF
DF00FFFFFF00EFEFEF006F6F6F0020005F00000000000000000000000000000000
00000000FF000000FF000000FF00FFFFFF0000FFFF000000FF000000FF00FFFF
FF007B7B7B00000000000000000000000000FFFFFF01B9B9B7818E8E8BFFF2F2
F2FFF6F6FAFF8E8BD1FF7675D8FF6F72EFFF6F72EFFF6F72EFFF7072EBFF7E7A
C9FFFDFDFDFFD4D4D2FF797975FFFBFBFB07FAFAFA09838380FFF0F0EFFFDBC2
A8FFC5A07CFFC6A992FFC6A992FFC5A07CFFDDC5ACFFC0C0BEFF76A2B3FF6AC9
F3FF52ABE3FFF8F8F8FF9B9B98FFF1F1F11F9F7FDF00FFFFFF00FFFFFF00BFBF
BF009F9F9F00DFDFDF00DFDFDF00DFDFDF00BFBFBF00DFDFDF00FFFFFF00FFFF
FF00FFFFFF00BFBFBF00BFBFBF0020005F00000000000000000000000000000000
0000FFFFFF00FFFFFF000000FF000000FF000000FF000000FF0000FFFF00FFFF
FF007B7B7B00000000000000000000000B9B9B7818E8E8BFFF2F2F2FFFDFD
FDFF8E8BD1FF7675D8FF6F72EFFF6F72EFFF6F72EFFF7072EBFF7E7AC9FFF1F1
F8FFD4D4D2FF797975FFD7D7D649FFFFFF01BFBFBD7FB7B7B5FFFDFDFDFFD2B3
92FFC6A07BFFC5A78DFFC6A78FFFC49A6EFFF6F6F5FF888884FF6EC3E7FF5EB7
E9FF9ECFEEFFC9C9C7FFC4C4C28BFFFFFF017F7FFF00FFFFFF00FFFFFF00FFFF
FF00BFBFBF009F9F9F00DFDFDF00DFDFDF00FFFFFF00FFFFFF009FDF9F00FFFF
FF00FFFFFF00FFFFFF003F3F3F003F3FBF0000000000000000000000000000000
0000FFFFFF00FF0000000000FF000000FF000000FF000000FF00000000000000
00007B7B7B00000000000000000000000868682FFF2F2F2FFFDFDFDFFFDFD
FDFF9592D4FF7776D8FF6F72EFFF6F72EFFF7072EBFF7E7AC9FFF1F1F8FFD4D4
D2FF797975FFD7D7D649FFFFFF01FFFFFF01858581FFF1F1F1FFFDFDFDFFFDFD
FDFFE9DACAFFC59D72FFC6A078FFDDC5ACFFBDBDBBFF819FB0FF56AEE4FF57AD
E3FFF6F6F5FF9A9A96FFF4F4F419FFFFFF017F7FFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00BFBFBF009F9F9F00FFFFFF00DF9FDF00BF5F5F009FBF9F00DFDF
DF007F7F7F007F7F7F00FFFFFF007F7FFF0000000000000000000000000000000
000000FFFF000000FF000000FF00FFFFFF000000000000000FF000000FF00FFFF
FF007B7B7B00000000000000000000000969693FFFDFDFDFFE5E5E5FFE0E0
E0FFEEEEEFFFCECDE6FF7776D8FF7072EBFF7E7AC9FFF1F1F8FFD4D4D2FF7979
75FFD7D7D649FFFFFF01FFFFFF01FFFFFF01838380FFF6F6F5FFC2C2C4FFA0A0
A4FFB2B2B5FFFDFDFDFFEDE1D5FFF4F4F4FF858581FFBEBEBEFFE5F1F9FFFDFD
FDFFC7C7C5FFC7C7C585FFFFFF01FFFFFF017F7FFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00BFBFBF009F9F9F00BFBFFF00DFDFDF007F7F7F007F7F
7F00FFFFFF00FFFFFF00FFFFFF007F7FFF0000000000000000000000000000000
00000000FF000000FF000000FF00FFFFFF000000000000000FF000000FF000000
FF0000000000000000000000000000000000969693FFEAEAEBFFB3B3B5FFB7B7
B8FFA8A8AAFFEFEFEFFF9592D4FF7F7BCAFFF1F1F8FFD4D4D2FF797975FFD7D7
D649FFFFFF01FFFFFF01FFFFFF01FFFFFF01ACACA9A5D1D1CFFF97979BFFEDED
EEFFB8B8BBFFD4D4D6FFFDFDFDFFBABAB8FFB6B6B4FFC3C3C3FFF4F4F4FFF6F6
F5FF999995FFF7F7F613FFFFFF01FFFFFF017F7FDF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00BFBFBF006F6F6F007F7F7F00FFFFFF00FFFF

FF00FFFFFF00FFFFFF00FFFFFF007F7FDF000000000000000000000000000000
FF000000FF00FFFFFF0000FFFF00FFFFFF0000000000FFFFFF00000000000000
00000000FF00000000000000000000000000969693FFD0D0D1FFCACACBFFF2F2
F2FFBCBCBDFFDBDBDCFFFDFDFDFFFDFDFDFFD4D4D2FF797975FFD7D7D649FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01D1D1CF5BABABA9FFB0B0B3FFBABA
BCFF9A9A9DFFF2F2F2FFF3F3F2FF858581FFCECECEFFBDBDBDFFFBFBFBFFC4C4
C2FFC7C7C585FFFFFF01FFFFFF01FFFFFF017F7FDF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF007F7FDF00000000000000FF000000FF000000
FF00000000000000000000000000000000000000000000000000000000000000
000000000000000000000000FF0000000000969693FFDADADBFFC1C1C3FFC6C6
C7FFB3B3B5FFE9E9E9FFFDFDFDFFD4D4D2FF797975FFD7D7D649FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01F5F5F513888884FFFCFCFBFFD2D2
D3FFD1D1D1FFB1B1AFFF8B8B87FFA5A5A3FFCCCCCCFFC4C4C3FFB0B0ADFFBFBF
BD97F9F9F90DFFFFFF01FFFFFF01FFFFFF017F7FDF007F7FDF007F7FDF007F7F
DF003F3FDF00BFBFDF007F7FDF00BFBFDF007F7FDF007F7FDF007F7FDF00BFBF
FF007F7FDF000000BF00BFBFFF007F7FDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FF00969693FFFDFDFDFFDADADBFFD5D5
D6FFEBEBEBFFFDFDFDFFD2D2D1FF797975FFD7D7D649FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01B5B5B2939E9E9BFF8686
83FFA3A3A0B7C2C2C07B9F9F9BFFA9A9A6FFA0A09DE1C3C3C18DE2E2E045FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF017F7FBF003F3F9F007F7FBF00BFBF
DF003F3F9F00BFBFDF007F7FBF007F7FBF007F7FBF007F7FBF003F3F9F007F7F
BF00BFBFDF003F3F9F007F7FBF00BFBFDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000ADADAB97767672FF767672FF7676
72FF767672FF767672FFA0A09EAFD9D9D845FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01FFFFFF01DFDFDE3FF7F7
F70FFFFFFF01FFFFFF01F1F1F021E9E9E833FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01FFFFFF010000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000EFEFEF00AFAFAF00A0A0A000CFCFCF000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000DFDFDF00DFDFDF00DFDF
DF00DFDFDF00DFDFDF009F9F9F00306F6F00003F3F00003F3F00007F7F006060
6000DFDFDF00DFDFDF00DFDFDF00DFDFDF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000008080
8000808080008080800080808000808080008080800080808000808080008080
80008080800080808000808080000000000000000000000000009F9F9F008080
8000808080008080800080808000808080008080800080808000808080008080
8000DFDFDF0000000000000000000000000000005F5F5F004040400040404000404040
40004040400040404000003F3F0000BFBF0000BFBF00007F7F0000FFFF00007F

7F00202020004040400040404000A0A0A0000000000000000000000000008080
8000808080008080800080808000808080008080800080808000808080008080
8000808080008080800080808000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000080808000808080000000000003F3FDF000000BF000000
BF000000BF000000BF000000BF000000BF000000BF000000BF000000BF006060
8000808080008DFDFDF0000000000000000000000000003F3F3F003F3F3F007F7F
7F007F7F7F007F7F7F003F7F7F0000BFBF00007F7F00007F7F00007F7F0000BF
BF003F3F3F003F3F3F0000000000A0A0A0000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000080808000808080000000000000000000000BFBF00FFFF
FF0000BFBF008080800000BFBF008080800000BFBF008080800000BFBF008080
800000BFBF00808080000000000008080800007F7FBF0000007F0000007F000000
7F0000007F0000007F0000007F0000007F0000003F003F3F9F007F7F7F003F3F
9F006060800080808000DFDFDF0000000000000000007F7F7F00000000000000
0000000000000000000000000000003F3F0060DFDF0060DFDF0030AFAF00BFBF
BF00000000007F7F7F0000000000A0A0A0000000000000000000BFBF008080
800000BFBF008080800000BFBF008080800000BFBF008080800000BFBF008080
800000BFBF00808080000000000008080800000000000000000000000000000BF
BF00FFFFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF0000000000080808000BFBFDF003F3F7F007F7FBF007F7F
BF007F7FBF007F7FBF007F7FBF007F7FBF007F7F9F00000000007F7F7F007F7F
7F003F3F9F006060800080808000DFDFDF00000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F007F7F7F00003F3F00007F7F007F7F7F007F7F
7F00BFBFBF007F7F7F0000000000A0A0A0000000000000000000FFFFFF0000BF
BF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF0000000000808080000000000000000000808080000000
000000BFBF00FFFFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF0000FFFF0000BFBF0000000000000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F0000000000000000000000007F7F7F007F7F
7F007F7F7F003F3F9F0060608000BFBFBF00000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F00000000000000000007F7FBF00404080007F7F7F007F7F
7F00BFBFBF007F7F7F0000000000A0A0A0000000000000000000BFBF0000FF
FF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF00808080000000000008080800000000000000000008080800080
80000000000000BFBF00FFFFFF0000BFBF00FFFFFF0000BFBF00FFFFFF0000BF
BF00FFFFFF0000BFBF00FFFFFF000000000000000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F007F7F7F00000000000000000007F7F7F007F7F
7F007F7F7F007F7F7F003F3F9F00BFBFBF00000000007F7F7F0000000000FFBF
BF00FF7F7F00FFBFBF0000000000000000007F7FBF00404080000FF7F7F00FF7F
7F00000000007F7F7F0000000000A0A0A0000000000000000000FFFFFF0000BF
BF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF00000000008080800000000000000000008080800080
80008080800000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000007F7F7F00BFBFBF007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF0000000000000000007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF00000000007F7F7F0000000000FFBF
BF00FF000000FFBFBF0000000000000000007F7FBF00404080000FF3F3F00FF7F
7F00000000007F7F7F0000000000A0A0A0000000000000000000BFBF0000FF
FF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF00808080000000000008080800000000000000000008080800080
80008080800080808000808080008080800080808000808080008080800080

```
800080808000808080000000000000000000000007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F
7F007F7F7F007F7F7F007F7F7F00BFBFBF00000000007F7F7F0000000000000
0000FF000000000000000000000000000007F7FBF0040408000FF000000FFBF
BF000000000007F7F7F0000000000A0A0A000000000000000000FFFFFF0000BF
BF00FFFFFF0000BFBF00FFFFFF0000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF0000FFFF0000BFBF000000000080808000000000000000000000000000000
000000000000000000000000000080808000808080008080800080808000808
800080808000000000000000000000000000000000000000007F7F7F007F7F
BF000000BF003F3FDF007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF00000000007F7F7F0000000000FFBF
BF00FF00000000000000BFBFBF00000000003F3F7F0040408000FF7F7F00FF00
0000000000007F7F7F0000000000A0A0A000000000000000000000000000000
000000000000000000000000000FFFFFF0000BFBF0000FFFF0000BFBF0000FF
FF0000BFBF000000000080808000000000000000000000000000000000000000
000000000000000000000000000080808000808080008080800080808000808
800080808000000000000000000000000000000000000000007F7F
7F007F7F7F007F7F7F007F7FBF000000BF003F3FDF007F7F7F007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F00BFBFBF00000000007F7F7F0000000000FFBF
BF00FF00000000000000BFBFBF007F7F7F005F5F7F0060608000FF000000FF3F
3F000000000007F7F7F0000000000A0A0A000000000000000000000000000000
000000000000000000000000000000BFBF0000FFFF0000BFBF0000FFFF0000BF
BF008080800000000000808080000000000000000000000000000000000000000
000000000000000000000000000080808000808080008080800080808000808
800080808000000000000000000000000000000000000000000000000000000
00007F7F7F007F7F7F007F7F7F007F7F7F007F7F7F007F7FBF000000BF003F3F
DF007F7F7F007F7F7F007F7F7F00BFBFBF00000000007F7F7F00BFBFBF000000
000000000000000000000000000000000000000000000000000000000000000
0000BFBFBF007F7F7F0000000000A0A0A000000000000000000000000000000
000000000000000000000000000FFFFFF0000BFBF00FFFFFF0000BFBF00FFFF
FF0000BFBF000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
0000000000007F7F7F007F7F7F003F3FDF000000BF007F7FBF007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000DFDFDF000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
```

```
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000424D3E000000000000003E000000
    2800000040000000300000000100010000000000800100000000000000000000
    0000000000000000000000FFFFFF00FFC7800000000000E000000000000000
    EF04000000000000E804000000000000E804000000000000C004000000000000
    8004000000000000004000000000000000400000000000000040000000000000
    007C000000000000003C00000000000081E0000000000000EFE12A8000000000
    EFE3000000000000E00700000000000008001FFFE000000000000BFFD00000000
    0000900300000000000C00700000000000E00700000000000E00700000000000
    0000F00700000000000F00700000000000F00700000000000F00700000000000
    0000F00700000000000F00F000000000000E01700000000000008FFD00000000
    0000FFFE000000000000FFFF00000000FFFFFFFFFFE1FFFFFFFFFFFFF8000FFFF
    E001C0070000E001C00080030000C000800000013E088000800000400000800 0
    800081C002008000800080C022088000800080C022088000800180003608800 0
    C003C0002408C001FC03E0002408FC01FC03F0001EF0FC03FE07F8010000FE07
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000000000000000000000000000000
    000000000000}
end
object DeleteDocumentQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'delete from case_document'
    'where idcase_document = :casedocid')
  Params = <
    item
      DataType = ftUnknown
      Name = 'casedocid'
      ParamType = ptUnknown
    end>
  Left = 64
  Top = 120
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'casedocid'
      ParamType = ptUnknown
    end>
end
object DeleteCaseQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'delete from case_file'
    'where idcase = :caseid')
  Params = <
    item
      DataType = ftUnknown
      Name = 'caseid'
      ParamType = ptUnknown
    end>
  Left = 160
  Top = 72
```

```
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'caseid'
        ParamType = ptUnknown
      end>
  end
  object DeleteYearQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'delete From year where idyear = :yearid')
    Params = <
      item
        DataType = ftUnknown
        Name = 'yearid'
        ParamType = ptUnknown
      end>
    Left = 184
    Top = 176
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'yearid'
        ParamType = ptUnknown
      end>
  end
  object DeleteUniqMatchTextQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'delete from unique_matched_text where idcase = :case')
    Params = <
      item
        DataType = ftUnknown
        Name = 'case'
        ParamType = ptUnknown
      end>
    Left = 64
    Top = 176
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'case'
        ParamType = ptUnknown
      end>
  end
  object NullDateScannedQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'update case_document'
      'set datetime_last_scan = NULL'
      'where idcase = :case')
    Params = <
```

```
      item
        DataType = ftUnknown
        Name = 'case'
        ParamType = ptUnknown
      end>
    Left = 296
    Top = 176
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'case'
        ParamType = ptUnknown
      end>
  end
end
```

NewYear.pas

```
unit NewYear;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, DB, ZAbstractRODataset, ZAbstractDataset,
ZAbstractTable, ZDataset, StdCtrls, Buttons, Mask, DBCtrls;

type
  TNew_Year_Form = class(TForm)
    DBEdit1:                        TDBEdit;
    Label1:                         TLabel;
    BitBtn1:                        TBitBtn;
    BitBtn2:                        TBitBtn;

    NewYearTable:                   TZTable;
    NewYearDS:                      TDataSource;
    NewYearTableidyear:             TIntegerField;
    NewYearTableyear:               TWideStringField;

    NextYearQuery:                  TZQuery;
    NextYearQuerynextyear:          TFloatField;

    procedure FormClose             (Sender: TObject;
                                     var Action: TCloseAction);
    procedure FormCreate            (Sender: TObject);
    procedure BitBtn2Click          (Sender: TObject);
    procedure FormCloseQuery        (Sender: TObject;
                                     var CanClose: Boolean);
    procedure NewYearTableAfterPost(DataSet: TDataSet);
    procedure NewYearTablePostError(DataSet: TDataSet;
                                     E: EDatabaseError;
                                     var Action: TDataAction);
```

```
    procedure BitBtn1Click         (Sender: TObject);
  private
    function YearValid(S: String):  Boolean;
    { Private declarations }
  protected
    procedure CreateParams(var Params: TCreateParams); override;
  public
    class procedure openNewYear    (ID:   Integer);
    { Public declarations }
  end;

var
  New_Year_Form: TNew_Year_Form;

implementation

uses Thesis_Main;

{$R *.dfm}

procedure TNew_Year_Form.BitBtn1Click(Sender: TObject);
begin
   if YearValid(DBEdit1.Text) then
   begin
      NewYearTable.Post;
      Close;
   end
   else
      MessageDlg('Year Provided is Invalid', mtError, [mbOK], 0);
end;

procedure TNew_Year_Form.BitBtn2Click(Sender: TObject);
begin
   Close;
end;

procedure TNew_Year_Form.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

procedure TNew_Year_Form.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   NewYearTable.Close;
   Action := caFree;
end;

procedure TNew_Year_Form.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
```

```
begin
   if NewYearTable.State = dsInsert then
   begin
      if (MessageDlg('Cancel New Year Operation?',
                     mtConfirmation, [mbYes, mbNo], 0)) = mrYes then
         CanClose := True
      else
         CanClose := False;
   end
   else
      CanClose := True;
end;

procedure TNew_Year_Form.FormCreate(Sender: TObject);
begin
   NewYearTable.Open;
   NewYearTable.Insert;
   NextYearQuery.Open;
   NewYearTableYear.AsString := NextYearQuerynextYear.AsString;
   NextYearQuery.Close;
end;

procedure TNew_Year_Form.NewYearTableAfterPost(DataSet: TDataSet);
begin
   Scanner_Main_Form.AddNewYearNode(NewYearTableidYear.AsInteger,
                                    NewYearTableyear.AsString);
end;

procedure TNew_Year_Form.NewYearTablePostError(DataSet: TDataSet;
                                               E: EDatabaseError;
                                               var Action:
                                               TDataAction);
begin
   MessageDlg('Error Posting Year'+#10#13+'Ensure Year is Unique',
              mtError, [mbOK], 0);
   Action := daAbort;
end;

class procedure TNew_Year_Form.openNewYear(ID: Integer);
var
   F, Child: TNew_Year_Form;
   I:        Integer;
begin
   F := Nil;
   for I := Screen.FormCount - 1 DownTo 0 do
   begin
      if (Screen.Forms[I] is TNew_Year_Form) then
         F := Screen.Forms[I] As TNew_Year_Form;
   end;
   if F = Nil then
   begin
      Child := TNew_Year_Form.Create(Application);
```

```
      Child.Show;
   end
   else
   begin
      if F.WindowState = wsMinimized then
         F.WindowState := wsNormal;
      F.BringToFront;
   end;
end;

function TNew_Year_Form.YearValid(S: String): Boolean;
begin
   if ((Length(S) <> 4) or
       (not(CharInSet(S[1], ['0'..'9']))) or
       (not(CharInSet(S[2], ['0'..'9']))) or
       (not(CharInSet(S[3], ['0'..'9']))) or
       (not(CharInSet(S[4], ['0'..'9']))) or
       (not ((StrToInt(S) > 1950) and (StrToInt(S) < 2100)))
       ) then
      Result := False
   else
      Result := True;
end;

end.
```

NewYear.dfm

```
object New_Year_Form: TNew_Year_Form
  Left = 0
  Top = 0
  ActiveControl = DBEdit1
  BorderIcons = [biSystemMenu]
  Caption = 'Add New Year'
  ClientHeight = 111
  ClientWidth = 429
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  DesignSize = (
    429
    111)
  PixelsPerInch = 96
  TextHeight = 13
```

```
object Label1: TLabel
  Left = 64
  Top = 17
  Width = 22
  Height = 13
  Caption = 'Year'
end
object DBEdit1: TDBEdit
  Left = 64
  Top = 36
  Width = 121
  Height = 21
  DataField = 'year'
  DataSource = NewYearDS
  TabOrder = 0
end
object BitBtn1: TBitBtn
  Left = 280
  Top = 26
  Width = 130
  Height = 25
  Anchors = [akRight, akBottom]
  Caption = 'Create Year'
  Default = True
  DoubleBuffered = True
  Glyph.Data = {
    36030000424D3603000000000000360000002800000010000000100000000100
    1800000000000003000000000000000000000000000000000000FFFFFFFFFFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFCFCFCFA0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0
    A0A0CFCFCFA0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0DFDFDFBFBFBF7F7F7F
    7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F
    7F7F7F7F606060DFDFDF7F7F7FDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF60
    6060DFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF404040DFDFDF7F7F7FBFBFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF606060BFBFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFF404040DFDFDF7F7F7FDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF60
    6060DFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF404040DFDFDF7F7F7FBFBFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF606060BFBFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFF404040DFDFDF7F7F7FDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF60
    6060DFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF404040DFDFDF7F7F7FBFBFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF606060BFBFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFF404040DFDFDF7F7F7FDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF60
    6060DFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDF404040DFDFDF7F7F7FFFBFBF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF606060FFFFFFFFFFFFFFFFFFFFBFBFFF7F
    7FFFBFBF404040DFDFDF7F7F7FFF7F7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF60
    6060FFFFFFFFFFFFFFFFFFFFFFFFFF7F7FFFFFFF404040DFDFDF7F7F7FFF7F7F
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF606060FFFFFFFFFFFFFFFFFFFFBFBFFF7F
    7FFF7F7F404040DFDFDFBFBFBF7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F
    7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF}
```

```
    ParentDoubleBuffered = False
    TabOrder = 1
    OnClick = BitBtn1Click
  end
  object BitBtn2: TBitBtn
    Left = 280
    Top = 57
    Width = 130
    Height = 25
    Anchors = [akRight, akBottom]
    Cancel = True
    Caption = 'Cancel'
    DoubleBuffered = True
    Glyph.Data = {
      BE060000424DBE06000000000000360400002800000024000000120000000100
      0800000000008802000000000000000000000001000000000000000000000000
      80000080000000808000800000008000800080800000C0C0C000C0DCC000F0CA
      A600000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      000000000000000000000000000000000000F0FBFF00A4A0A000808080000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00030303030303
      0303030303030303030303030303030303030303030303030303030303030303
      0303F8F80303030303030303030303030303030303FF03030303030303030303
      0303030303F90101F80303030303F9F80303030303030303F8F8FF0303030303
      03FF03030303030303F9010101F8030303F90101F8030303030303F8FF03F8FF
```

```
      030303FFF8F8FF0303030303F901010101F803F901010101F80303030303F8
      FF0303F8FF03FFF80303F8FF0303030303F901010101F80101010101F80303
      030303F8FF030303F8FFF803030303F8FF0303030303F90101010101010101
      F803030303030303F8FF030303F803030303FFF8030303030303030303F9010101
      010101F80303030303030303F8FF0303030303FFF8030303030303030303
      030101010101F80303030303030303030303F8FF0303030303F8030303030303
      0303030303F901010101F80303030303030303030303F8FF030303F8030303
      030303030303030303F90101010101F80303030303030303030303F803030303
      F8FF030303030303030303F9010101F8010101F80303030303030303030303F803
      03030303F8FF0303030303030303F9010101F803F9010101F803030303030303
      03F8030303F8FF0303F8FF03030303030303F90101F8030303F9010101F80303
      03030303F8FF0303F803F8FF0303F8FF03030303030303F9010303030303F901
      0101030303030303F8FFFFF8030303F8FF0303F8FF030303030303030303030303
      030303F901F903030303030303F8F80303030303F8FFFFFFF803030303030303
      0303030303030303030303030303030303030303030303030303F8F8F803030303
      0303030303030303030303030303030303030303030303030303030303030303
      0303}
    NumGlyphs = 2
    ParentDoubleBuffered = False
    TabOrder = 2
    OnClick = BitBtn2Click
  end
  object NewYearTable: TZTable
    Connection = Scanner_Main_Form.ZConnection1
    AfterPost = NewYearTableAfterPost
    OnPostError = NewYearTablePostError
    TableName = 'year'
    Left = 208
    Top = 64
    object NewYearTableidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object NewYearTableyear: TWideStringField
      FieldName = 'year'
      Required = True
      Size = 4
    end
  end
  object NewYearDS: TDataSource
    DataSet = NewYearTable
    Left = 136
    Top = 64
  end
  object NextYearQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select max(year)+1 as nextyear'
      'from year')
    Params = <>
    Left = 40
    Top = 64
```

```
    object NextYearQuerynextyear: TFloatField
      FieldName = 'nextyear'
      ReadOnly = True
    end
  end
end
```

NewCase.pas

```
unit NewCase;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, StdCtrls, Buttons, ComCtrls,
adpDBDateTimePicker, Mask, DBCtrls, Masks, ZDataset, AbstractRODataset,
ZAbstractDataset, ZAbstractTable;

type
  TNew_Case_Form = class(TForm)
    DBEdit1:                      TDBEdit;
    adpDBDateTimePicker1:         TadpDBDateTimePicker;
    Label1:                       TLabel;
    Label2:                       TLabel;
    BitBtn1:                      TBitBtn;
    BitBtn2:                      TBitBtn;

    CaseTableDS:                  TDataSource;
    CaseTable:                    TZTable;
    CaseTableidcase:              TIntegerField;
    CaseTableidyear:              TIntegerField;
    CaseTablecase_number:         TWideStringField;
    CaseTablecase_start_date:     TDateField;
    CaseTablecase_close_date:     TDateField;

    yearQuery:                    TZQuery;
    yearQueryyear:                TWideStringField;
    yearQueryidyear:              TIntegerField;

    NextCaseNoQuery:              TZQuery;
    NextCaseNoQuerynext_case_no: TFloatField;

    procedure BitBtn2Click       (Sender: TObject);
    procedure BitBtn1Click       (Sender: TObject);
    procedure FormCreate         (Sender: TObject);
    procedure CaseTablePostError (DataSet: TDataSet; E: EDatabaseError;
                                  var Action: TDataAction);
    procedure CaseTableAfterPost (DataSet: TDataSet);
    procedure FormClose          (Sender: TObject;
                                  var Action: TCloseAction);
    procedure FormCloseQuery     (Sender: TObject;
```

```
                                    var CanClose: Boolean);
    procedure DBEdit1KeyPress    (Sender: TObject; var Key: Char);
  private
    function CaseNumberValid     (S: String):  Boolean;
    { Private declarations }
  protected
    procedure CreateParams       (var Params: TCreateParams); override;
  public
    class procedure openNewCase  (ID:  Integer);
    { Public declarations }
  end;

var
  New_Case_Form: TNew_Case_Form;

implementation

uses Thesis_Main;

{$R *.dfm}

procedure TNew_Case_Form.BitBtn1Click(Sender: TObject);
begin
   if CaseNumberValid(DBEdit1.Text) then
   begin
      CaseTable.Post;
      Close;
   end
   else
      MessageDlg('Case Number Provided is Invalid', mtError,
                 [mbOK], 0);
end;

procedure TNew_Case_Form.BitBtn2Click(Sender: TObject);
begin
   Close;
end;

function TNew_Case_Form.CaseNumberValid(S: String): Boolean;
begin
   If MatchesMask(S, '[1-2][09][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9]')
then
      Result := True
   else
      Result := False;
end;

procedure TNew_Case_Form.CaseTableAfterPost(DataSet: TDataSet);
begin
   Scanner_Main_Form.AddNewCaseNode(yearQueryyear.AsString,
                                    CaseTableidcase.AsInteger,
                                    CaseTablecase_number.AsString);
```

```
end;

procedure TNew_Case_Form.CaseTablePostError(DataSet: TDataSet;
                                            E: EDatabaseError;
                                            var Action: TDataAction);
begin
   MessageDlg('Error Posting Case' + #10#13 + 'Ensure Case Number is
Unique',
              mtError, [mbOK], 0);
   Action := daAbort;
end;

procedure TNew_Case_Form.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

procedure TNew_Case_Form.DBEdit1KeyPress(Sender: TObject; var Key:
Char);
begin
   if (CharInSet(Key, ['-'])) then
   begin
      if (pos('-',DBEdit1.text) <> 0) then
         Key := #0
   end
   else if Not (CharInSet(Key,['0'..'9',#8])) then
      Key := #0;
end;

procedure TNew_Case_Form.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   CaseTable.Close;
   YearQuery.Close;
   Action := caFree;
end;

procedure TNew_Case_Form.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
   if CaseTable.State = dsInsert then
   begin
      if (MessageDlg('Cancel New Case Operation?',
                     mtConfirmation, [mbYes, mbNo],
                    0)) = mrYes then
         CanClose := True
      else
         CanClose := False;
   end
   else
```

```
      CanClose := True;
end;

procedure TNew_Case_Form.FormCreate(Sender: TObject);
begin
   YearQuery.ParamByName('idyr').Value :=

integer(Scanner_Main_Form.TreeView1.Selected.Data);
   YearQuery.Open;
   CaseTable.Open;
   NextCaseNoQuery.ParamByName('inyear').Value := integer(

Scanner_Main_Form.TreeView1.Selected.Data);
   NextCaseNoQuery.Open;
   CaseTable.Insert;
   CaseTableidyear.Value :=
integer(Scanner_Main_Form.TreeView1.Selected.Data);
   CaseTableCase_Number.AsString := YearQueryYear.AsString + '-' +

NextCaseNoQueryNext_Case_No.AsString;
   NextCaseNoQuery.Close;
   CaseTableCase_Start_Date.AsDateTime := Date;
   DBEdit1.SelStart := Length(DBEdit1.Text);
   DBEdit1.SelLength := 0;
end;

class procedure TNew_Case_Form.openNewCase(ID: Integer);
var
   F: TNew_Case_Form;
   Child: TNew_Case_Form;
   I: Integer;
begin
   F := Nil;
   for I := Screen.FormCount - 1 DownTo 0 do
   begin
      if (Screen.Forms[I] is TNew_Case_Form) then
      begin
         F := Screen.Forms[I] As TNew_Case_Form;
         if F.yearQueryidyear.AsInteger <> ID then
            F := Nil
         else
            Break;
      end;
   end;
   if F = Nil then
   begin
      Child := TNew_Case_Form.Create(Application);
      Child.Show;
   end
   else
   begin
      if F.WindowState = wsMinimized then
```

```
         F.WindowState := wsNormal;
      F.BringToFront;
   end;
end;

end.
```

NewCase.dfm

```
object New_Case_Form: TNew_Case_Form
  Left = 0
  Top = 0
  ActiveControl = DBEdit1
  BorderIcons = [biSystemMenu]
  Caption = 'Add New Case'
  ClientHeight = 144
  ClientWidth = 424
  Color = clBtnFace
  Constraints.MinHeight = 180
  Constraints.MinWidth = 440
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  DesignSize = (
    424
    144)
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 16
    Top = 5
    Width = 64
    Height = 13
    Caption = 'Case Number'
    FocusControl = DBEdit1
  end
  object Label2: TLabel
    Left = 25
    Top = 64
    Width = 77
    Height = 13
    Caption = 'Case Start Date'
  end
  object DBEdit1: TDBEdit
    Left = 16
```

```
    Top = 24
    Width = 134
    Height = 21
    AutoSelect = False
    DataField = 'case_number'
    DataSource = CaseTableDS
    TabOrder = 0
    OnKeyPress = DBEdit1KeyPress
  end
  object adpDBDateTimePicker1: TadpDBDateTimePicker
    Left = 24
    Top = 80
    Width = 185
    Height = 21
    Date = 40767.008367291670000000
    Time = 40767.008367291670000000
    TabOrder = 1
    DataField = 'case_start_date'
    DataSource = CaseTableDS
    ReadOnly = False
  end
  object BitBtn1: TBitBtn
    Left = 286
    Top = 21
    Width = 130
    Height = 25
    Anchors = [akRight, akBottom]
    Caption = 'Create Case'
    Default = True
    DoubleBuffered = True
    Glyph.Data = {
      F6000000424DF600000000000000760000002800000010000000100000000100
      0400000000008000000000000000000000000100000010000000000000000000
      BF0000BF000000BFBF00BF000000BF00BF00BFBF0000C0C0C000808080000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00777777777777
      7777777777777777777778888888888887770000000000088703838383838
      380870F3B3B3B3B3B308703B3B3B3B3B380870F3B3B3B3B3B308703B3B3B3B3B
      380870F3F3F3B3B3B3087700000F3B3B308777777703B3B380877777770F3F3F
      307777777700000077777777777777777777777777777777}
    ParentDoubleBuffered = False
    TabOrder = 2
    OnClick = BitBtn1Click
  end
  object BitBtn2: TBitBtn
    Left = 286
    Top = 59
    Width = 130
    Height = 25
    Anchors = [akRight, akBottom]
    Cancel = True
    Caption = 'Cancel'
    DoubleBuffered = True
```

```
Glyph.Data = {
  BE060000424DBE06000000000000360400002800000024000000120000000100
  0800000000008802000000000000000000000000000100000000000000000000
  80000080000000808000800000008000800080800000C0C0C000C0DCC000F0CA
  A600000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000F0FBFF00A4A0A000808080000000
  FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00030303030303
  0303030303030303030303030303030303030303030303030303030303030303
  0303F8F80303030303030303030303030303030303FF03030303030303030303
  0303030303F90101F80303030303F9F80303030303030303F8F8FF0303030303
  03FF03030303030303F9010101F8030303F90101F8030303030303F8FF03F8FF
  030303FFF8F8FF030303030303F901010101F803F901010101F80303030303F8
  FF0303F8FF03FFF80303F8FF030303030303F901010101F80101010101F80303
  030303F8FF030303F8FFF803030303F8FF030303030303F90101010101010101
  F80303030303030303F8FF030303F803030303FFF80303030303030303F9010101
  010101F8030303030303030303F8FF030303030303FFF8030303030303030303
  030101010101F80303030303030303030303F8FF0303030303F8030303030303
  0303030303F901010101F8030303030303030303030303F8FF030303F8030303
  0303030303030303F90101010101F8030303030303030303030303F803030303
  F8FF030303030303030303F9010101F8010101F803030303030303030303F803
  03030303F8FF0303030303030303F9010101F803F9010101F803030303030303
  03F8030303F8FF0303F8FF03030303030303F90101F8030303F9010101F80303
  03030303F8FF0303F803F8FF0303F8FF03030303030303F9010303030303F901
  0101030303030303F8FFFFF8030303F8FF0303F8FF0303030303030303030303
```

```
      030303F901F9030303030303F8F80303030303F8FFFFFFF803030303030303
      0303030303030303030303030303030303030303030303F8F8F803030303
      0303030303030303030303030303030303030303030303030303030303030303
      0303}
    NumGlyphs = 2
    ParentDoubleBuffered = False
    TabOrder = 3
    OnClick = BitBtn2Click
  end
  object CaseTableDS: TDataSource
    DataSet = CaseTable
    Left = 24
    Top = 112
  end
  object CaseTable: TZTable
    Connection = Scanner_Main_Form.ZConnection1
    AfterPost = CaseTableAfterPost
    OnPostError = CaseTablePostError
    TableName = 'case_file'
    Left = 64
    Top = 112
    object CaseTableidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object CaseTableidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object CaseTablecase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
    object CaseTablecase_start_date: TDateField
      FieldName = 'case_start_date'
      Required = True
    end
    object CaseTablecase_close_date: TDateField
      FieldName = 'case_close_date'
    end
  end
  object yearQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select '
      '   idyear,'
      '   year'
      'from '
      '   year'
      'where'
      '   idyear = :idyr')
```

```
    Params = <
      item
        DataType = ftUnknown
        Name = 'idyr'
        ParamType = ptUnknown
      end>
    Left = 152
    Top = 104
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'idyr'
        ParamType = ptUnknown
      end>
    object yearQueryyear: TWideStringField
      FieldName = 'year'
      Required = True
      Size = 4
    end
    object yearQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
  end
  object NextCaseNoQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select'
      '   max(substr(case_number,6,5))+1 as next_case_no'
      'from '
      '   case_file '
      'where '
      '   idyear = :inyear')
    Params = <
      item
        DataType = ftUnknown
        Name = 'inyear'
        ParamType = ptUnknown
      end>
    Left = 248
    Top = 96
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'inyear'
        ParamType = ptUnknown
      end>
    object NextCaseNoQuerynext_case_no: TFloatField
      FieldName = 'next_case_no'
      ReadOnly = True
    end
  end
```

```
end
```

NewDocument.pas

```
unit NewDocument;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ZAbstractRODataset, ZAbstractDataset,
ZDataset, ZAbstractTable, Mask, StdCtrls, Buttons, DBCtrls, RVScroll,
RichView, SHFolder, RVEdit, DBRV, RVStyle, ShlObj, RVOfficeCnv;

type
  TNew_Doc_Form = class(TForm)
    Label1:                                  TLabel;
    Label2:                                  TLabel;
    DBEdit1:                                 TDBEdit;
    BitBtn1:                                 TBitBtn;
    BitBtn2:                                 TBitBtn;
    oc:                                      TRVOfficeConverter;
    od:                                      TOpenDialog;
    CheckBox1:                               TCheckBox;
    DBLookupComboBox1:                       TDBLookupComboBox;

    RVStyle1:                                TRVStyle;
    DBRichViewEdit1:                         TDBRichViewEdit;

    YearQuery:                               TZQuery;
    YearQueryyear:                           TWideStringField;

    CaseQuery:                               TZQuery;
    CaseQuerycase_number:                    TWideStringField;

    NewDocumentQueryDS:                      TDataSource;
    NewDocumentQuery:                        TZQuery;
    NewDocumentQueryidcase_document:         TIntegerField;
    NewDocumentQueryidyear:                  TIntegerField;
    NewDocumentQueryidcase:                  TIntegerField;
    NewDocumentQueryiddocument_type:         TIntegerField;
    NewDocumentQuerydocument_name:           TWideStringField;
    NewDocumentQuerydatetime_created:        TDateTimeField;
    NewDocumentQuerydatetime_last_accessed:  TDateTimeField;
    NewDocumentQuerydatetime_last_modified:  TDateTimeField;
    NewDocumentQuerydatetime_last_scan:      TDateTimeField;

    DocumentTypeQueryDS:                     TDataSource;
    DocumentTypeQuery:                       TZQuery;
    DocumentTypeQueryiddocument_type:        TIntegerField;
    DocumentTypeQuerydocument_type_name:     TWideStringField;
```

```
    DocumentQueryDS:                          TDataSource;
    DocumentQuery:                            TZQuery;
    DocumentQueryidcase_document:             TIntegerField;
    DocumentQueryidyear:                      TIntegerField;
    DocumentQueryidcase:                      TIntegerField;
    DocumentQuerydocument_text:               TBlobField;

    procedure FormCreate                     (Sender: TObject);
    procedure FormClose                      (Sender: TObject;
                                              var Action: TCloseAction);
    procedure BitBtn1Click                   (Sender: TObject);
    procedure NewDocumentQueryAfterPost      (DataSet: TDataSet);
    procedure BitBtn2Click                   (Sender: TObject);
    procedure FormCloseQuery                 (Sender: TObject;
                                              var CanClose: Boolean);
    procedure NewDocumentQueryBeforePost     (DataSet: TDataSet);
    procedure NewDocumentQueryPostError      (DataSet: TDataSet;
                                              E: EDatabaseError;
                                              var Action: TDataAction);
  private
    procedure ImportRVDoc                    (YID, CID, CDID: Integer);
    function GetSpecialFolderPath            (Folder: Integer):  string;
    { Private declarations }
  protected
    procedure CreateParams                   (var Params:
TCreateParams);override;
  public
    { Public declarations }
  end;

var
  New_Doc_Form: TNew_Doc_Form;

implementation

uses Thesis_Main, Document;

{$R *.dfm}

procedure TNew_Doc_Form.BitBtn1Click(Sender: TObject);
var
   Hold_DocumentName:  String;
begin
   Hold_DocumentName := Trim(NewDocumentQueryDocument_Name.AsString);
   if Hold_DocumentName <> '' then
   begin
      NewDocumentQueryDocument_Name.AsString := Hold_DocumentName;
      if DBLookupComboBox1.Text <> '' then
      begin
         NewDocumentQuery.Post;
         Close;
      end
```

```
      else
         MessageDlg('You Must Select a Document Type!',
                    mtError, [mbOK], 0);
   end
   else
      MessageDlg('Document Name Cannot Be Blank', mtError, [mbOK], 0);
end;

procedure TNew_Doc_Form.BitBtn2Click(Sender: TObject);
begin
   Close;
end;

procedure TNew_Doc_Form.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

procedure TNew_Doc_Form.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   NewDocumentQuery.Close;
   DocumentTypeQuery.Close;
   Action := caFree;
end;

procedure TNew_Doc_Form.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
   if NewDocumentQuery.State = dsInsert then
   begin
      if (MessageDlg('Cancel New Document Operation?',
                     mtConfirmation, [mbYes, mbNo], 0)) = mrYes then
         CanClose := True
      else
         CanClose := False;
   end
   else
      CanClose := True;
end;

procedure TNew_Doc_Form.FormCreate(Sender: TObject);
begin
   NewDocumentQuery.Open;
   DocumentTypeQuery.Open;
   NewDocumentQuery.Insert;
   NewDocumentQueryidcase.AsInteger := integer(

Scanner_Main_Form.TreeView1.Selected.Data);
   NewDocumentQueryidYear.AsInteger := integer(
```

```
Scanner_Main_Form.TreeView1.Selected.Parent.Data);
   YearQuery.ParamByName('yr').Value := integer(

Scanner_Main_Form.TreeView1.Selected.Parent.Data);
   CaseQuery.ParamByName('casefl').Value := integer(

Scanner_Main_Form.TreeView1.Selected.Data);
   CaseQuery.Open;
   Caption := Caption + ' - ' + CaseQueryCase_Number.AsString;
   CaseQuery.Close;
end;

function TNew_Doc_Form.GetSpecialFolderPath(Folder:  integer):  string;
const
   SHGFP_TYPE_CURRENT = 0;
var
   Path: Array[0..MAX_PATH] of char;
begin
   if SUCCEEDED(SHGetFolderPath(0,folder,0,
                SHGFP_TYPE_CURRENT,@path[0])) then
      Result := path
   else
      Result := '';
end;

procedure TNew_Doc_Form.ImportRVDoc(YID, CID, CDID: Integer);
var
   ChildForm:  TCase_Doc_Form;
begin
   oc.ExtensionsInFilter := True;
   od.Filter := 'Rich Text Format Document|*.rtf|'+oc.GetImportFilter;
   od.InitialDir := GetSpecialFolderPath(CSIDL_PERSONAL)+
                                          '\Thesis_Docs\Word_Documents';
   if od.Execute then
   begin
      DocumentQuery.ParamByName('csdoc').Value := CDID;
      DocumentQuery.ParamByName('idyr').Value := YID;
      DocumentQuery.ParamByName('idcs').Value := CID;
      DocumentQuery.Open;
      DocumentQuery.Edit;
      DBRichViewEdit1.Clear;
      DBRichViewEdit1.Format;
      DBRichViewEdit1.Update;
      DBRichViewEdit1.Clear;
      Screen.Cursor := crSQLWait;
      try
         if od.FilterIndex = 1 then
            DBRichViewEdit1.LoadRTF(od.FileName)
         else
            oc.ImportRV(od.FileName, DBRichViewEdit1,od.FilterIndex-2);
      finally
```

```
         Screen.Cursor := crDefault;
      end;
      DBRichViewEdit1.Format;
      DBRichViewEdit1.Change;
      DBRichViewEdit1.Modified := True;
      DocumentQuery.Post;
   end;
   ChildForm := TCase_Doc_Form.CreateWithID(Application, CDID);
   ChildForm.Show;
end;

procedure TNew_Doc_Form.NewDocumentQueryAfterPost(DataSet: TDataSet);
var
   DocYear,
   DocCase,
   NodeText:  String;
   NewDocID:  Integer;
begin
   YearQuery.Open;
   DocYear := YearQueryyear.AsString;
   YearQuery.Close;
   CaseQuery.Open;
   DocCase := CaseQueryCase_Number.AsString;
   CaseQuery.Close;
   NewDocID := NewDocumentQueryidcase_document.AsInteger;
   NodeText := NewDocumentQueryDocument_Name.AsString + ' - '+
               DBLookupComboBox1.Text;
   Scanner_Main_Form.AddNewDocumentNode(DocYear, DocCase, NewDocID,
NodeText);
   if CheckBox1.Checked then
      ImportRVDoc(NewDocumentQueryidyear.AsInteger,
                  NewDocumentQueryidCase.AsInteger,
                  NewDocumentQueryidcase_Document.AsInteger);
end;

procedure TNew_Doc_Form.NewDocumentQueryBeforePost(DataSet: TDataSet);
begin
   NewDocumentQuerydatetime_created.AsDateTime := Now;
   NewDocumentQuerydatetime_last_accessed.AsDateTime := Now;
   NewDocumentQuerydatetime_last_modified.AsDateTime := Now;
end;

procedure TNew_Doc_Form.NewDocumentQueryPostError(DataSet: TDataSet;
                                                  E: EDatabaseError;
                                                  var Action:
                                                  TDataAction);
begin
   MessageDlg('Error Posting Document'+#10#13+
              'Ensure that Document Type and Document Name'+#10#13+
              'is Unique Within a Case', mtError, [mbOK],0);
   Action := daAbort;
end;
```

```
end.
```

NewDocument.dfm

```
object New_Doc_Form: TNew_Doc_Form
  Left = 0
  Top = 0
  ActiveControl = DBEdit1
  BorderIcons = [biSystemMenu]
  Caption = 'Add New Document'
  ClientHeight = 134
  ClientWidth = 648
  Color = clBtnFace
  Constraints.MinHeight = 170
  Constraints.MinWidth = 530
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  DesignSize = (
    648
    134)
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 80
    Top = 21
    Width = 78
    Height = 13
    AutoSize = False
    Caption = 'Document Name'
  end
  object Label2: TLabel
    Left = 80
    Top = 67
    Width = 75
    Height = 13
    AutoSize = False
    Caption = 'Document Type'
  end
  object BitBtn1: TBitBtn
    Left = 500
    Top = 38
    Width = 130
    Height = 25
```

```
  Anchors = [akRight, akBottom]
  Caption = 'Create Document'
  Default = True
  DoubleBuffered = True
  Glyph.Data = {
    16020000424D160200000000000076000000280000001A0000001A0000000100
    040000000000A001000000000000000000001000000000000000000000000000
    BF0000BF000000BFBF00BF000000BF00BF00BFBF0000C0C0C000808080000000
    FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00777777777777
    777777777777770A000477777777777777777777777777000000777777777777
    777777777777700000077777788888888888887777770000007777778888888
    888888887777770000007777744444444444448877777700000077774BFBFBF
    BFBFB4887777770000007777 4FBFBFBFBFBF4887777770000007777 4BFBFBF
    BFBFB4887777770707077777 4FBFBFBFBFBF4887777770000007777 4BFBFBF
    BFBFB4887777770707077777 4FBFBFBFBFBF4887777770000007777 4BFBFBF
    BFBFB4887777770707077777 4FBFBFBFBFBF4887777770000007777 4BFBFBF
    BFBFB4887777770707077777 4FBFBFBFBFBF4887777770000007777 4BFBFBF
    B444447777777707070777F77FFBFBFBF4FB47777777770000007777F7FBFBFBF
    B4B47777777777040404777 7FFFBFBFBF4477777777777000000777FFFFFFF444
    44777777777777FFFBFF7777FFF77777777777777777777000000777F7F7F7777
    77777777777777FBFFFB77777F77F77777777777777777777000007777777777
    77777777777777FFFBFF77777777777777777777777777000000}
  ParentDoubleBuffered = False
  TabOrder = 3
  OnClick = BitBtn1Click
end
object DBEdit1: TDBEdit
  Left = 80
  Top = 40
  Width = 385
  Height = 21
  Anchors = [akLeft, akTop, akRight]
  DataField = 'document_name'
  DataSource = NewDocumentQueryDS
  TabOrder = 0
end
object DBLookupComboBox1: TDBLookupComboBox
  Left = 80
  Top = 84
  Width = 329
  Height = 21
  Anchors = [akLeft, akTop, akRight]
  DataField = 'iddocument_type'
  DataSource = NewDocumentQueryDS
  KeyField = 'iddocument_type'
  ListField = 'document_type_name'
  ListSource = DocumentTypeQueryDS
  TabOrder = 1
end
object BitBtn2: TBitBtn
  Left = 500
  Top = 76
```

```
Width = 130
Height = 25
Anchors = [akRight, akBottom]
Cancel = True
Caption = 'Cancel'
DoubleBuffered = True
Glyph.Data = {
```


```
  BE060000424DBE06000000000000360400002800000024000000120000000100
  0800000000008802000000000000000000000000010000000000000000000000
  80000080000000808000800000008000800080800000C0C0C000C0DCC000F0CA
  A600000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000F0FBFF00A4A0A000808080000000
  FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00030303030303
  0303030303030303030303030303030303030303030303030303030303030303
  0303F8F80303030303030303030303030303030303FF03030303030303030303
  0303030303F90101F80303030303F9F803030303030303F8F8FF030303030303
  03FF0303030303030303F9010101F8030303F90101F803030303030303F8FF03F8FF
  030303FFF8F8FF030303030303F901010101F803F901010101F80303030303F8
  FF0303F8FF03FFF80303F8FF030303030303F901010101F80101010101F80303
  030303F8FF030303F8FFF803030303F8FF030303030303F90101010101010101
  F8030303030303F8FF030303F803030303FFF8030303030303030303F9010101
  010101F8030303030303030303F8FF030303030303FFF8030303030303030303
  030101010101F8030303030303030303030303F8FF0303030303F8030303030303
  0303030303F901010101F803030303030303030303030303F8FF030303F8030303
```

```
      0303030303030303F90101010101F80303030303030303030303F803030303
      F8FF030303030303030303F9010101F8010101F80303030303030303030303F803
      03030303F8FF030303030303030303F9010101F803F9010101F803030303030303
      03F8030303F8FF0303F8FF030303030303F90101F8030303F9010101F80303
      03030303F8FF0303F803F8FF0303F8FF030303030303030F9010303030303F901
      0101030303030303F8FFFFF8030303F8FF0303F8FF030303030303030303030303
      030303F901F903030303030303F8F80303030303F8FFFFFFF803030303030303
      0303030303030303030303030303030303030303030303030303F8F8F803030303
      0303030303030303030303030303030303030303030303030303030303030303
      0303}
    NumGlyphs = 2
    ParentDoubleBuffered = False
    TabOrder = 4
    OnClick = BitBtn2Click
  end
  object DBRichViewEdit1: TDBRichViewEdit
    Left = 128
    Top = 127
    Width = 100
    Height = 52
    DataField = 'document_text'
    DataSource = DocumentQueryDS
    ReadOnly = False
    TabOrder = 5
    Visible = False
    DoInPaletteMode = rvpaCreateCopies
    RTFReadProperties.TextStyleMode = rvrsAddIfNeeded
    RTFReadProperties.ParaStyleMode = rvrsAddIfNeeded
    RVFOptions = [rvfoSavePicturesBody, rvfoSaveControlsBody,
rvfoSaveBinary, rvfoSaveTextStyles, rvfoSaveParaStyles,
rvfoSaveDocProperties, rvfoLoadDocProperties]
    Style = RVStyle1
  end
  object CheckBox1: TCheckBox
    Left = 8
    Top = 112
    Width = 81
    Height = 17
    Anchors = [akLeft, akBottom]
    Caption = 'Import Now'
    Checked = True
    State = cbChecked
    TabOrder = 2
  end
  object NewDocumentQueryDS: TDataSource
    DataSet = NewDocumentQuery
    Left = 568
    Top = 104
  end
  object DocumentTypeQueryDS: TDataSource
    DataSet = DocumentTypeQuery
    Left = 480
```

```
    Top = 104
  end
  object NewDocumentQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    BeforePost = NewDocumentQueryBeforePost
    AfterPost = NewDocumentQueryAfterPost
    OnPostError = NewDocumentQueryPostError
    SQL.Strings = (
      'select '
      '   idcase_document, '
      '   idyear, '
      '   idcase, '
      '   iddocument_type, '
      '   document_name,'
      '   datetime_created,'
      '   datetime_last_accessed,'
      '   datetime_last_modified,'
      '   datetime_last_scan'
      'from '
      '   case_document')
    Params = <>
    Left = 608
    Top = 104
    object NewDocumentQueryidcase_document: TIntegerField
      FieldName = 'idcase_document'
      Required = True
    end
    object NewDocumentQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object NewDocumentQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object NewDocumentQueryiddocument_type: TIntegerField
      FieldName = 'iddocument_type'
      Required = True
    end
    object NewDocumentQuerydocument_name: TWideStringField
      FieldName = 'document_name'
      Required = True
      Size = 45
    end
    object NewDocumentQuerydatetime_created: TDateTimeField
      FieldName = 'datetime_created'
      Required = True
    end
    object NewDocumentQuerydatetime_last_accessed: TDateTimeField
      FieldName = 'datetime_last_accessed'
      Required = True
    end
```

```
    object NewDocumentQuerydatetime_last_modified: TDateTimeField
      FieldName = 'datetime_last_modified'
      Required = True
    end
    object NewDocumentQuerydatetime_last_scan: TDateTimeField
      FieldName = 'datetime_last_scan'
    end
  end
  object DocumentTypeQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select * from document_type'
      'order by document_type_name')
    Params = <>
    Left = 520
    Top = 104
    object DocumentTypeQueryiddocument_type: TIntegerField
      FieldName = 'iddocument_type'
      Required = True
    end
    object DocumentTypeQuerydocument_type_name: TWideStringField
      FieldName = 'document_type_name'
      Required = True
      Size = 45
    end
  end
  object YearQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select year from year where idyear = :yr')
    Params = <
      item
        DataType = ftUnknown
        Name = 'yr'
        ParamType = ptUnknown
      end>
    Left = 16
    Top = 8
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'yr'
        ParamType = ptUnknown
      end>
    object YearQueryyear: TWideStringField
      FieldName = 'year'
      Required = True
      Size = 4
    end
  end
  object CaseQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
```

```
    SQL.Strings = (
      'select case_number from case_file'
      'where idcase = :casefl')
    Params = <
      item
        DataType = ftUnknown
        Name = 'casefl'
        ParamType = ptUnknown
      end>
    Left = 16
    Top = 56
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'casefl'
        ParamType = ptUnknown
      end>
    object CaseQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
  end
  object RVStyle1: TRVStyle
    TextStyles = <
      item
        StyleName = 'Normal text'
        FontName = 'Arial'
        Unicode = True
      end
      item
        StyleName = 'Heading'
        FontName = 'Arial'
        Style = [fsBold]
        Color = clBlue
        Unicode = True
      end
      item
        StyleName = 'Subheading'
        FontName = 'Arial'
        Style = [fsBold]
        Color = clNavy
        Unicode = True
      end
      item
        StyleName = 'Keywords'
        FontName = 'Arial'
        Style = [fsItalic]
        Color = clMaroon
        Unicode = True
      end
      item
```

```
      StyleName = 'Jump 1'
      FontName = 'Arial'
      Style = [fsUnderline]
      Color = clGreen
      Jump = True
      Unicode = True
    end
    item
      StyleName = 'Jump 2'
      FontName = 'Arial'
      Style = [fsUnderline]
      Color = clGreen
      Jump = True
      Unicode = True
    end>
  ParaStyles = <
    item
      StyleName = 'Paragraph Style'
      Tabs = <>
    end
    item
      StyleName = 'Centered'
      Alignment = rvaCenter
      Tabs = <>
    end>
  ListStyles = <>
  InvalidPicture.Data = {
```

```
    07544269746D617036100000424D361000000000000036000000280000002000
    0000200000000100200000000000000010000000000000000000000000000000
    0000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
```

```
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF000000FF000000FF00FFFF
FF00FFFFFF000000FF000000FF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF000000FF000000
FF000000FF000000FF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF000000
FF000000FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF000000FF000000
FF000000FF000000FF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF000000FF000000FF00FFFF
FF00FFFFFF000000FF000000FF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
```

```
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF008080800080808000808080008080800080808000808080008080
800080808000808080008080800080808000808080008080800080808000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
8000808080008080800080808000808080008080800080808000808080008080
8000808080008080800080808000808080008080800080808000808080008080
8000808080008080800080808000808080008080800080808000808080008080
```

```
    800080808000808080008080800080808000808080008080800080808000808080008080
    8000}
  StyleTemplates = <>
  Left = 272
  Top = 132
end
object oc: TRVOfficeConverter
  Left = 328
  Top = 120
end
object DocumentQueryDS: TDataSource
  DataSet = DocumentQuery
  Left = 240
  Top = 120
end
object DocumentQuery: TZQuery
  Connection = Scanner_Main_Form.ZConnection1
  SQL.Strings = (
    'select'
    '   idcase_document,'
    '   idyear, '
    '   idcase,'
    '   document_text'
    'from case_document'
    'where'
    '   idcase_document = :csdoc and'
    '   idyear = :idyr and'
    '   idcase = :idcs')
  Params = <
    item
      DataType = ftUnknown
      Name = 'csdoc'
      ParamType = ptUnknown
    end
    item
      DataType = ftUnknown
      Name = 'idyr'
      ParamType = ptUnknown
    end
    item
      DataType = ftUnknown
      Name = 'idcs'
      ParamType = ptUnknown
    end>
  Left = 288
  Top = 120
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'csdoc'
      ParamType = ptUnknown
    end
```

```
      item
        DataType = ftUnknown
        Name = 'idyr'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'idcs'
        ParamType = ptUnknown
      end>
    object DocumentQueryidcase_document: TIntegerField
      FieldName = 'idcase_document'
      Required = True
    end
    object DocumentQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object DocumentQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object DocumentQuerydocument_text: TBlobField
      FieldName = 'document_text'
    end
  end
  object od: TOpenDialog
    Left = 48
    Top = 128
  end
end
```

CaseFile.pas

```
unit CaseFile;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ZAbstractRODataset, ZAbstractDataset,
ZDataset, StdCtrls, Mask, DBCtrls, Buttons, ComCtrls,
adpDBDateTimePicker;

type
  TCaseFile_Form = class(TForm)
    DBEdit1:                          TDBEdit;
    Label1:                           TLabel;
    Label2:                           TLabel;
    Label3:                           TLabel;
```

```
    BitBtn1:                          TBitBtn;
    BitBtn2:                          TBitBtn;
    BitBtn3:                          TBitBtn;
    adpDBDateTimePicker1:             TadpDBDateTimePicker;
    adpDBDateTimePicker2:             TadpDBDateTimePicker;

    CaseDS:                           TDataSource;
    CaseQuery:                        TZQuery;
    CaseQueryidcase:                  TIntegerField;
    CaseQueryidyear:                  TIntegerField;
    CaseQuerycase_number:             TWideStringField;
    CaseQuerycase_start_date:         TDateField;
    CaseQuerycase_close_date:         TDateField;

    procedure FormCreate             (Sender: TObject);
    procedure FormClose              (Sender: TObject;
                                      var Action: TCloseAction);
    procedure BitBtn1Click           (Sender: TObject);
    procedure BitBtn2Click           (Sender: TObject);
    procedure CaseDSStateChange      (Sender: TObject);
    procedure BitBtn3Click           (Sender: TObject);
    procedure FormCloseQuery         (Sender: TObject;
                                      var CanClose: Boolean);
    procedure CaseQueryAfterPost     (DataSet: TDataSet);
  private
    CaseFileID:                       Integer;
    { Private declarations }
  protected
    procedure CreateParams           (var Params: TCreateParams);
override;
  public
    constructor CreateWithID         (AOwner: TApplication;
                                      CaseID:  Integer);
    class function CaseInEditMode:    Boolean;
    class procedure OpenCaseForm     (ID:  Integer);
    { Public declarations }
  end;

var
   CaseFile_Form: TCaseFile_Form;

implementation

{$R *.dfm}

uses Thesis_Main;

procedure TCaseFile_Form.BitBtn1Click(Sender: TObject);
begin
   CaseQuery.Post;
end;
```

```
procedure TCaseFile_Form.BitBtn2Click(Sender: TObject);
begin
   caseQuery.Cancel;
end;

procedure TCaseFile_Form.BitBtn3Click(Sender: TObject);
begin
   Close;
end;

procedure TCaseFile_Form.CaseDSStateChange(Sender: TObject);
begin
   BitBtn1.Enabled := CaseQuery.State = dsEdit;
   BitBtn2.Enabled := CaseQuery.State = dsEdit;
end;

class function TCaseFile_Form.CaseInEditMode: Boolean;
var
   F: TCaseFile_Form;
   I: Integer;
begin
   F := nil;
   for i := Screen.FormCount - 1 DownTo 0 do
   begin
      if (Screen.Forms[i] is TCaseFile_Form) then
      begin
         F := Screen.Forms[I] As TCaseFile_Form;
         if F.CaseQuery.State <> dsEdit then
            F := nil
         else
            break;
      end;
   end;
   if F = nil then
      Result := False
   else
   begin
      if F.WindowState = wsMinimized then
         F.WindowState := wsNormal;
      F.BringToFront;
      Result := True;
   end;
end;

procedure TCaseFile_Form.CaseQueryAfterPost(DataSet: TDataSet);
begin
   if CaseQuerycase_close_date.AsString = '' then
      Scanner_Main_Form.BoldCaseNodes(
                                  CaseQuerycase_number.AsString, True)
   else
      Scanner_Main_Form.BoldCaseNodes(
                                 CaseQuerycase_number.AsString, False)
```

```
end;

procedure TCaseFile_Form.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

constructor TCaseFile_Form.CreateWithID(AOwner: TApplication; CaseID:
Integer);
begin
   Inherited Create(AOwner);
   CaseFileID := CaseID;
end;

procedure TCaseFile_Form.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   CaseQuery.Close;
   Action := caFree;
end;

procedure TCaseFile_Form.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
   if CaseQuery.State <> dsEdit then
      CanClose := True
   else
   begin
      MessageDlg('Please Complete or Cancel Edits Before Closing',
                 mtWarning, [mbOK],0);
      CanClose := False;
   end;
end;

procedure TCaseFile_Form.FormCreate(Sender: TObject);
begin
   adpDBDateTimePicker1.Date := Date;
   adpDBDateTimePicker2.Date := Date;
   CaseQuery.ParamByName('cs').Value := CaseFileID;
   CaseQuery.Open;
   Caption := Caption + ' ' + CaseQuerycase_number.AsString;
end;

class procedure TCaseFile_Form.OpenCaseForm(ID: Integer);
var
   F, Child: TCaseFile_Form;
   I:        Integer;
begin
   F := Nil;
   for I := Screen.FormCount - 1 DownTo 0 do
```

```
   begin
      if (Screen.Forms[I] is TCaseFile_Form) then
      begin
         F := Screen.Forms[I] As TCaseFile_Form;
         if F.CaseQueryidcase.AsInteger <> ID then
            F := Nil
         else
            Break;
      end;
   end;
   if F = Nil then
   begin
      Child := TCaseFile_Form.CreateWithID(Application, ID);
      Child.Show;
   end
   else
   begin
      if F.WindowState = wsMinimized then
         F.WindowState := wsNormal;
      F.BringToFront;
   end;
end;

end.
```

CaseFile.dfm

```
object CaseFile_Form: TCaseFile_Form
  Left = 0
  Top = 0
  ActiveControl = DBEdit1
  Caption = 'Case File'
  ClientHeight = 204
  ClientWidth = 534
  Color = clBtnFace
  Constraints.MinHeight = 240
  Constraints.MinWidth = 550
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 24
    Top = 16
```

```
    Width = 64
    Height = 13
    Caption = 'Case Number'
    FocusControl = DBEdit1
  end
  object Label2: TLabel
    Left = 57
    Top = 80
    Width = 77
    Height = 13
    Caption = 'Case Start Date'
  end
  object Label3: TLabel
    Left = 288
    Top = 80
    Width = 85
    Height = 13
    Caption = 'Case Closed Date'
  end
  object DBEdit1: TDBEdit
    Left = 24
    Top = 32
    Width = 134
    Height = 21
    DataField = 'case_number'
    DataSource = CaseDS
    TabOrder = 0
  end
  object adpDBDateTimePicker1: TadpDBDateTimePicker
    Left = 56
    Top = 96
    Width = 185
    Height = 21
    Date = 40767.008367291670000000
    Time = 40767.008367291670000000
    TabOrder = 1
    DataField = 'case_start_date'
    DataSource = CaseDS
    ReadOnly = False
  end
  object adpDBDateTimePicker2: TadpDBDateTimePicker
    Left = 288
    Top = 99
    Width = 185
    Height = 21
    Date = 40767.008451319450000000
    Time = 40767.008451319450000000
    TabOrder = 2
    DataField = 'case_close_date'
    DataSource = CaseDS
    ReadOnly = False
  end
```

```
object BitBtn1: TBitBtn
  Left = 345
  Top = 152
  Width = 75
  Height = 25
  Caption = 'Save'
  Default = True
  DoubleBuffered = True
  Glyph.Data = {
    66010000424D660100000000000076000000280000001400000014000000100
    040000000000F000000000000000000000000100000001000000000000000000
    800000800000008080008000000080008000808000008080800C0C0C0000000
    FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00877777777777
    77777777FF660000000000000000000007FFDD0877777777777777707FFFF0888
    888888888888707FF770888998888888888807FFFF088000000000000008807
    FFBB088888888888888888807FF770880000000000000008807FFDD0F8888888888
    88888807FFFF0FFFFFFFFFFFFFFFFF07FF770000000000000000008FFDF8888
    8888888888888888FFFF888888888EE788888888FFFF88888888EEEE78888888
    FFFF8888888EEEEEE7888888000888888EEEEEEEE78888822228888888888EE7
    88888888AAAA888888888EE78888888888888888888888EE78888888800008888
    88888EE8888888880000}
  ParentDoubleBuffered = False
  TabOrder = 3
  OnClick = BitBtn1Click
end
object BitBtn2: TBitBtn
  Left = 256
  Top = 152
  Width = 75
  Height = 25
  Caption = 'Cancel'
  DoubleBuffered = True
  Glyph.Data = {
    BE060000424DBE0600000000000036040000280000002400000012000000100
    0800000000008802000000000000000000000001000000000000000000000000
    800000800000008080008000000080008000808000000C0C0C000C0DCC000F0CA
    A60000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
```

```
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000F0FBFF00A4A0A000808080000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00030303030303
      0303030303030303030303030303030303030303030303030303030303030303
      0303F8F8030303030303030303030303030303FF0303030303030303030303
      0303030303F90101F80303030303F9F8030303030303030303F8F8FF0303030303
      03FF030303030303F9010101F8030303F90101F80303030303030303F8FF03F8FF
      030303FFF8F8FF030303030303F901010101F803F901010101F80303030303F8
      FF0303F8FF03FFF80303F8FF030303030303F901010101F80101010101F80303
      030303F8FF030303F8FFF803030303F8FF030303030303F90101010101010101
      F8030303030303030303F8FF030303F803030303FFF8030303030303030303F9010101
      010101F80303030303030303030303F8FF030303030303FFF8030303030303030303
      0301010101F80303030303030303030303F8FF0303030303F8030303030303
      0303030303F901010101F80303030303030303030303030303F8FF030303F8030303
      0303030303030303F90101010101F80303030303030303030303030303F803030303
      F8FF030303030303030303F9010101F8010101F80303030303030303030303F803
      03030303F8FF030303030303030303F9010101F803F9010101F803030303030303
      03F8030303F8FF0303F8FF030303030303F90101F8030303F9010101F80303
      03030303F8FF0303F803F8FF0303F8FF03030303030303F9010303030303F901
      0101030303030303F8FFFFF8030303F8FF0303F8FF030303030303030303030303
      030303F901F9030303030303F8F80303030303F8FFFFFFF803030303030303
      0303030303030303030303030303030303030303030303F8F8F803030303
      0303030303030303030303030303030303030303030303030303030303030303
      0303}
  NumGlyphs = 2
  ParentDoubleBuffered = False
  TabOrder = 4
  OnClick = BitBtn2Click
end
object BitBtn3: TBitBtn
  Left = 432
  Top = 152
  Width = 75
  Height = 25
  Cancel = True
  Caption = 'Close'
  DoubleBuffered = True
  Glyph.Data = {
    4E010000424D4E0100000000000076000000280000001200000012000000100
    040000000000D8000000000000000000000001000000010000000000000000000
```

```
      8000008000000080800080000000800080008080000080808000C0C0C0000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00888888888888
      8888880000008888800888888888880000008888 0FF00888888880000008880
      F00FF00888888800000880F0FF00FF00888880000080F0F78FF00FF0888800
      0000880F70078FF00888880000080F70FB0078FF088880000008800FBFBF007
      088888000000880FBFBFBFB00888880000080FBFBFBFBFBF088880000008800
      BFBFBFBF088888000000888800FBFBF0880888000000888888800B80880008800
      0000888888880088000008000000888888888888880888000000888888888888
      880888000000888888888888888888000000}
    ParentDoubleBuffered = False
    TabOrder = 5
    OnClick = BitBtn3Click
  end
  object CaseDS: TDataSource
    DataSet = CaseQuery
    OnStateChange = CaseDSStateChange
    Left = 352
    Top = 8
  end
  object CaseQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    AfterPost = CaseQueryAfterPost
    SQL.Strings = (
      'Select * from case_file'
      'where idcase = :cs')
    Params = <
      item
        DataType = ftUnknown
        Name = 'cs'
        ParamType = ptUnknown
      end>
    Left = 400
    Top = 8
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'cs'
        ParamType = ptUnknown
      end>
    object CaseQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object CaseQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object CaseQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
```

```
    object CaseQuerycase_start_date: TDateField
      FieldName = 'case_start_date'
      Required = True
    end
    object CaseQuerycase_close_date: TDateField
      FieldName = 'case_close_date'
    end
  end
end
```

Document.pas

```
unit Document;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ZAbstractRODataset, ZAbstractDataset,
ZDataset, RVScroll, RichView, RVEdit, DBRV, RVStyle, RichViewActions,
ActnList, ImgList, Menus, RVOfficeCnv, PtblRV, ExtCtrls, Ruler,
RVRuler, ComCtrls, DBActns,ToolWin, RVTable, CRVData,
RVadlFontComboBox1, RVadlFontSizeComboBox1, RVWordEnum, RVFontCombos,
CRVFData, ShellAPI, RVTypes, StdCtrls, RVLinear, SHFolder, RVMisc,
MarkSearch;

type
   TWordCounter = class(TRVWordEnumerator)
   private
      fCounter:                          Integer;
   protected
      function ProcessWord:              Boolean; override;
   public
      function GetWordCount(rve: TCustomRichViewEdit): Integer;
   end;

  TCase_Doc_Form = class(TForm)
    Edit1:                                TEdit;
    StatusBar1:                           TStatusBar;
    CoolBar1:                             TCoolBar;
    ToolBar1:                             TToolBar;
    ToolBar2:                             TToolBar;
    ToolBar3:                             TToolBar;
    ToolButton1:                          TToolButton;
    ToolButton2:                          TToolButton;
    ToolButton3:                          TToolButton;
    ToolButton4:                          TToolButton;
    ToolButton5:                          TToolButton;
    ToolButton6:                          TToolButton;
    ToolButton7:                          TToolButton;
    ToolButton8:                          TToolButton;
    ToolButton9:                          TToolButton;
```

```
    ToolButton10:                   TToolButton;
    ToolButton11:                   TToolButton;
    ToolButton12:                   TToolButton;
    ToolButton13:                   TToolButton;
    ToolButton14:                   TToolButton;
    ToolButton15:                   TToolButton;
    ToolButton16:                   TToolButton;
    ToolButton17:                   TToolButton;
    ToolButton18:                   TToolButton;
    ToolButton19:                   TToolButton;
    ToolButton20:                   TToolButton;
    ToolButton21:                   TToolButton;
    ToolButton22:                   TToolButton;
    ToolButton23:                   TToolButton;
    ToolButton24:                   TToolButton;
    ToolButton25:                   TToolButton;
    ToolButton26:                   TToolButton;
    ToolButton27:                   TToolButton;
    ToolButton28:                   TToolButton;
    ToolButton29:                   TToolButton;
    ToolButton30:                   TToolButton;
    ToolButton31:                   TToolButton;
    ToolButton32:                   TToolButton;
    ToolButton33:                   TToolButton;
    ToolButton34:                   TToolButton;
    ToolButton35:                   TToolButton;
    ToolButton36:                   TToolButton;
    ToolButton37:                   TToolButton;
    ToolButton38:                   TToolButton;
    ToolButton39:                   TToolButton;
    ToolButton40:                   TToolButton;
    ToolButton41:                   TToolButton;
    ToolButton42:                   TToolButton;
    ToolButton43:                   TToolButton;
    ToolButton44:                   TToolButton;
    ToolButton45:                   TToolButton;
    ToolButton46:                   TToolButton;
    ToolButton47:                   TToolButton;
    ToolButton48:                   TToolButton;
    ToolButton49:                   TToolButton;
    ToolButton50:                   TToolButton;
    ToolButton51:                   TToolButton;
    ToolButton52:                   TToolButton;
    ToolButton53:                   TToolButton;
    ToolButton54:                   TToolButton;
    ToolButton55:                   TToolButton;
    ToolButton56:                   TToolButton;
    ToolButton57:                   TToolButton;
    ToolButton58:                   TToolButton;
    ToolButton59:                   TToolButton;
    ToolButton60:                   TToolButton;
    ToolButton61:                   TToolButton;
```

```
    ToolButton62:                          TToolButton;
    ToolButton63:                          TToolButton;
    ToolButton64:                          TToolButton;
    ToolButton65:                          TToolButton;
    ToolButton66:                          TToolButton;
    ToolButton67:                          TToolButton;
    ToolButton68:                          TToolButton;
    ToolButton69:                          TToolButton;
    ToolButton70:                          TToolButton;
    ToolButton71:                          TToolButton;
    ToolButton72:                          TToolButton;
    ToolButton73:                          TToolButton;
    ToolButton74:                          TToolButton;
    ToolButton75:                          TToolButton;
    ToolButton76:                          TToolButton;
    ToolButton77:                          TToolButton;
    ToolButton78:                          TToolButton;
    ToolButton79:                          TToolButton;
    ToolButton80:                          TToolButton;
    ToolButton82:                          TToolButton;
    ToolButton81:                          TToolButton;
    ToolButton83:                          TToolButton;
    ToolButton84:                          TToolButton;
    ToolButton85:                          TToolButton;
    ToolButton86:                          TToolButton;
    ToolButton87:                          TToolButton;
    ToolButton88:                          TToolButton;
    ImageList1:                            TImageList;
    ActionList1:                           TActionList;
    rd:                                    TReplaceDialog;

    DocumentDS:                            TDataSource;
    DocumentQuery:                         TZQuery;
    DocumentQueryidcase_document:          TIntegerField;
    DocumentQueryidyear:                   TIntegerField;
    DocumentQueryidcase:                   TIntegerField;
    DocumentQueryiddocument_type:          TIntegerField;
    DocumentQuerydocument_name:            TWideStringField;
    DocumentQuerydocument_text:            TBlobField;
    DocumentQuerydatetime_last_scan:       TDateTimeField;
    DocumentQuerydatetime_last_modified:   TDateTimeField;
    DocumentQuerydatetime_created:         TDateTimeField;
    DocumentQuerydatetime_last_accessed:   TDateTimeField;

    RVStyle1:                              TRVStyle;
    DBRichViewEdit1:                       TDBRichViewEdit;
    RVPrint1:                              TRVPrint;
    RVAPopupMenu1:                         TRVAPopupMenu;
    RVAControlPanel1:                      TRVAControlPanel;
    RVOfficeConverter1:                    TRVOfficeConverter;
    RVRuler1:                              TRVRuler;
    RVRuler2:                              TRVRuler;
```

```
    RVRulerItemSelector1:                TRVRulerItemSelector;
    RVFontComboBox1:                     TRVadlFontComboBox;
    RVFontSizeComboBox1:                 TRVFontSizeComboBox1;

    rvActionEvent1:                      TrvActionEvent;
    rvActionNew1:                        TrvActionNew;
    rvActionOpen1:                       TrvActionOpen;
    rvActionSave1:                       TrvActionSave;
    rvActionExport1:                     TrvActionExport;
    rvActionSaveAs1:                     TrvActionSaveAs;
    rvActionPrintPreview1:               TrvActionPrintPreview;
    rvActionPrint1:                      TrvActionPrint;
    rvActionQuickPrint1:                 TrvActionQuickPrint;
    rvActionPageSetup1:                  TrvActionPageSetup;
    rvActionCut1:                        TrvActionCut;
    rvActionCopy1:                       TrvActionCopy;
    rvActionPaste1:                      TrvActionPaste;
    rvActionPasteAsText1:                TrvActionPasteAsText;
    rvActionPasteSpecial1:               TrvActionPasteSpecial;
    rvActionUndo1:                       TrvActionUndo;
    rvActionRedo1:                       TrvActionRedo;
    rvActionSelectAll1:                  TrvActionSelectAll;
    rvActionFind1:                       TrvActionFind;
    rvActionFindNext1:                   TrvActionFindNext;
    rvActionReplace1:                    TrvActionReplace;
    rvActionCharCase1:                   TrvActionCharCase;
    rvActionFonts1:                      TrvActionFonts;
    rvActionFontEx1:                     TrvActionFontEx;
    rvActionFontBold1:                   TrvActionFontBold;
    rvActionFontItalic1:                 TrvActionFontItalic;
    rvActionFontUnderline1:              TrvActionFontUnderline;
    rvActionFontStrikeout1:              TrvActionFontStrikeout;
    rvActionFontGrow1:                   TrvActionFontGrow;
    rvActionFontShrink1:                 TrvActionFontShrink;
    rvActionFontGrowOnePoint1:           TrvActionFontGrowOnePoint;
    rvActionFontShrinkOnePoint1:         TrvActionFontShrinkOnePoint;
    rvActionFontAllCaps1:                TrvActionFontAllCaps;
    rvActionFontOverline1:               TrvActionFontOverline;
    rvActionFontColor1:                  TrvActionFontColor;
    rvActionFontBackColor1:              TrvActionFontBackColor;
    rvActionSubscript1:                  TrvActionSubscript;
    rvActionSuperscript1:                TrvActionSuperscript;
    rvActionParagraph1:                  TrvActionParagraph;
    rvActionParaBorder1:                 TrvActionParaBorder;
    rvActionWordWrap1:                   TrvActionWordWrap;
    rvActionAlignLeft1:                  TrvActionAlignLeft;
    rvActionAlignRight1:                 TrvActionAlignRight;
    rvActionAlignCenter1:                TrvActionAlignCenter;
    rvActionAlignJustify1:               TrvActionAlignJustify;
    rvActionIndentInc1:                  TrvActionIndentInc;
    rvActionIndentDec1:                  TrvActionIndentDec;
    rvActionParaColor1:                  TrvActionParaColor;
```

```
    rvActionLineSpacing1001:              TrvActionLineSpacing100;
    rvActionLineSpacing1501:              TrvActionLineSpacing150;
    rvActionLineSpacing2001:              TrvActionLineSpacing200;
    rvActionClearLeft1:                   TrvActionClearLeft;
    rvActionClearRight1:                  TrvActionClearRight;
    rvActionClearBoth1:                   TrvActionClearBoth;
    rvActionClearNone1:                   TrvActionClearNone;
    rvActionParaList1:                    TrvActionParaList;
    rvActionParaBullets1:                 TrvActionParaBullets;
    rvActionParaNumbering1:               TrvActionParaNumbering;
    rvActionTextRTL1:                     TrvActionTextRTL;
    rvActionTextLTR1:                     TrvActionTextLTR;
    rvActionParaRTL1:                     TrvActionParaRTL;
    rvActionParaLTR1:                     TrvActionParaLTR;
    rvActionInsertFile1:                  TrvActionInsertFile;
    rvActionInsertPicture1:               TrvActionInsertPicture;
    rvActionInsertHLine1:                 TrvActionInsertHLine;
    rvActionInsertHyperlink1:             TrvActionInsertHyperlink;
    rvActionInsertSymbol1:                TrvActionInsertSymbol;
    rvActionInsertText1:                  TrvActionInsertText;
    rvActionColor1:                       TrvActionColor;
    rvActionBackground1:                  TrvActionBackground;
    rvActionFillColor1:                   TrvActionFillColor;
    rvActionInsertPageBreak1:             TrvActionInsertPageBreak;
    rvActionRemovePageBreak1:             TrvActionRemovePageBreak;
    rvActionRemoveHyperlinks1:            TrvActionRemoveHyperlinks;
    rvActionItemProperties1:              TrvActionItemProperties;
    rvActionVAlign1:                      TrvActionVAlign;
    rvActionShowSpecialCharacters1:     TrvActionShowSpecialCharacters;
    rvActionInsertTable1:                 TrvActionInsertTable;
    rvActionTableInsertRowsBelow1:      TrvActionTableInsertRowsBelow;
    rvActionTableInsertRowsAbove1:      TrvActionTableInsertRowsAbove;
    rvActionTableInsertColLeft1:          TrvActionTableInsertColLeft;
    rvActionTableInsertColRight1:         TrvActionTableInsertColRight;
    rvActionTableDeleteRows1:             TrvActionTableDeleteRows;
    rvActionTableDeleteCols1:             TrvActionTableDeleteCols;
    rvActionTableDeleteTable1:            TrvActionTableDeleteTable;
    rvActionTableMergeCells1:             TrvActionTableMergeCells;
    rvActionTableSplitCells1:             TrvActionTableSplitCells;
    rvActionTableSelectTable1:            TrvActionTableSelectTable;
    rvActionTableSelectRows1:             TrvActionTableSelectRows;
    rvActionTableSelectCols1:             TrvActionTableSelectCols;
    rvActionTableSelectCell1:             TrvActionTableSelectCell;
    rvActionTableCellVAlignTop1:          TrvActionTableCellVAlignTop;
    rvActionTableCellVAlignMiddle1:     TrvActionTableCellVAlignMiddle;
    rvActionTableCellVAlignBottom1:     TrvActionTableCellVAlignBottom;
    rvActionTableCellVAlignDefault1:   TrvActionTableCellVAlignDefault;
    rvActionTableCellLeftBorder1:         TrvActionTableCellLeftBorder;
    rvActionTableCellRightBorder1:      TrvActionTableCellRightBorder;
    rvActionTableCellTopBorder1:          TrvActionTableCellTopBorder;
    rvActionTableCellBottomBorder1:    TrvActionTableCellBottomBorder;
    rvActionTableCellAllBorders1:         TrvActionTableCellAllBorders;
```

```
    rvActionTableCellNoBorders1:          TrvActionTableCellNoBorders;
    rvActionTableProperties1:             TrvActionTableProperties;
    rvActionTableGrid1:                   TrvActionTableGrid;
    DataSetPost1:                         TDataSetPost;
    DataSetCancel1:                       TDataSetCancel;

    procedure FormClose                  (Sender: TObject;
                                          var Action: TCloseAction);
    procedure FormCreate                 (Sender: TObject);
    procedure FormCloseQuery             (Sender: TObject;
                                          var CanClose: Boolean);
    procedure DBRichViewEdit1CurTextStyleChanged(Sender: TObject);
    procedure DBRichViewEdit1Select      (Sender: TObject);
    procedure DBRichViewEdit1StyleConversion(
                                          Sender: TCustomRichViewEdit;
                                          StyleNo,UserData: Integer;
                                          AppliedToText: Boolean;
                                          var NewStyleNo: Integer);
    procedure RVFontComboBox1Change      (Sender: TObject);
    procedure RVFontSizeComboBox1Change  (Sender: TObject);
    procedure DocumentDSStateChange      (Sender: TObject);
    procedure DocumentQueryBeforeCancel  (DataSet: TDataSet);
    procedure DBRichViewEdit1CaretMove   (Sender: TObject);
    procedure DBRichViewEdit1Change      (Sender: TObject);
    procedure StatusBar1DblClick         (Sender: TObject);
    procedure DBRichViewEdit1KeyPress    (Sender: TObject;
                                          var Key: Char);
    procedure DBRichViewEdit1KeyDown     (Sender: TObject;
                                          var Key: Word;
                                          Shift: TShiftState);
    procedure DocumentQueryAfterOpen     (DataSet: TDataSet);
    procedure DocumentQueryBeforePost    (DataSet: TDataSet);
    procedure DBRichViewEdit1Jump        (Sender: TObject;
                                          id: Integer);
    procedure ToolButton83Click          (Sender: TObject);
    procedure ToolButton88Click          (Sender: TObject);
    procedure Edit1KeyPress              (Sender: TObject;
                                          var Key: Char);
    procedure DBRichViewEdit1ReadHyperlink(Sender: TCustomRichView;
                                          const Target, Extras:
                                          string; DocFormat:
                                          TRVLoadFormat; var StyleNo,
                                          ItemTag: Integer;
                                          var ItemName:
                                          TRVRawByteString);
    procedure DBRichViewEdit1WriteHyperlink(Sender: TCustomRichView;
                                          id: Integer;
                                          RVData: TCustomRVData;
                                          ItemNo: Integer;
                                          SaveFormat: TRVSaveFormat;
                                          var Target, Extras: string);
    procedure DocumentQueryBeforeEdit    (DataSet: TDataSet);
```

```
  private
    DocumentID:                          Integer;
    FontSize:                            Integer;
    Curr_Line:                           Integer;
    Curr_Col:                            Integer;
    IgnoreChanges:                       Boolean;
    IgnoreNextChar:                      Boolean;
    InsertMode:                          Boolean;
    Op_Cancelled:                        Boolean;
    Date_Editing:                        Boolean;
    JustOpened:                          Boolean;
    FontName:                            String;
    procedure RemoveAllPageBreaks       (RVData:TCustomRVData;
                                         RVE:TCustomRichViewEdit);
    function GetSpecialFolderPath       (Folder: integer): string;
    { Private declarations }
  protected
override;
    { Protected declarations }
  public
    constructor CreateWithID            (AOwner:TApplication;
                                         DocID:Integer);
    class procedure OpenDocForm         (ID: Integer);
    class function DocInEditMode:        Boolean;
    { Public declarations }
  end;

var
   Case_Doc_Form: TCase_Doc_Form;

implementation

{$R *.dfm}

uses Thesis_Main;

const
  TEXT_APPLYFONTNAME  = 4;
  TEXT_APPLYFONTSIZE  = 6;

constructor
TCase_Doc_Form.CreateWithID(AOwner:TApplication;DocID:Integer);
begin
   inherited Create(AOwner);
   DocumentID := DocID;
end;

class procedure TCase_Doc_Form.OpenDocForm(ID:  Integer);
var
   F, Child: TCase_Doc_Form;
   I:        Integer;
begin
```

```
   F := Nil;
   for I := Screen.FormCount - 1 DownTo 0 do
   begin
      if (Screen.Forms[I] is TCase_Doc_Form) then
      begin
         F := Screen.Forms[I] As TCase_Doc_Form;
         if F.DocumentQueryidcase_document.AsInteger <> ID then
            F := Nil
         else
            Break;
      end;
   end;
   if F = Nil then
   begin
      Child := TCase_Doc_Form.CreateWithID(nil, ID);
      Child.Show;
   end
   else
   begin
      if F.WindowState = wsMinimized then
         F.WindowState := wsNormal;
      F.BringToFront;
   end;
end;

procedure TCase_Doc_Form.DBRichViewEdit1CaretMove(Sender: TObject);
begin
   DBRichViewEdit1.GetCurrentLineCol(Curr_Line, Curr_Col);
   StatusBar1.Panels[2].Text := 'Line: ' + IntToStr(Curr_Line);
   StatusBar1.Panels[3].Text := 'Col: ' + IntToStr(Curr_Col);
end;

procedure TCase_Doc_Form.DBRichViewEdit1Change(Sender: TObject);

   function GetParagraphCount(RVData: TCustomRVData): Integer;
   var
      I, R, C: Integer;
      Table:   TRVTableItemInfo;
   begin
      Result := 0;
      for I := 0 to RVData.ItemCount - 1 do
      begin
         if RVData.IsParaStart(I) then
            Inc(Result);
         if RVData.GetItemStyle(I) = rvsTable then
         begin
            Table := TRVTableItemInfo(RVData.GetItem((I)));
            for R := 0 to Table.RowCount - 1 do
               for C := 0 to Table.ColCount - 1 do
                  if table.Cells[R,C] <> nil then
                     Inc(Result, GetParagraphCount(
                         Table.Cells[R,C].GetRVData));
```

```
      end;
   end;
end;

function GetLineCount(RVData: TCustomRVFormattedData):  Integer;
var
   I, R, C: Integer;
   Table:   TRVTableItemInfo;
begin
   Result := 0;
   for I := 0 to RVData.DrawItems.Count - 1 do
   begin
      if RVData.DrawItems[I].FromNewLine then
         Inc(Result);
      if RVData.GetItemStyle(
                      RVData.DrawItems[I].ItemNo) = rvsTable then
      begin
         Table := TRVTableItemInfo(RVData.GetItem(
                  RVdata.DrawItems[i].ItemNo));
         for R := 0 to Table.RowCount - 1 do
            for C := 0 to Table.ColCount - 1 do
               if Table.Cells[R,C] <> nil then
                  Inc(Result, GetLineCount(TCustomRVFormattedData(
                      Table.Cells[R,C].GetRVData)));
      end;
   end;
end;

function GetCharCount(RVData: TCustomRVData): Integer;
var
   I, R, C: Integer;
   Table:   TRVTableItemInfo;
begin
   Result := 0;
   for I := 0 to RVData.Items.Count-1 do
   begin
      if RVData.GetItemStyle(i) >= 0 then
         Inc(Result, RVData.ItemLength(i))
       else if RVData.GetItemStyle(I) = rvsTab then
          Inc(Result)
       else if RVData.GetItemStyle(I) = rvsTable then
       begin
          Table := TRVTableItemInfo(RVData.GetItem(I));
          for R := 0 to Table.Rows.Count-1 do
             for C := 0 to Table.Rows[r].Count-1 do
                if Table.Cells[R,C] <> nil then
                   Inc(Result, GetCharCount(
                       Table.Cells[R,C].GetRVData));
       end;
   end;
end;
```

```
   function GetDBRVEWordCount(RVE: TCustomRichViewEdit): Integer;
   var
      WordCount: TWordCounter;
   begin
      WordCount := TWordCounter.Create;
      try
         Result := WordCount.GetWordCount(RVE);
      finally
         WordCount.Free;
      end;
   end;

   function FormatNumber(N:  Integer): String;
   begin
      Result := Format('%1.0n',[1.0 * N]);
   end;

begin
   StatusBar1.Panels[5].Text := 'Lines:  '+
                                FormatNumber(GetLineCount(
                                DBRichViewEdit1.RVData));
   StatusBar1.Panels[6].Text := 'Paras:  '+
                                FormatNumber(
                                GetParagraphCount(
                                DBRichViewEdit1.RVData));
   StatusBar1.Panels[7].Text := 'Words:  '+
                                FormatNumber(GetDBRVEWordCount(
                                DBRichViewEdit1));
   StatusBar1.Panels[8].Text := 'Chars:  '+FormatNumber(GetCharCount(

DBRichViewEdit1.RVData));
end;

procedure TCase_Doc_Form.DBRichViewEdit1CurTextStyleChanged(
                                                      Sender: TObject);
var
   fi:  TFontInfo;
begin
   IgnoreChanges := True;
   fi := RVStyle1.TextStyles[DBRichViewEdit1.CurTextStyleNo];
   RVFontComboBox1.ItemIndex :=
                           RVFontComboBox1.Items.IndexOf(fi.FontName);
   RVFontSizeComboBox1.Text := IntToStr(fi.Size);
   IgnoreChanges := False;
end;


procedure TCase_Doc_Form.DBRichViewEdit1Jump(Sender: TObject;
                                             id: Integer);
var
   URL:  String;
   RVData:  TCustomRVFormattedData;
```

```
   ItemNo:  Integer;
begin
   DBRichViewEdit1.GetJumpPointLocation(id, RVData, ItemNo);
   URL := PChar(RVData.GetItemTag(ItemNo));
   ShellExecute(0, 'open', PChar(URL), nil, nil, SW_SHOW);
end;

procedure TCase_Doc_Form.DBRichViewEdit1KeyDown(Sender: TObject;
                                                var Key: Word;
                                                Shift: TShiftState);
begin
   if Key = VK_INSERT then
   begin
      InsertMode := Not InsertMode;
      if InsertMode then
         StatusBar1.Panels[4].Text := 'INS'
      else
         StatusBar1.Panels[4].Text := 'OVR';
      Key := 0;
   end;
   if Not InsertMode then
      IgnoreNextChar := DBRichViewEdit1.SelectionExists;
end;

procedure TCase_Doc_Form.DBRichViewEdit1KeyPress(Sender: TObject;
                                                 var Key: Char);
var
   ItemNo, Offs: Integer;
begin
   if Not InsertMode then
   begin
      if IgnoreNextChar then
      begin
         IgnoreNextChar := False;
         Exit;
      end;
      IgnoreNextChar := False;
      if not ((Key = #9) or (Key >= ' ')) then
         Exit;
      if DBRichViewEdit1.SelectionExists then
         Exit;
      ItemNo := DBRichViewEdit1.CurItemNo;
      Offs   := DBRichViewEdit1.OffsetInCurItem;
      if (Offs >= DBRichViewEdit1.GetOffsAfterItem(ItemNo)) then
      begin
         if (ItemNo + 1 < DBRichViewEdit1.ItemCount) and
            not DBRichViewEdit1.IsFromNewLine(ItemNo + 1) then
         begin
            Inc(ItemNo);
            Offs := DBRichViewEdit1.GetOffsBeforeItem(ItemNo);
         end
         else
```

```
            Exit;
      end;
      DBRichViewEdit1.SetSelectionBounds(ItemNo, Offs, ItemNo, Offs+1);
      DBRichViewEdit1.Invalidate;
   end;
end;

procedure TCase_Doc_Form.DBRichViewEdit1ReadHyperlink(
                                              Sender: TCustomRichView;
                                              const Target,
                                              Extras: string;
                                              DocFormat: TRVLoadFormat;
                                              var StyleNo,
                                              ItemTag: Integer;
                                              var ItemName:

TRVRawByteString);
begin
   ItemTag := Integer(StrNew(PChar(Target)));
end;

procedure TCase_Doc_Form.DBRichViewEdit1Select(Sender: TObject);
var
   fi: TFontInfo;
begin
   IgnoreChanges := True;
   fi := RVStyle1.TextStyles[DBRichViewEdit1.CurTextStyleNo];
   RVFontComboBox1.ItemIndex :=
                            RVFontComboBox1.Items.IndexOf(fi.FontName);
   RVFontSizeComboBox1.Text := IntToStr(fi.Size);
   IgnoreChanges := False;
end;

procedure TCase_Doc_Form.DBRichViewEdit1StyleConversion(
                                                  Sender:
                                                  TCustomRichViewEdit;
                                                  StyleNo, UserData:
                                                  Integer;
                                                  AppliedToText:
                                                  Boolean;
                                                  var NewStyleNo:
                                                  Integer);
var
   FontInfo: TFontInfo;
begin
   FontInfo := TFontInfo.Create(nil);
   try
      FontInfo.Assign(RVStyle1.TextStyles[StyleNo]);
      case UserData of
         TEXT_APPLYFONTNAME: FontInfo.FontName := FontName;
         TEXT_APPLYFONTSIZE: FontInfo.Size     := FontSize;
      end;
```

```
      NewStyleNo := RVStyle1.TextStyles.FindSuchStyle(
                    StyleNo,FontInfo,RVAllFontInfoProperties);
      if NewStyleNo = -1 then
      begin
         RVStyle1.TextStyles.Add;
         NewStyleNo := RVStyle1.TextStyles.Count-1;
         RVStyle1.TextStyles[NewStyleNo].Assign(FontInfo);
         RVStyle1.TextStyles[NewStyleNo].Standard := False;
      end;
   finally
      FontInfo.Free;
   end;
end;

procedure TCase_Doc_Form.DBRichViewEdit1WriteHyperlink(Sender:
                                                       TCustomRichView;
                                                       id: Integer;
                                                       RVData:
                                                       TCustomRVData;
                                                       ItemNo: Integer;
                                                       SaveFormat:
                                                       TRVSaveFormat;
                                                       var Target,
                                                       Extras: string);
begin
   Target := PChar(RVData.GetItemTag(ItemNo));
end;

class function TCase_Doc_Form.DocInEditMode: Boolean;
var
   F: TCase_Doc_Form;
   I: Integer;
begin
   F := Nil;
   for I := Screen.FormCount - 1 DownTo 0 do
   begin
      if (Screen.Forms[I] is TCase_Doc_Form) then
      begin
         F := Screen.Forms[I] As TCase_Doc_Form;
         if F.DocumentQuery.State <> dsEdit then
            F := Nil
         else
            Break;
      end;
   end;
   if F = Nil then
      Result := False
   else
   begin
      if F.WindowState = wsMinimized then
         F.WindowState := wsNormal;
      F.BringToFront;
```

```
      Result := True;
   end;
end;

procedure TCase_Doc_Form.DocumentDSStateChange(Sender: TObject);
begin
   if (Not (JustOpened) and Not (Date_Editing)) then
   begin
      if DocumentQuery.State in [dsEdit, dsInsert] then
         StatusBar1.Panels[1].Text := 'Status:  Editing'
      else
      begin
         if Op_Cancelled then
         begin
            StatusBar1.Panels[1].Text := 'Status:  Cancelled';
            Op_Cancelled := False;
         end
         else
            StatusBar1.Panels[1].Text := 'Status:  Saved';
      end;
   end;
end;

procedure TCase_Doc_Form.DocumentQueryAfterOpen(DataSet: TDataSet);
begin
   Date_Editing := True;
   DocumentQuery.Edit;
   DocumentQueryDatetime_Last_Accessed.AsDateTime := Now;
   DocumentQuery.Post;
   Date_Editing := False;
end;

procedure TCase_Doc_Form.DocumentQueryBeforeCancel(DataSet: TDataSet);
begin
   Op_Cancelled := True;
end;

procedure TCase_Doc_Form.DocumentQueryBeforeEdit(DataSet: TDataSet);
begin
   JustOpened := False;
end;

procedure TCase_Doc_Form.DocumentQueryBeforePost(DataSet: TDataSet);
begin
   if Not Date_Editing then
      DocumentQuerydatetime_last_modified.AsDateTime := Now;
end;

procedure TCase_Doc_Form.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
   if Key = #13 then
      ToolButton83Click(Sender);
```

```
end;

procedure TCase_Doc_Form.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   DocumentQuery.Close;
   Action := caFree;
end;

procedure TCase_Doc_Form.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
   if DocumentQuery.State = dsEdit then
   begin
      case MessageDlg('Cancel Edits?', mtWarning,
                      [mbYes, mbNo, mbCancel],0) of
         mrYes: begin
                   DocumentQuery.Cancel;
                   CanClose := True;
                end;
         mrNo : CanClose := False;
         mrCancel:  CanClose := False;
      end;
   end;
end;

procedure TCase_Doc_Form.FormCreate(Sender: TObject);

  procedure InitializeVariables;
  begin
     JustOpened := True;
     rvActionExport1.InitialDir :=
                                 GetSpecialFolderPath(CSIDL_PERSONAL)+
                                   '\Thesis_Docs\Word_Documents';
     rvActionInsertFile1.InitialDir :=
                                 GetSpecialFolderPath(CSIDL_PERSONAL)+
                                       '\Thesis_Docs\Word_Documents';
     StatusBar1.Panels[4].Text := 'INS';
     StatusBar1.Panels[1].Text := 'Status:  Unchanged';
     InsertMode := True;
     IgnoreNextChar := False;
     Op_Cancelled := False;
     Date_Editing := False;
     Top := 0;
     Left := 25;
     Height := Screen.Height - 30;
  end;

var
   fi: TFontInfo;
begin
   InitializeVariables;
```

```
   RVVisibleSpecialCharacters := RVVisibleSpecialCharacters -
                                 [rvscParagraphAttrs];
   DocumentQuery.ParamByName(
                    'idcase_document_param').Value := DocumentID;
   DocumentQuery.Open;
   DBRichViewEdit1Change(Sender);
   Caption := Caption + ' - '+DocumentQuerydocument_name.AsString;
   fi := RVStyle1.TextStyles[DBRichViewEdit1.CurTextStyleNo];
   RVFontComboBox1.ItemIndex :=
RVFontComboBox1.Items.IndexOf(fi.FontName);
   RVFontSizeComboBox1.Text := IntToStr(fi.Size);
end;

function TCase_Doc_Form.GetSpecialFolderPath(
                                        Folder:  integer):  string;
const
   SHGFP_TYPE_CURRENT = 0;
var
   Path: Array[0..MAX_PATH] of char;
begin
   if SUCCEEDED(SHGetFolderPath(0,folder,0,
                SHGFP_TYPE_CURRENT,@path[0])) then
      Result := path
   else
      Result := '';
end;

procedure TCase_Doc_Form.RemoveAllPageBreaks(RVData: TCustomRVData;
  RVE: TCustomRichViewEdit);
var
   I, R, C: Integer;
   Table:   TRVTableItemInfo;
begin
   try
      for I := 0 to RVData.ItemCount - 1 do
      begin
         if RVData.PageBreaksBeforeItems[I] then
         begin
            RVData := RVData.Edit;
            TCustomRVFormattedData(RVData).SetSelectionBounds(I,
                      RVData.GetOffsBeforeItem(I), I,
                      RVData.GetOffsBeforeItem(I));
            RVE.TopLevelEditor.RemoveCurrentPageBreak;
         end;
         if RVData.GetItemStyle(I) = rvsTable then
         begin
            Table := TRVTableItemInfo(RVData.GetItem(I));
            for R := 0 to Table.RowCount - 1 do
               for C := 0 to Table.ColCount - 1 do
                  if Table.Cells[R,C]<> nil then
                     RemoveAllPageBreaks(Table.Cells[R,C].GetRVData,
                                         RVE);
```

```
         end;
      end;
   except

   end;
end;

procedure TCase_Doc_Form.RVFontComboBox1Change(Sender: TObject);
begin
   if (RVFontComboBox1.ItemIndex <> -1) and (Not IgnoreChanges) then
   begin
      if not IgnoreChanges then
      begin
         FontName := RVFontComboBox1.Items[RVFontComboBox1.ItemIndex];
         DBRichViewEdit1.ApplyStyleConversion(TEXT_APPLYFONTNAME);
      end;
   end;
   if (Visible) then
      DBRichViewEdit1.SetFocus;
end;

procedure TCase_Doc_Form.RVFontSizeComboBox1Change(Sender: TObject);
begin
   if (RVFontSizeComboBox1.ItemIndex <> -1) and not IgnoreChanges then
   begin
      FontSize := StrToIntDef(RVFontSizeComboBox1.Text, 10);
      DBRichViewEdit1.ApplyStyleConversion(TEXT_APPLYFONTSIZE);
   end;
   if (Visible) then
      DBRichViewEdit1.SetFocus;
end;

procedure TCase_Doc_Form.StatusBar1DblClick(Sender: TObject);
var
   mpt:             TPoint;
   x, j,
   clicked_header:  Integer;
begin
   mpt := mouse.CursorPos;
   mpt := StatusBar1.ScreenToClient(mpt);
   clicked_header := -1;
   x := 0;
   for j := 0 to StatusBar1.Panels.Count - 1 do
   begin
      x := x + StatusBar1.panels[j].Width;
      if mpt.X < x  then
      begin
         clicked_header := j;
         break
      end;
   end;
   case clicked_header of
```

```
      4: begin
            DBRichViewEdit1.Perform(WM_KEYDOWN, VK_INSERT,0);
            DBRichViewEdit1.Perform(WM_KEYUP, VK_INSERT,0);
         end;
   end;
end;

procedure TCase_Doc_Form.ToolButton83Click(Sender: TObject);
begin
   if Edit1.Text = '' then
      MessageDlg('Nothing to Mark', mtError, [mbok],0)
   else
   begin
      if Not (DocumentQuery.State in [dsEdit]) then
      begin
         DocumentQuery.ReadOnly := True;
         MarkSubstringW(Edit1.Text,clRed,clYellow,True,
                        False,DBRichViewEdit1,DBRichViewEdit1.RVData);
         Edit1.Text := '';
         DBRichViewEdit1.Format;
         DBRichViewEdit1.SetFocus;
         DBRichViewEdit1.ReadOnly := True;
      end
      else
         MessageDlg('Formatting Changes Cannot Be Undone,'+
                    slinebreak+
                    'Save or Cancel Changes Before Executing Search',
                    mtWarning, [mbOK],0);
   end;
end;

procedure TCase_Doc_Form.ToolButton88Click(Sender: TObject);

   procedure RemoveAllPageBreaks(RVData: TCustomRVData;
                                 RVE: TCustomRichViewEdit);
   var
      I, R, C: Integer;
      Table: TRVTableItemInfo;
   begin
      for I := 0 to RVData.ItemCount - 1 do
      begin
         if RVData.PageBreaksBeforeItems[I] then
         begin
            RVData := RVData.Edit;
            TCustomRVFormattedData(RVData).SetSelectionBounds(
                        I, RVData.GetOffsBeforeItem(I),
                        I, RVData.GetOffsBeforeItem(I));
            RVE.TopLevelEditor.RemoveCurrentPageBreak;
         end;
         if RVData.GetItemStyle(I) = rvsTable then
         begin
            Table := TRVTableItemInfo(RVData.GetItem(I));
```

```
            for R := 0 to Table.RowCount - 1 do
               for C := 0 to Table.ColCount - 1 do
                  if Table.Cells[R,C]<> nil then
                     RemoveAllPageBreaks(Table.Cells[R,C].GetRVData,
                                         RVE);
         end;
      end;
   end;

var
   RVCaretPos:  Integer;
begin
   try
      DBRichViewEdit1.BeginUpdate;
      RVCaretPos := RVGetLinearCaretPos(DBRichViewEdit1);
      RemoveAllPageBreaks(DBRichViewEdit1.RVData, DBRichViewEdit1);
      RVSetLinearCaretPos(DBRichViewEdit1, 0);
      rd.FindText := #$A0;
      rd.ReplaceText := #32;
      while DBRichViewEdit1.SearchText(rd.FindText,
            GetRVESearchOptions(rd.Options)) do
         DBRichViewEdit1.InsertText(rd.ReplaceText, False);
      RVSetLinearCaretPos(DBRichViewEdit1, RVCaretPos);
      DBRichViewEdit1.EndUpdate;
   except
      MessageDlg('Unknown Error Attempting Operation!', mtError,
[mbOK],0);
   end;
end;

{ TWordCounter }

function TWordCounter.ProcessWord: Boolean;
begin
   Inc(fCounter);
   Result := True;
end;

function TWordCounter.GetWordCount(rve: TCustomRichViewEdit): Integer;
begin
   fCounter := 0;
   Run(rve, rvesFromStart);
   Result := fCounter;
end;

end.
```

Document.dfm

```
object Case_Doc_Form: TCase_Doc_Form
  Left = 0
```

```
  Top = 0
  ActiveControl = DBRichViewEdit1
  BorderIcons = [biSystemMenu]
  Caption = 'Case Document'
  ClientHeight = 700
  ClientWidth = 930
  Color = clBtnFace
  Constraints.MinHeight = 200
  Constraints.MinWidth = 900
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  Position = poDesigned
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  DesignSize = (
    930
    700)
  PixelsPerInch = 96
  TextHeight = 13
  object DBRichViewEdit1: TDBRichViewEdit
    Left = 31
    Top = 106
    Width = 888
    Height = 575
    DataField = 'document_text'
    DataSource = DocumentDS
    ReadOnly = False
    AcceptDragDropFormats = [rvddRVF, rvddRTF, rvddText,
rvddUnicodeText, rvddBitmap, rvddMetafile, rvddURL, rvddFiles]
    OnCaretMove = DBRichViewEdit1CaretMove
    OnChange = DBRichViewEdit1Change
    OnCurTextStyleChanged = DBRichViewEdit1CurTextStyleChanged
    OnStyleConversion = DBRichViewEdit1StyleConversion
    Anchors = [akLeft, akTop, akRight, akBottom]
    PopupMenu = RVAPopupMenu1
    TabOrder = 0
    OnKeyDown = DBRichViewEdit1KeyDown
    OnKeyPress = DBRichViewEdit1KeyPress
    DocParameters.LeftMargin = 1.000000000000000000
    DocParameters.RightMargin = 1.000000000000000000
    DoInPaletteMode = rvpaCreateCopies
    Options = [rvoAllowSelection, rvoScrollToEnd, rvoShowPageBreaks,
rvoTagsArePChars, rvoAutoCopyText, rvoAutoCopyUnicodeText,
rvoAutoCopyRVF, rvoAutoCopyImage, rvoAutoCopyRTF, rvoFormatInvalidate,
rvoDblClickSelectsWord, rvoRClickDeselects, rvoShowItemHints,
rvoFastFormatting]
    RightMargin = 20
```

```
    RTFOptions = [rvrtfDuplicateUnicode, rvrtfSaveJpegAsJpeg]
    RTFReadProperties.TextStyleMode = rvrsAddIfNeeded
    RTFReadProperties.ParaStyleMode = rvrsAddIfNeeded
    RVFOptions = [rvfoSavePicturesBody, rvfoSaveControlsBody,
rvfoSaveBinary, rvfoSaveBack, rvfoLoadBack, rvfoSaveTextStyles,
rvfoSaveParaStyles, rvfoSaveLayout, rvfoLoadLayout,
rvfoSaveDocProperties, rvfoLoadDocProperties]
    Style = RVStyle1
    OnJump = DBRichViewEdit1Jump
    OnReadHyperlink = DBRichViewEdit1ReadHyperlink
    OnSelect = DBRichViewEdit1Select
    OnWriteHyperlink = DBRichViewEdit1WriteHyperlink
  end
  object RVRuler1: TRVRuler
    Left = 31
    Top = 83
    Width = 891
    Height = 25
    Anchors = [akLeft, akTop, akRight]
    BottomMargin = 1.000000000000000000
    DefaultTabWidth = 0.500000000000000000
    Flat = False
    LeftMargin = 0.200000000000000000
    ParentBackground = False
    RightMargin = 0.200000000000000000
    RulerTexts.HintColumnMove = 'Resize table column'
    RulerTexts.HintIndentFirst = 'First Indent'
    RulerTexts.HintIndentLeft = 'Left Indent'
    RulerTexts.HintIndentHanging = 'Hanging Indent'
    RulerTexts.HintIndentRight = 'Right Indent'
    RulerTexts.HintLevelDec = 'Decrease level'
    RulerTexts.HintLevelInc = 'Increase level'
    RulerTexts.HintMarginBottom = 'Bottom margin'
    RulerTexts.HintMarginLeft = 'Left margin'
    RulerTexts.HintMarginRight = 'Right margin'
    RulerTexts.HintMarginTop = 'Top margin'
    RulerTexts.HintRowMove = 'Resize table row'
    RulerTexts.HintTabCenter = 'Center aligned tab'
    RulerTexts.HintTabDecimal = 'Decimal aligned tab'
    RulerTexts.HintTabLeft = 'Left aligned tab'
    RulerTexts.HintTabRight = 'Right aligned tab'
    RulerTexts.HintTabWordBar = 'Word Bar aligned tab'
    RulerTexts.MenuTabCenter = 'Center align'
    RulerTexts.MenuTabDecimal = 'Decimal align'
    RulerTexts.MenuTabLeft = 'Left align'
    RulerTexts.MenuTabRight = 'Right align'
    RulerTexts.MenuTabWordBar = 'Word Bar align'
    Tabs = <>
    TabSettings.DeleteCursor = crDrag
    TopMargin = 1.000000000000000000
    UnitsDisplay = ruInches
    UnitsProgram = ruInches
```

```
    RichViewEdit = DBRichViewEdit1
    TableEditor.Active = False
    TableEditor.CellIndex = 0
    TableEditor.Cells = <>
    TableEditor.RowIndex = 0
    TableEditor.Rows = <>
    TableEditor.TableOffset = 0
  end
  object RVRuler2: TRVRuler
    Left = 2
    Top = 106
    Width = 25
    Height = 575
    Anchors = [akLeft, akTop, akBottom]
    BottomMargin = 0.200000000000000000
    DefaultTabWidth = 0.500000000000000000
    Flat = False
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -11
    Font.Name = 'Tahoma'
    Font.Orientation = 900
    Font.Style = []
    IndentSettings.Options = []
    LeftMargin = 1.000000000000000000
    MarginSettings.DragCursor = crSizeNS
    MaxTabs = 0
    ParentBackground = False
    ParentFont = False
    RightMargin = 1.000000000000000000
    RulerTexts.HintColumnMove = 'Resize table column'
    RulerTexts.HintIndentFirst = 'First Indent'
    RulerTexts.HintIndentLeft = 'Left Indent'
    RulerTexts.HintIndentHanging = 'Hanging Indent'
    RulerTexts.HintIndentRight = 'Right Indent'
    RulerTexts.HintLevelDec = 'Decrease level'
    RulerTexts.HintLevelInc = 'Increase level'
    RulerTexts.HintMarginBottom = 'Bottom margin'
    RulerTexts.HintMarginLeft = 'Left margin'
    RulerTexts.HintMarginRight = 'Right margin'
    RulerTexts.HintMarginTop = 'Top margin'
    RulerTexts.HintRowMove = 'Resize table row'
    RulerTexts.HintTabCenter = 'Center aligned tab'
    RulerTexts.HintTabDecimal = 'Decimal aligned tab'
    RulerTexts.HintTabLeft = 'Left aligned tab'
    RulerTexts.HintTabRight = 'Right aligned tab'
    RulerTexts.HintTabWordBar = 'Word Bar aligned tab'
    RulerTexts.MenuTabCenter = 'Center align'
    RulerTexts.MenuTabDecimal = 'Decimal align'
    RulerTexts.MenuTabLeft = 'Left align'
    RulerTexts.MenuTabRight = 'Right align'
    RulerTexts.MenuTabWordBar = 'Word Bar align'
```

```
    RulerType = rtVertical
    Tabs = <>
    TabSettings.DeleteCursor = crDrag
    TabSettings.Options = []
    TopMargin = 0.200000000000000000
    UnitsDisplay = ruInches
    UnitsProgram = ruInches
    RichViewEdit = DBRichViewEdit1
    TableEditor.Active = False
    TableEditor.CellIndex = 0
    TableEditor.Cells = <>
    TableEditor.DragCursor = crVSplit
    TableEditor.RowIndex = 0
    TableEditor.Rows = <>
    TableEditor.TableOffset = 0
  end
  object RVRulerItemSelector1: TRVRulerItemSelector
    Left = 8
    Top = 83
    Width = 17
    Height = 17
    Ruler = RVRuler1
    TabOrder = 3
  end
  object CoolBar1: TCoolBar
    Left = 0
    Top = 0
    Width = 930
    Height = 81
    Bands = <
      item
        Control = ToolBar1
        ImageIndex = -1
        Width = 924
      end
      item
        Control = ToolBar2
        ImageIndex = -1
        Width = 924
      end
      item
        Control = ToolBar3
        ImageIndex = -1
        Width = 924
      end>
    object ToolBar1: TToolBar
      Left = 11
      Top = 0
      Width = 915
      Height = 25
      Caption = 'ToolBar1'
      Images = ImageList1
```

```
TabOrder = 0
object ToolButton1: TToolButton
  Left = 0
  Top = 0
  Action = DataSetPost1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton2: TToolButton
  Left = 23
  Top = 0
  Action = DataSetCancel1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton62: TToolButton
  Left = 46
  Top = 0
  Width = 8
  Caption = 'ToolButton62'
  ImageIndex = 83
  Style = tbsSeparator
end
object ToolButton82: TToolButton
  Left = 54
  Top = 0
  Action = rvActionPrintPreview1
end
object ToolButton7: TToolButton
  Left = 77
  Top = 0
  Action = rvActionPrint1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton6: TToolButton
  Left = 100
  Top = 0
  Width = 8
  Caption = 'ToolButton6'
  ImageIndex = 86
  Style = tbsSeparator
end
object ToolButton4: TToolButton
  Left = 108
  Top = 0
  Action = rvActionInsertFile1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton5: TToolButton
  Left = 131
```

```
    Top = 0
    Action = rvActionExport1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton8: TToolButton
    Left = 154
    Top = 0
    Width = 8
    Caption = 'ToolButton8'
    ImageIndex = 87
    Style = tbsSeparator
  end
  object ToolButton3: TToolButton
    Left = 162
    Top = 0
    Action = rvActionFind1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton9: TToolButton
    Left = 185
    Top = 0
    Action = rvActionFindNext1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton36: TToolButton
    Left = 208
    Top = 0
    Action = rvActionReplace1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton10: TToolButton
    Left = 231
    Top = 0
    Width = 8
    Caption = 'ToolButton10'
    ImageIndex = 16
    Style = tbsSeparator
  end
  object ToolButton11: TToolButton
    Left = 239
    Top = 0
    Hint = 'Font Properties'
    Action = rvActionFontEx1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton12: TToolButton
    Left = 262
```

```
    Top = 0
    Hint = 'Bullet List Properties'
    Action = rvActionParaList1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton13: TToolButton
    Left = 285
    Top = 0
    Hint = 'Paragraph Properties'
    Action = rvActionParagraph1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton14: TToolButton
    Left = 308
    Top = 0
    Action = rvActionTableProperties1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton15: TToolButton
    Left = 331
    Top = 0
    Width = 8
    Caption = 'ToolButton15'
    ImageIndex = 81
    Style = tbsSeparator
  end
  object ToolButton63: TToolButton
    Left = 339
    Top = 0
    Action = rvActionAlignLeft1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton64: TToolButton
    Left = 362
    Top = 0
    Action = rvActionAlignCenter1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton65: TToolButton
    Left = 385
    Top = 0
    Action = rvActionAlignRight1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton66: TToolButton
    Left = 408
```

```
      Top = 0
      Action = rvActionAlignJustify1
      ParentShowHint = False
      ShowHint = True
    end
    object ToolButton16: TToolButton
      Left = 431
      Top = 0
      Width = 8
      Caption = 'ToolButton16'
      ImageIndex = 82
      Style = tbsSeparator
    end
    object RVFontComboBox1: TRVadlFontComboBox
      Left = 439
      Top = 0
      Width = 184
      Height = 21
      TabOrder = 0
      OnChange = RVFontComboBox1Change
      OnCloseUp = RVFontComboBox1Change
    end
    object RVFontSizeComboBox1: TRVFontSizeComboBox1
      Left = 623
      Top = 0
      Width = 65
      Height = 21
      TabOrder = 1
      OnChange = RVFontSizeComboBox1Change
      OnCloseUp = RVFontSizeComboBox1Change
      Items.Strings = (
        '5'
        '8'
        '9'
        '11'
        '12'
        '14'
        '16'
        '18'
        '20'
        '22'
        '24'
        '26'
        '28'
        '36'
        '48'
        '72')
    end
  end
  object ToolBar2: TToolBar
    Left = 11
    Top = 27
```

```
Width = 915
Height = 25
Caption = 'ToolBar2'
Images = ImageList1
TabOrder = 1
object ToolButton34: TToolButton
  Left = 0
  Top = 0
  Action = rvActionUndo1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton35: TToolButton
  Left = 23
  Top = 0
  Action = rvActionRedo1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton80: TToolButton
  Left = 46
  Top = 0
  Width = 8
  Caption = 'ToolButton80'
  ImageIndex = 58
  Style = tbsSeparator
end
object ToolButton17: TToolButton
  Left = 54
  Top = 0
  Action = rvActionCut1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton18: TToolButton
  Left = 77
  Top = 0
  Action = rvActionCopy1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton19: TToolButton
  Left = 100
  Top = 0
  Action = rvActionPaste1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton20: TToolButton
  Left = 123
  Top = 0
  Action = rvActionPasteSpecial1
```

```
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton21: TToolButton
    Left = 146
    Top = 0
    Width = 8
    Caption = 'ToolButton21'
    ImageIndex = 87
    Style = tbsSeparator
  end
  object ToolButton29: TToolButton
    Left = 154
    Top = 0
    Action = rvActionFontColor1
  end
  object ToolButton31: TToolButton
    Left = 177
    Top = 0
    Action = rvActionFontBackColor1
  end
  object ToolButton85: TToolButton
    Left = 200
    Top = 0
    Action = rvActionParaColor1
  end
  object ToolButton30: TToolButton
    Left = 223
    Top = 0
    Action = rvActionColor1
  end
  object ToolButton74: TToolButton
    Left = 246
    Top = 0
    Action = rvActionBackground1
  end
  object ToolButton32: TToolButton
    Left = 269
    Top = 0
    Width = 8
    Caption = 'ToolButton32'
    ImageIndex = 90
    Style = tbsSeparator
  end
  object ToolButton22: TToolButton
    Left = 277
    Top = 0
    Action = rvActionFontBold1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton23: TToolButton
```

```
  Left = 300
  Top = 0
  Action = rvActionFontItalic1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton24: TToolButton
  Left = 323
  Top = 0
  Action = rvActionFontUnderline1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton25: TToolButton
  Left = 346
  Top = 0
  Action = rvActionFontStrikeout1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton26: TToolButton
  Left = 369
  Top = 0
  Action = rvActionSuperscript1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton27: TToolButton
  Left = 392
  Top = 0
  Action = rvActionSubscript1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton28: TToolButton
  Left = 415
  Top = 0
  Width = 8
  Caption = 'ToolButton28'
  ImageIndex = 33
  Style = tbsSeparator
end
object ToolButton33: TToolButton
  Left = 423
  Top = 0
  Action = rvActionParaBullets1
  ParentShowHint = False
  ShowHint = True
end
object ToolButton37: TToolButton
  Left = 446
  Top = 0
```

```
    Action = rvActionParaNumbering1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton38: TToolButton
    Left = 469
    Top = 0
    Width = 8
    Caption = 'ToolButton38'
    ImageIndex = 49
    Style = tbsSeparator
  end
  object ToolButton39: TToolButton
    Left = 477
    Top = 0
    Action = rvActionIndentDec1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton40: TToolButton
    Left = 500
    Top = 0
    Action = rvActionIndentInc1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton41: TToolButton
    Left = 523
    Top = 0
    Action = rvActionShowSpecialCharacters1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton43: TToolButton
    Left = 546
    Top = 0
    Width = 8
    Caption = 'ToolButton43'
    ImageIndex = 62
    Style = tbsSeparator
  end
  object ToolButton42: TToolButton
    Left = 554
    Top = 0
    Action = rvActionLineSpacing1001
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton44: TToolButton
    Left = 577
    Top = 0
    Action = rvActionLineSpacing1501
```

```
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton45: TToolButton
    Left = 600
    Top = 0
    Action = rvActionLineSpacing2001
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton46: TToolButton
    Left = 623
    Top = 0
    Width = 8
    Caption = 'ToolButton46'
    ImageIndex = 46
    Style = tbsSeparator
  end
  object ToolButton47: TToolButton
    Left = 631
    Top = 0
    Action = rvActionInsertPicture1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton48: TToolButton
    Left = 654
    Top = 0
    Action = rvActionInsertHLine1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton49: TToolButton
    Left = 677
    Top = 0
    Action = rvActionInsertSymbol1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton50: TToolButton
    Left = 700
    Top = 0
    Action = rvActionInsertPageBreak1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton84: TToolButton
    Left = 723
    Top = 0
    Action = rvActionInsertHyperlink1
    ParentShowHint = False
    ShowHint = True
```

```
    end
    object ToolButton87: TToolButton
      Left = 746
      Top = 0
      Width = 8
      Caption = 'ToolButton87'
      ImageIndex = 90
      Style = tbsSeparator
    end
    object ToolButton86: TToolButton
      Left = 754
      Top = 0
      Action = rvActionRemoveHyperlinks1
      ParentShowHint = False
      ShowHint = True
    end
    object ToolButton88: TToolButton
      Left = 777
      Top = 0
      Hint = 'Remove All Page Breaks'
      Caption = 'ToolButton88'
      ImageIndex = 90
      ParentShowHint = False
      ShowHint = True
      OnClick = ToolButton88Click
    end
  end
  object ToolBar3: TToolBar
    Left = 11
    Top = 54
    Width = 915
    Height = 25
    Caption = 'ToolBar3'
    Images = ImageList1
    TabOrder = 2
    object ToolButton51: TToolButton
      Left = 0
      Top = 0
      Action = rvActionInsertTable1
      ParentShowHint = False
      ShowHint = True
    end
    object ToolButton70: TToolButton
      Left = 23
      Top = 0
      Action = rvActionTableDeleteTable1
      ParentShowHint = False
      ShowHint = True
    end
    object ToolButton53: TToolButton
      Left = 46
      Top = 0
```

```
    Width = 8
    Caption = 'ToolButton53'
    ImageIndex = 65
    Style = tbsSeparator
  end
  object ToolButton52: TToolButton
    Left = 54
    Top = 0
    Action = rvActionTableInsertColLeft1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton54: TToolButton
    Left = 77
    Top = 0
    Action = rvActionTableInsertColRight1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton55: TToolButton
    Left = 100
    Top = 0
    Action = rvActionTableInsertRowsAbove1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton56: TToolButton
    Left = 123
    Top = 0
    Action = rvActionTableInsertRowsBelow1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton71: TToolButton
    Left = 146
    Top = 0
    Action = rvActionTableDeleteCols1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton72: TToolButton
    Left = 169
    Top = 0
    Action = rvActionTableDeleteRows1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton73: TToolButton
    Left = 192
    Top = 0
    Width = 8
    Caption = 'ToolButton73'
```

```
    ImageIndex = 69
    Style = tbsSeparator
  end
  object ToolButton58: TToolButton
    Left = 200
    Top = 0
    Action = rvActionTableMergeCells1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton59: TToolButton
    Left = 223
    Top = 0
    Action = rvActionTableSplitCells1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton60: TToolButton
    Left = 246
    Top = 0
    Width = 8
    Caption = 'ToolButton60'
    ImageIndex = 71
    Style = tbsSeparator
  end
  object ToolButton61: TToolButton
    Left = 254
    Top = 0
    Action = rvActionTableCellVAlignTop1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton67: TToolButton
    Left = 277
    Top = 0
    Action = rvActionTableCellVAlignMiddle1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton68: TToolButton
    Left = 300
    Top = 0
    Action = rvActionTableCellVAlignBottom1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton69: TToolButton
    Left = 323
    Top = 0
    Width = 8
    Caption = 'ToolButton69'
    ImageIndex = 74
```

```
    Style = tbsSeparator
  end
  object ToolButton57: TToolButton
    Left = 331
    Top = 0
    Action = rvActionTableCellAllBorders1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton75: TToolButton
    Left = 354
    Top = 0
    Action = rvActionTableCellNoBorders1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton76: TToolButton
    Left = 377
    Top = 0
    Action = rvActionTableCellLeftBorder1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton77: TToolButton
    Left = 400
    Top = 0
    Action = rvActionTableCellBottomBorder1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton78: TToolButton
    Left = 423
    Top = 0
    Action = rvActionTableCellRightBorder1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton79: TToolButton
    Left = 446
    Top = 0
    Action = rvActionTableCellTopBorder1
    ParentShowHint = False
    ShowHint = True
  end
  object ToolButton81: TToolButton
    Left = 469
    Top = 0
    Width = 8
    Caption = 'ToolButton81'
    ImageIndex = 77
    Style = tbsSeparator
  end
```

```
      object Edit1: TEdit
        Left = 477
        Top = 0
        Width = 244
        Height = 22
        TabOrder = 0
        OnKeyPress = Edit1KeyPress
      end
      object ToolButton83: TToolButton
        Left = 721
        Top = 0
        Caption = 'ToolButton83'
        ImageIndex = 87
        OnClick = ToolButton83Click
      end
    end
  end
  object StatusBar1: TStatusBar
    Left = 0
    Top = 681
    Width = 930
    Height = 19
    Panels = <
      item
        Width = 10
      end
      item
        Width = 140
      end
      item
        Width = 80
      end
      item
        Width = 80
      end
      item
        Width = 80
      end
      item
        Width = 80
      end
      item
        Width = 80
      end
      item
        Width = 100
      end
      item
        Width = 100
      end
      item
        Width = 50
```

```
    end>
  OnDblClick = StatusBar1DblClick
end
object DocumentDS: TDataSource
  DataSet = DocumentQuery
  OnStateChange = DocumentDSStateChange
  Left = 264
  Top = 120
end
object DocumentQuery: TZQuery
  Connection = Scanner_Main_Form.ZConnection1
  AfterOpen = DocumentQueryAfterOpen
  BeforeEdit = DocumentQueryBeforeEdit
  BeforePost = DocumentQueryBeforePost
  BeforeCancel = DocumentQueryBeforeCancel
  SQL.Strings = (
    'Select *'
    'from'
    '   case_document'
    'where'
    '   idcase_document = :idcase_document_param')
  Params = <
    item
      DataType = ftUnknown
      Name = 'idcase_document_param'
      ParamType = ptUnknown
    end>
  Left = 344
  Top = 120
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'idcase_document_param'
      ParamType = ptUnknown
    end>
  object DocumentQueryidcase_document: TIntegerField
    FieldName = 'idcase_document'
    Required = True
  end
  object DocumentQueryidyear: TIntegerField
    FieldName = 'idyear'
    Required = True
  end
  object DocumentQueryidcase: TIntegerField
    FieldName = 'idcase'
    Required = True
  end
  object DocumentQueryiddocument_type: TIntegerField
    FieldName = 'iddocument_type'
    Required = True
  end
  object DocumentQuerydocument_name: TWideStringField
```

```
      FieldName = 'document_name'
      Required = True
      Size = 45
    end
    object DocumentQuerydocument_text: TBlobField
      FieldName = 'document_text'
    end
    object DocumentQuerydatetime_last_scan: TDateTimeField
      FieldName = 'datetime_last_scan'
    end
    object DocumentQuerydatetime_last_modified: TDateTimeField
      FieldName = 'datetime_last_modified'
      Required = True
    end
    object DocumentQuerydatetime_created: TDateTimeField
      FieldName = 'datetime_created'
      Required = True
    end
    object DocumentQuerydatetime_last_accessed: TDateTimeField
      FieldName = 'datetime_last_accessed'
      Required = True
    end
  end
  object ImageList1: TImageList
    Left = 112
    Top = 120
    Bitmap = {
      494C01015B00600004001000100FFFFFFFFFF10FFFFFFFFFFFFFFFF424D3600
      0000000000003600000028000000400000007001000001002000000000000070
      0100000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000006060
      6000BFBFBF00202020000000000000000000FFFFFF01E0E0E01FCCCCCC33CCCC
      CC33CCCCCC33DCDCDC23FFFFFF01FFFFFF01FFFFFF01FFFFFF01E0E0E01FCCCC
      CC33CCCCCC33CCCCCC33E0E0E01FFFFFFF01A2A2A2C1919191E5919191E59191
      91E5919191E5919191E5919191E5919191E5919191E5919191E5919191E59191
      91E5919191E5919191E5919191E5A2A2A2C1000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000003F3F0000000000000000000000000000000040404000BFBF
      BF00404040002020200000000000202020000E0E0E01FBDBBBBAD676564FF6765
      64FF636160FF9C9A99BDDCDCDC23FFFFFF01FFFFFF01E0E0E01F7380CCAD384D
      CCFF384DCBFF384DCCFF7380CCADE0E0E01F919191E5FFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF919191E5000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000007F7F0000000000000000000000000000000040404000F9F9
      9F009F9F9F00808080000202020009F9F9F00C4C3C1AD6A6866FFB7B5B3FFB2B0
      AEFFAEABA8FF615F5EFFAFAEABB9E2E2E21DFFFFFF01717FCCAD3852D7FF385D
      F9FF375DFAFF385DF9FF3852D7FF7280CCAD919191E5FFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF919191E5000000000000000000000000000000
```

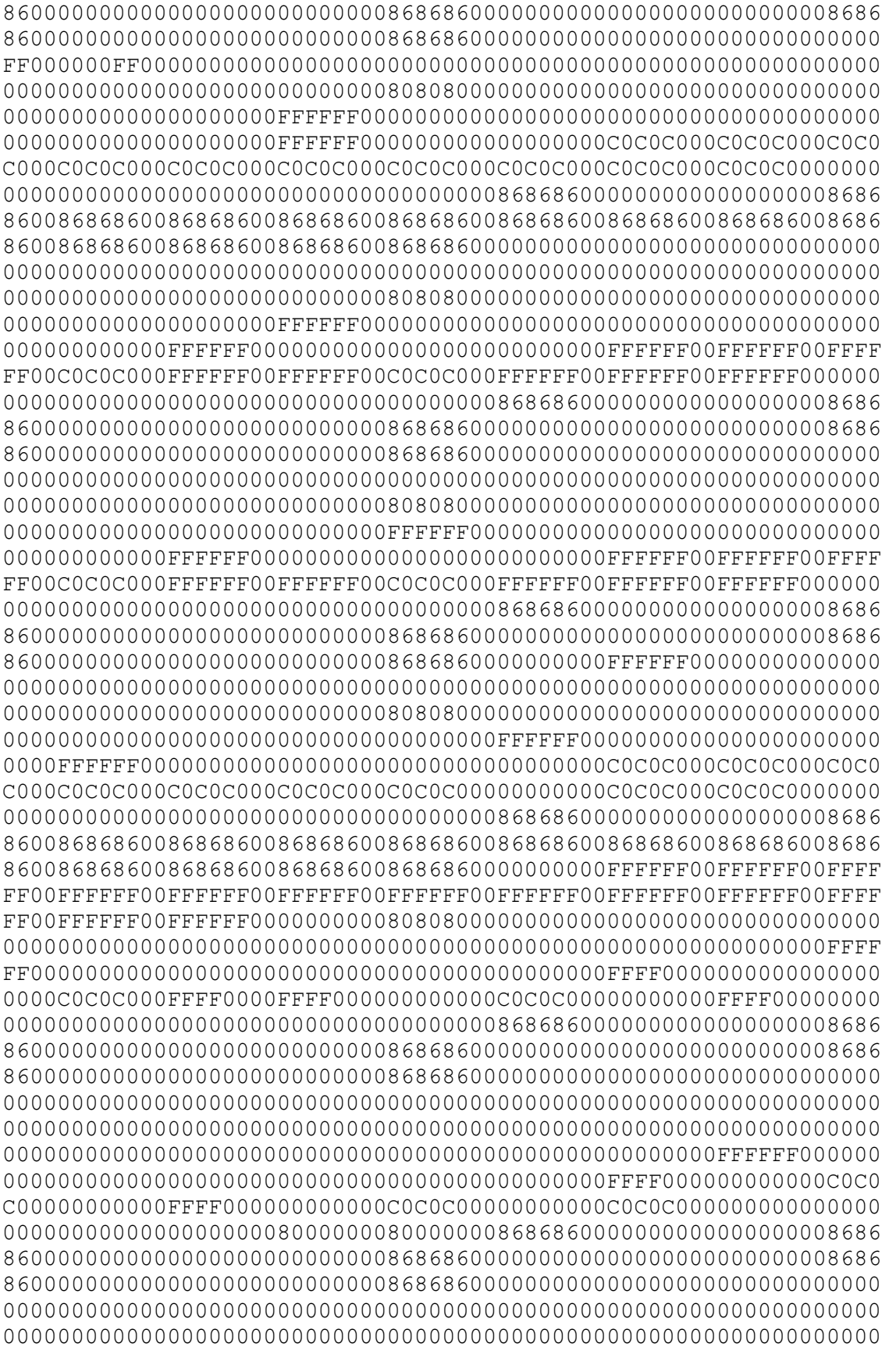
```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000007F7F0000000000000000000000000000000004040009F9F
9F009F9F9F00202020009F9F9F007F7F7F00767472FFC1C1BEFF7F7D7BFFD4D5
D575716F6DFF9E9C99FF63615EFFB8B4ABA9FFFFFF01334ACDFF4165FBFF3C60
FAFF3A5DF8FF3C60FAFF4165FBFF344BCCFF919191E5FEFEFEFF494949FF4949
49FFFEFEFEFFFEFEFEFFFEFEFEFFFFFEFEFFFFFEFEFFB0691EFFFFFEFEFFFEFE
FEFFFEFEFEFFFEFEFEFFFEFEFEFF919191E5000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000007F7F0000000000000000000000002020009F9F9F009F9F
9F00202020006060600060606000000000007C7A78FFD5D3D2FFB5B5B589FFFF
FF01CCCCCC33747270FFB1ADA8FF6D6858FFFFFFFF013048CCFFA9BAFFFFFFFF
FFFFFFFFFFFFFFFFFFFFA9BAFFFF3148CAFF919191E5FEFEFEFF494949FF4949
49FFFDFDFDFFFDFDFDFFFDFDFDFFFDFCFCFFB87733FFB1681FFFFDFCFCFFFDFC
FCFFFDFCFCFFFDFDFDFFFDFDFDFF919191E5000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000007F7F0000606060009F9F9F00808080009F9F9F009F9F9F002020
2000000000000000000000000000000007F7D7AFFEEEDECFF83817FFFCCCC
CC336D6A68FF585656FFB8B6B2FF757061FFFFFFFF013249CDFF5D79FFFF5776
FEFF5473FDFF5776FEFF5D79FFFF334ACBFF919191E5FEFEFEFF494949FF4949
49FFFCFCFCFFFCFCFCFFFCFCFCFFC0874CFFB87634FFB0681FFFAA5D0DFFA654
00FFFCFAFAFFFCFCFCFFFCFCFCFF919191E5000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000009F9F2000BFBFBF00404040009F9F9F009F9F9F00202020000000
0000000000000000000000000000000000CCCBCAAD7A7876FFEAEBEAFF8482
80FF7A7876FFB9B8B7FF585553FF424036FFFFFFFF018C9AE68F455EE0FF6F89
FFFF738BFFFF6F89FFFF465EDDFF8795E197919191E5FDFDFDFF494949FF4949
49FFFBFBFBFFFBFBFBFFFBFBFBFFFBF8F8FFB87634FFB0681EFFFBF8F8FFA654
00FFFBF8F8FFFBFBFBFFFBFBFBFF919191E5000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
000060606000DFDF5F00202020004040009F9F9F00202020003F3F3F000000
0000000000000000000000000000000000FFFFFF01E2E2E0A7787675FFEEED
ECFFD7D5D5FF797775FFF6F6F6FF5B5852FFCCCCCC33CCCCCC337F8BD1972D46
D4FF2C46D1FF3048CEFF8694E197FFFFFF01919191E5FDFDFDFF494949FF4949
49FFFBFBFBFFFBFBFBFFFBFBFBFFFBF8F8FFFBF8F8FFB1681FFFFBF8F8FFA654
00FFFBF8F8FFFBFBFBFFFBFBFBFF919191E5000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000006060
6000BFBF7F009F9F2000000000003F3F00002020200007F7F7F005F5F5F000000
0000000000000000000000000000000000FFFFFF01FFFFFF01E4E4E3977F7D
7BFF807E7CFF7F7D7BFF7A7876FFF6F6F5FF5B5852FF414036FF736E5FFF6C67
58FFBDB9AF97DEDEDE21FFFFFF01FFFFFF01919191E5FDFDFDFFF9F9F9FFF9F9
F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F5F5FFF9F5F5FFA654
00FFF9F5F5FFF9F9F9FFF9F9F9FF919191E5000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000BFBF
BF009F9F6000DFDF20007F7F0000202020009F9F9F007F7F7F00000000000000
0000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
```

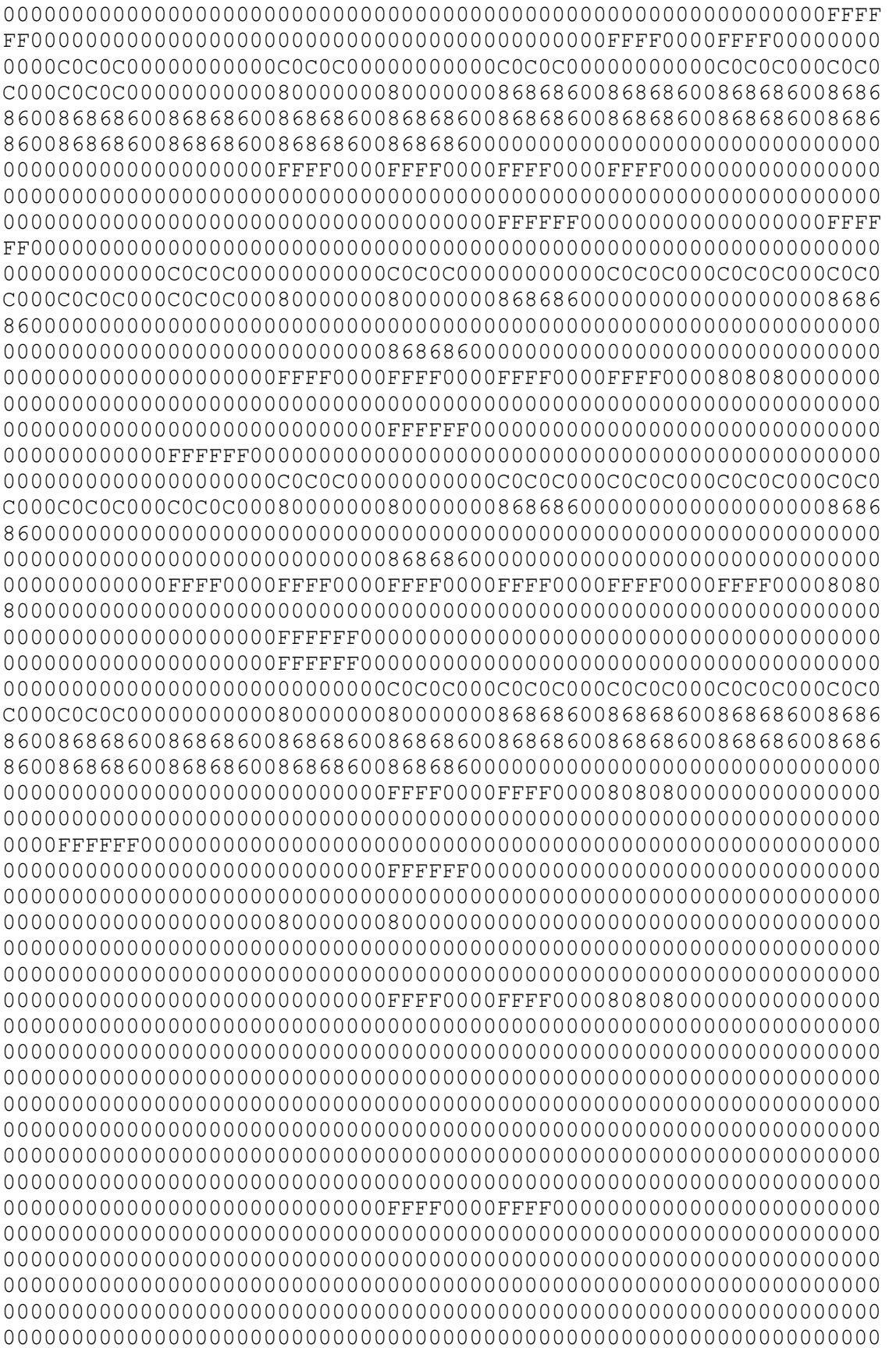
FF01FFFFFF01FFFFFF01FFFFFF017A7876FFF5F5F5FF565451FFB8B5B1FFB1AE
A8FF65625CFFB5B1A9B7E0E0E01FFFFFFF01919191E5FBFBFBFFC6C6C6FFBDBD
BDFFB5B5B5FFACACACFFA4A4A4FF9C9C9CFF959595FFF8F3F3FFF8F3F3FFA654
00FFF8F3F3FFF8F8F8FFF8F8F8FF919191E5000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000007F7F00007F7F00007F7F0000BFBF
4000DFDF5F00DFDF2000DFDF2000DFDF5F009F9F20007F7F00007F7F00003F3F
000000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF017D7B79FF777572FFF4F4F5FF575555FF6260
5EFFB1ADAAFF64615FFFA2A09EADFFFFFF01919191E5FBFBFBFFF7F7F7FFF7F7
F7FFF7F7F7FFF7F7F7FFF7F7F7FFF7F7F7FFF7F7F7FFF6F1F1FFF6F1F1FFF6F1
F1FFF6F1F1FFF7F7F7FFF7F7F7FF919191E5000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000202020009F9F
9F00DFDF5F009F9F20007F7F40009F9F5F000000000000000000000000000000
000000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF017F7D7CFFD6D4D3FF797775FF6E6C69FFFFFF
FF015F5D5BFFAFABA9FF666463FFFFFFFF01919191E5FAFAFAFFC7C7C7FFBDBD
BDFFB5B5B5FFACACACFFA4A4A4FF9C9C9CFF959595FF8F8F8FFF898989FF8585
85FF858585FF858585FFF5F5F5FF919191E5000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000202020009F9F9F009F9F
9F002020200007F7F000040404000000000000000000000000000000000000000000
000000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF017F7D7BFFEEEDECFF7F7D7AFFCCCCCC33EAEA
EA15BABABA99B2B0AEFF6A6866FFFFFFFF01919191E5FAFAFAFFF4F4F4FFF4F4
F4FFF4F4F4FFF4F4F4FFF4F4F4FFF4F4F4FFF4F4F4FFF4F4F4FFF4F4F4FFF4F4
F4FFF4F4F4FFF4F4F4FFF4F4F4FF919191E5000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000202020009F9F9F009F9F9F002020
2000000000007F7F000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01E4E4E397797775FFF5F5F4FF7E7B79FFB7B7
B7876C6A67FFB9B7B4FF6D6B6AFFFFFFFF01919191E5F9F9F9FFC6C6C6FFBEBE
BEFFB5B5B5FFACACACFFA4A4A4FF9C9C9CFF959595FF8F8F8FFF898989FF8585
85FF858585FF858585FFF3F3F3FF919191E5000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000808080009F9F9F00202020000000
0000000000007F7F000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01E1E1E0A77B7876FFEEEDECFFD5D3
D2FFC3C2C0FF73716EFFC3C2C197FFFFFF01919191E5FDFDFDFFF9F9F9FFF9F9
F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9F9FFF9F9
F9FFF9F9F9FFF9F9F9FFF9F9F9FF919191E5000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000002020200000000000000000000000000
0000000000007F7F000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000FFFFFF01FFFFFF01FFFFFF01FFFF
FF01FFFFFF01FFFFFF01FFFFFF01D0CFCE97817F7DFF7F7D
7CFF7C7A78FFC9C7C797FFFFFF01FFFFFF01A2A2A2C1919191E5919191E59191
91E5919191E5919191E5919191E5919191E5919191E5919191E59191
91E5919191E5919191E5919191E5A2A2A2C1000000000000000000000000000000

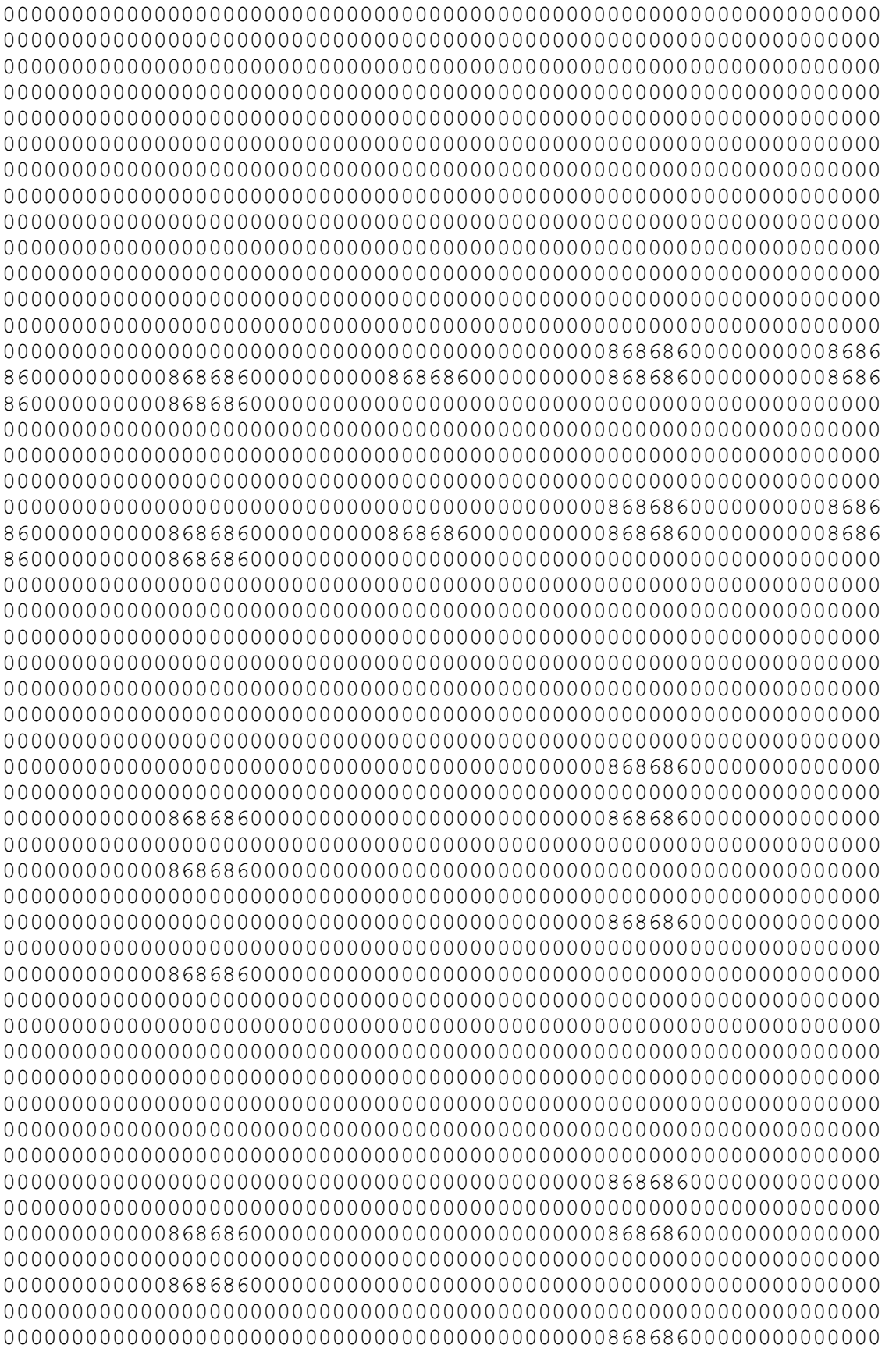
```
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0000000000000000000000000000000000000000
000000000000EFEFEF0080808000BFBFBF0000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000FF0000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF0000000000000000000000000000000000000000
0000FFFFFF0000000000FFFFFF0000000000000000000000000000000000000000
0000EFEFEF0070707000BFBFBF005F5F5F00BFBFBF0000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000FF0000
FF0000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0000000000000000000000000000000000EFEF
EF0070707000BFBFBF0000000000DFDFDF005F5F5F00BFBFBF0000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0000000000000000000000000000000000000000
00000000000000000000000000FF000000FF000000FF000000FF000000FF000000
FF000000FF0000000000000000000000007F7F7F0000000000000000000000
00000000000000000000000000000000000000000000000007F7F7F00FFFFFF000000
00000000000000000000000000000000000000000000000000EFEFEF007070
7000BFBFBF0000000000BFBFBF00DFDFDF00DFDFDF005F5F5F00BFBFBF000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF0000000000000000FFFFFF000000
00000000000000000000FFFFFF0000000000000000FFFFFF00FFFFFF00FFFF
FF00FFFFFF0000000000000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF0000000000000000000000000000FFFF0000FFFF0000FF
FF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000000000FFFFFF000000
0000FFFFFF00FFFFFF0000000000000000000000000000EFEFEF0070707000BFBF
BF0000000000BFBFBF0000000000BFBFBF00DFDFDF00DFDFDF005F5F5F00BFBF
BF0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0000000000000000FFFFFF00FFFFFF00FFFF
FF00FFFFFF0000000000000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF0000000000000000FFFF007F7F7F007F7F
7F007F7F7F007F7F7F007F7F7F0000FFFF0000FFFF0000000000FFFFFF000000
0000FFFFFF0000000000000000000000000000000070707000BFBFBF000000
0000BFBFBF003F3F3F007F7F7F0000000000DFDFDF00DFDFDF00DFDFDF005F5F
5F0000000000000000000000000000000000000000000000000000000000000000
FF00000000000000000000000000FFFFFF0000000000BFBFBF00000000000000
0000FFFFFF0000000000FFFFFF0000000000000000FFFFFF00000000000000
```

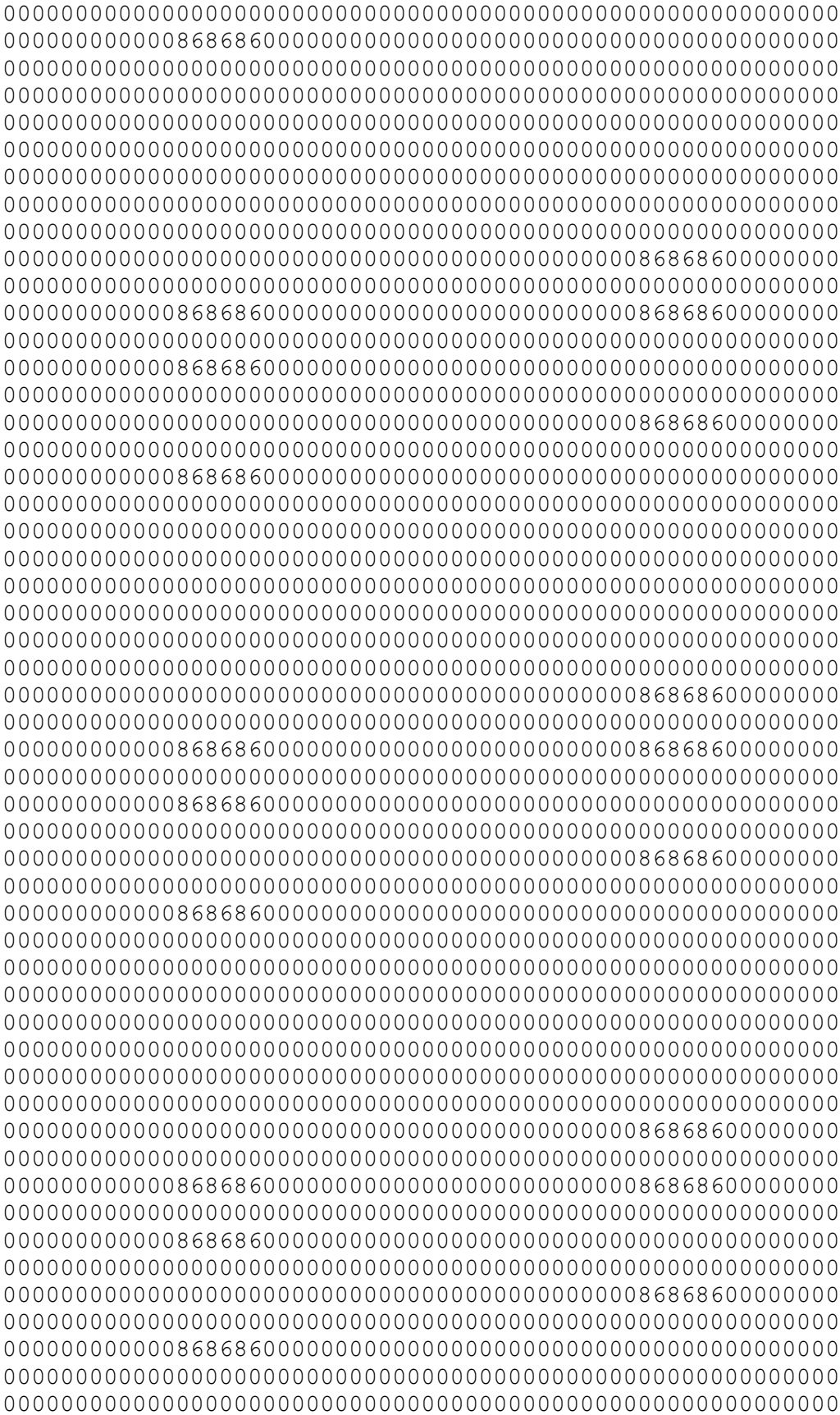
0000FFFFFF0000000000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF0000000000000000000000000000FFFF007F7F7F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF0000000000FFFFFF000000
00000000000000000000000000000000000BFBFBF00BFBFBF00DFDFDF00DFDF
DF00DFDFDF007F7F7F00BFBFBF0000000000000000000DFDFDF00BFBFBF00BFBF
BF00000000000000000000000000000000000000000000000000000000000000
FF000000FF0000000000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0000000000000000000FFFFFF00FFFFFF00FFFF
FF00FFFFFF0000000000000000FF000000FF000000FF000000FF000000FF000000
FF000000FF0000000000000000000000000000000000000FFFF007F7F7F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF0000000000000000000000
00000000FF0000000000000000000000000007F7F7F0000000000DFDFDF00BFBF
BF00DFDFDF00000000007F9F9F003F9F9F00BFDFDF00BFBFBF00BFBFBF000000
00000000000000000000000000000000000000000000000000FF000000FF000000
FF000000FF000000FF0000000000FFFFFF0000000000000000000FFFFFF000000
00000000000000000000000000000000000000000000000000FFFFFF00000000000000
0000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000000000000000FF000000
FF0000000000000000000000000000000000000000000000FFFF007F7F7F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF0000000000000000000000
FF000000FF000000FF0000000000000000000000000007F7F7F00DFDFDF00DFDF
DF00BFBFBF00DFDFDF00DFDFDF00009F9F0000BFBF009F9F9F0000000000000000
00000000000000000000000000000000000000000000000000FF000000FF000000
FF000000FF000000FF0000000000FFFFFF00FFFFFF00FFFFFF000000
0000FFFFFF00FFFFFF0000000000000000000000000000FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000000000000000FF000000
00000000000000000000000000000000000000000000000000FFFF007F7F7F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF0000000000000000FF000000
FF000000FF000000FF0000000000000000000000000000007F7F7F000000
0000DFDFDF00BFBFBF00000000009F9F9F0000BFBF0000BFBF00BFBFBF000000
00000000000000000000000000000000000000000000000000FF000000FF000000
FF000000FF000000FF000000FF0000000000BFBFBF00FFFFFF000000
0000FFFFFF0000000000000000000000000000000000FFFFFF00000000000000
0000FFFFFF0000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFF007F7F7F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000007F7F
7F00DFDFDF00DFDFDF00BFBFBF00BFBFBF00BFBFBF0000BFBF0000BFBF00BFBF
BF00000000000000000000000000000000000000000000FF000000FF000000
FF000000FF000000FF0000000000FFFFFF00FFFFFF00FFFFFF000000
00000000000000000000000000000000000000000000FFFFFF00FFFFFF00FFFF
FF00FFFFFF0000000000FFFFFF00FFFFFF0000000000000000000000000000
00000000000000000000000000000000000000000000FFFF007F7F7F00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF0000000000000000000000
FF000000FF000000FF0000000000000000000000000000000000000000000000
00007F7F7F00BFBFBF00BFBFBF0000000000000000000BFBFBF0000BFBF0000BF
BF00BFBFBF00000000000000000000000000000000000000FF000000FF000000
FF000000FF000000FF0000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFFFF0000000000BFBF
BF00FFFFFF0000000000FFFFFF0000000000000000000000000000000000000000
00000000000000000000000000000000000000000000007F7F7F0000000000000
00007F7F7F007F7F7F0000000000000000007F7F7F0000000000000000000000
FF000000FF000000FF0000000000000000000000000000000000000000000000

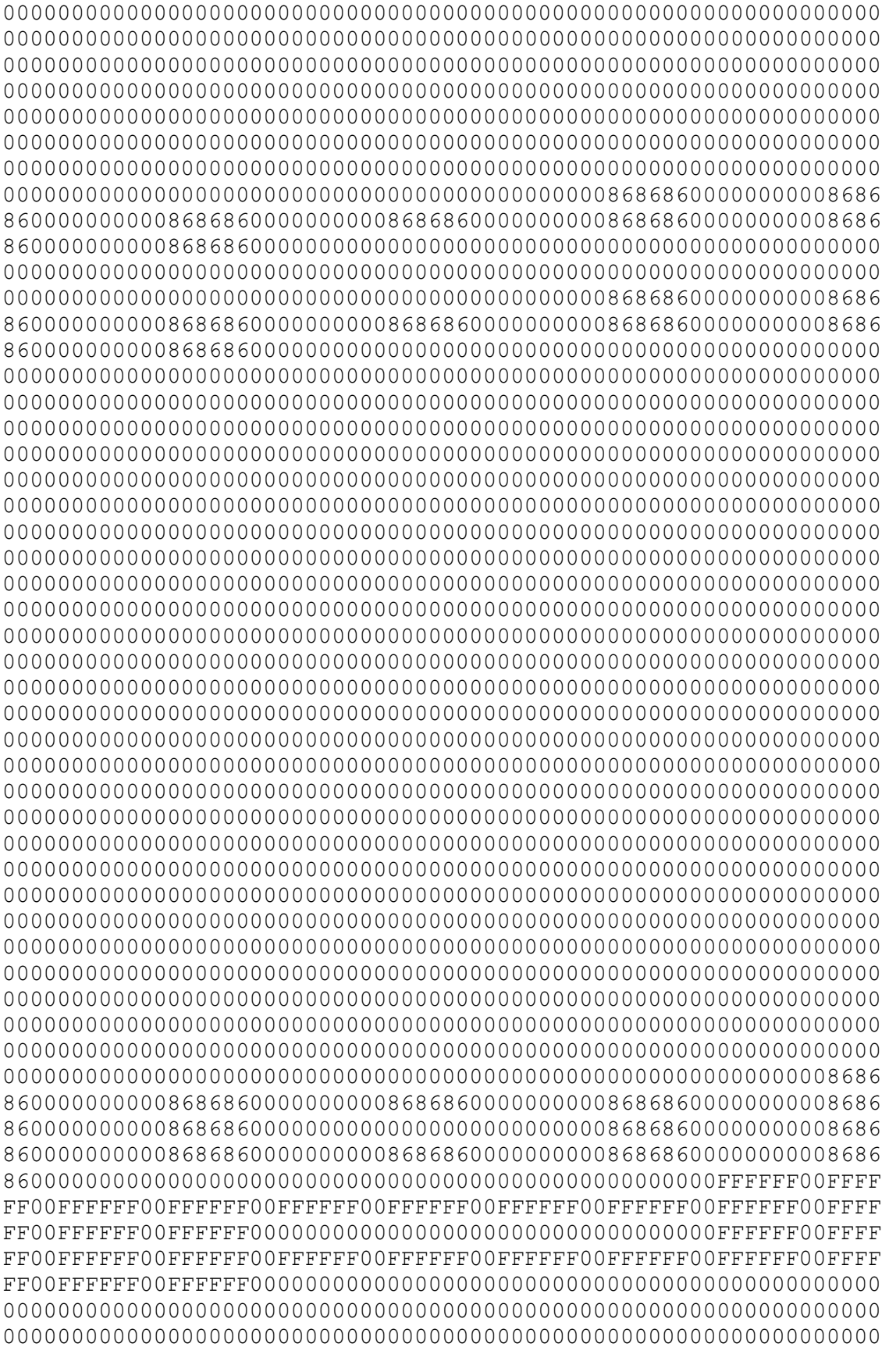
```
000000000000BFBFBF0000000000000000000000000000000000BFBFBF0000BF
BF0000BFBF00BFBFBF0000000000000000000000000000000000000000000000
FF000000FF0000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFFFF00FFFFFF00FFFF
FF00FFFFFF0000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFF007F7F7F007F7F
7F0000000000000000007F7F7F007F7F7F0000FFFF0000000000000000000000
FF000000FF000000FF0000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000BFBF
BF0000BFBF003F7F7F00BFBFBF0000000000000000000000000000000000000
FF0000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFF0000FFFF007F7F
7F0000000000000000007F7F7F0000FFFF0000FFFF000000FF000000FF000000
FF000000FF0000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
0000BFBFBF003F3F3F0000007F00BFBFBF0000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000007F7F7F0000000000000000000000
00000000000000000000000000000000000000007F7F7F0000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
000000000000BFBFBF0000007F00BFBFBF0000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000808080008080800808080
80008080800080808000808080008080800080808000808080008080800808080
80008080800080808000808080008080800000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000086868600868686008686860086868
86008686860086868600868686008686860086868600868686008686860086868
86008686860086868600868686008686860000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000080808000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFFFF0000000000FFFFFF00FFFFFF00FFFF
FF00C0C0C000FFFFFF00FFFFFF00C0C0C000FFFFFF00FFFFFF00FFFFFF000000
00000000000000000000000000000000000000000000868686000000000000000000008686
86000000000000000000000000000000868686000000000000000000000000008686
86000000000000000000000000000000868686000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000080808000000000000000000000000000
00000000000000FFFFFF0000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFFFF00FFFFFF00FFFF
FF00C0C0C000FFFFFF00FFFFFF00C0C0C000FFFFFF00FFFFFF00FFFFFF000000
00000000000000000000000000000000000000000000868686000000000000000000008686
```

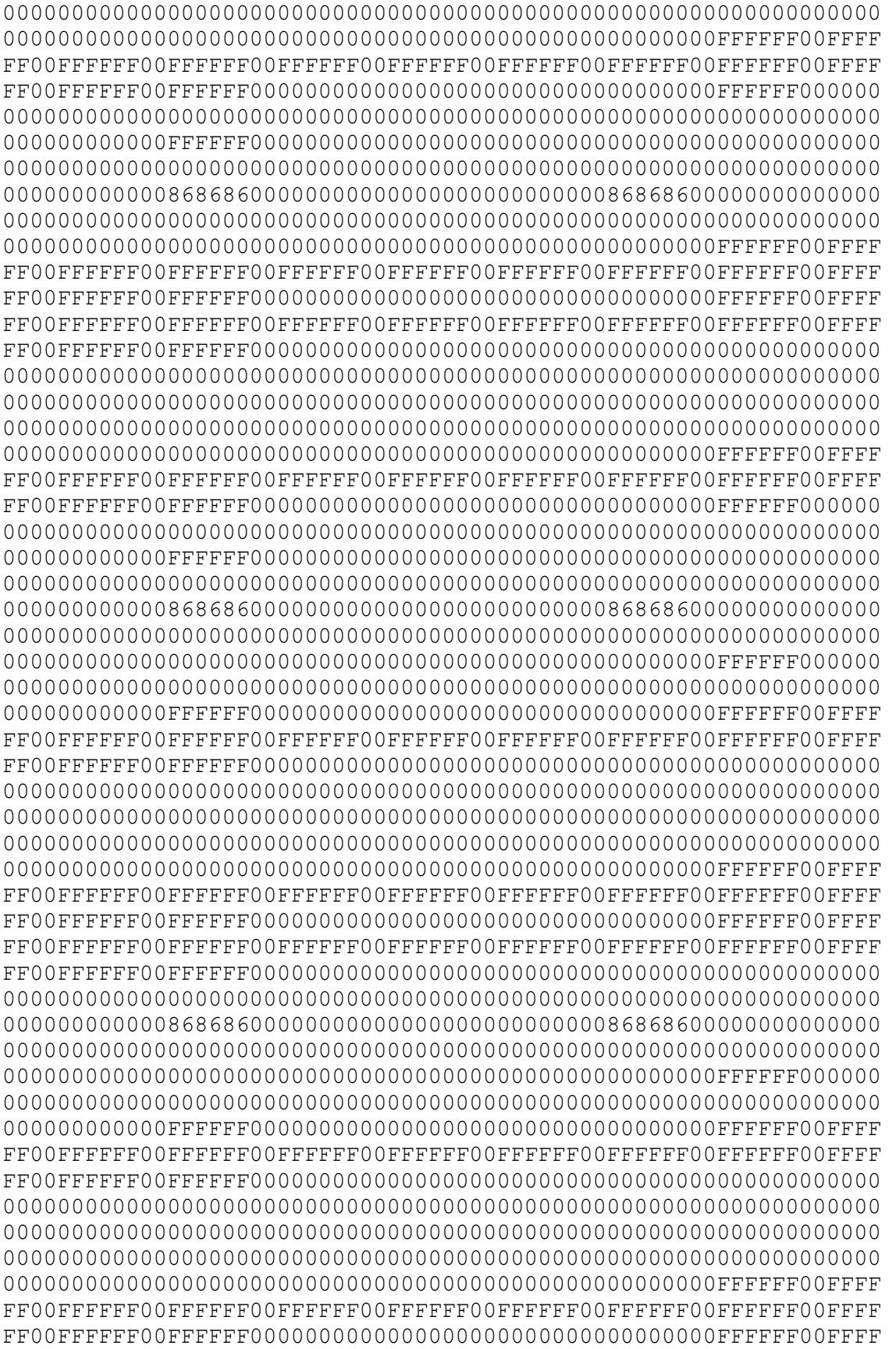




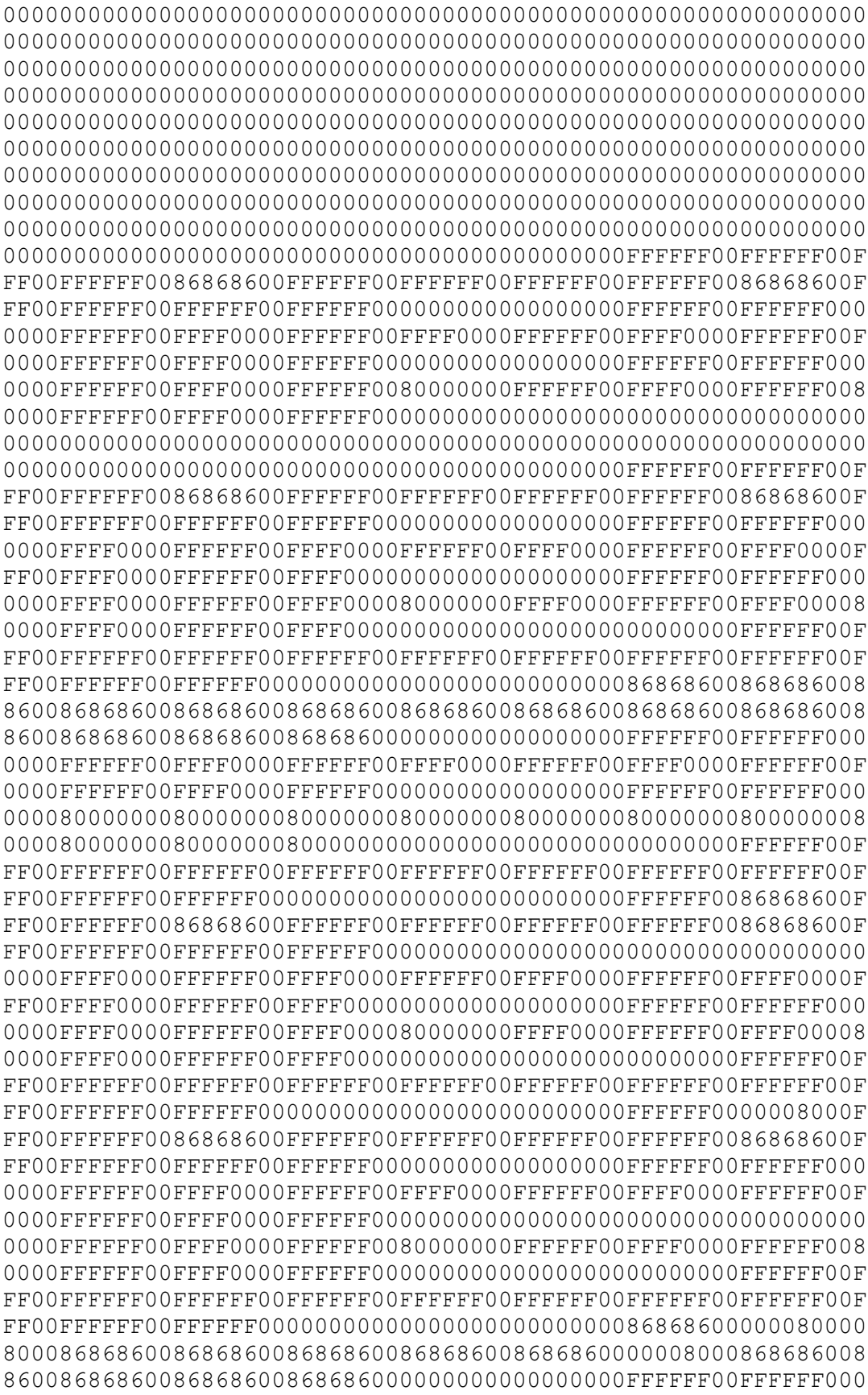


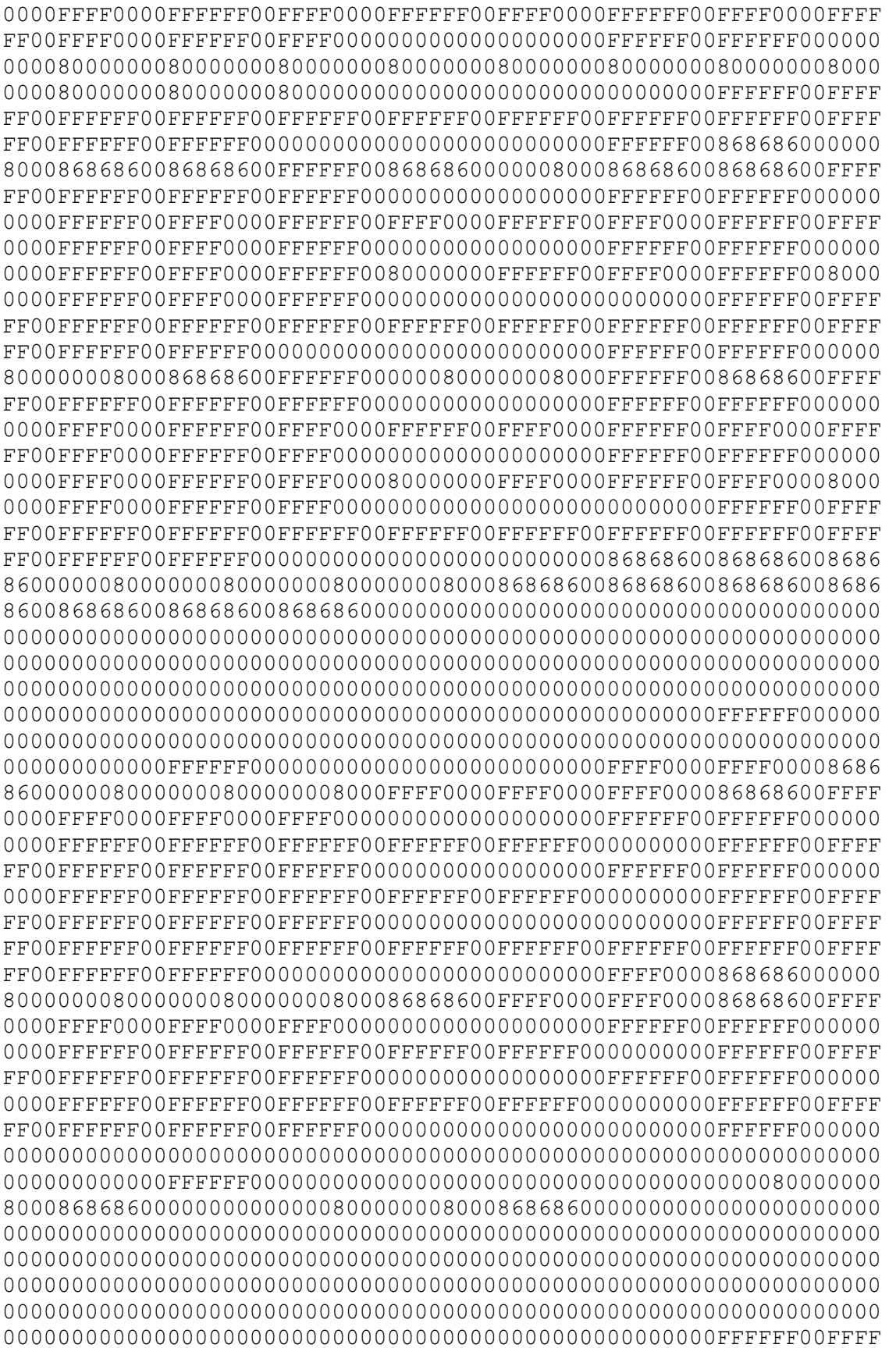




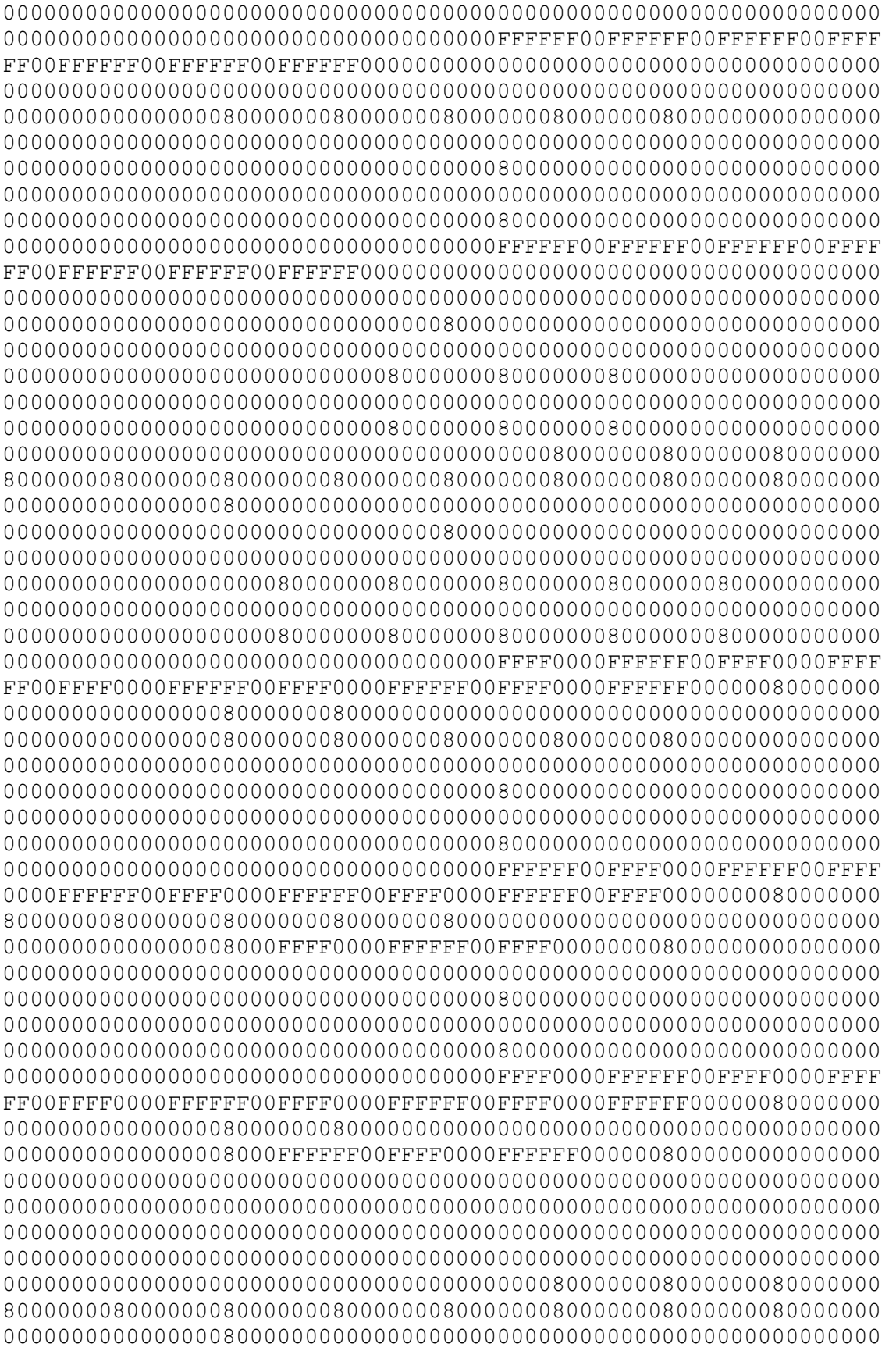


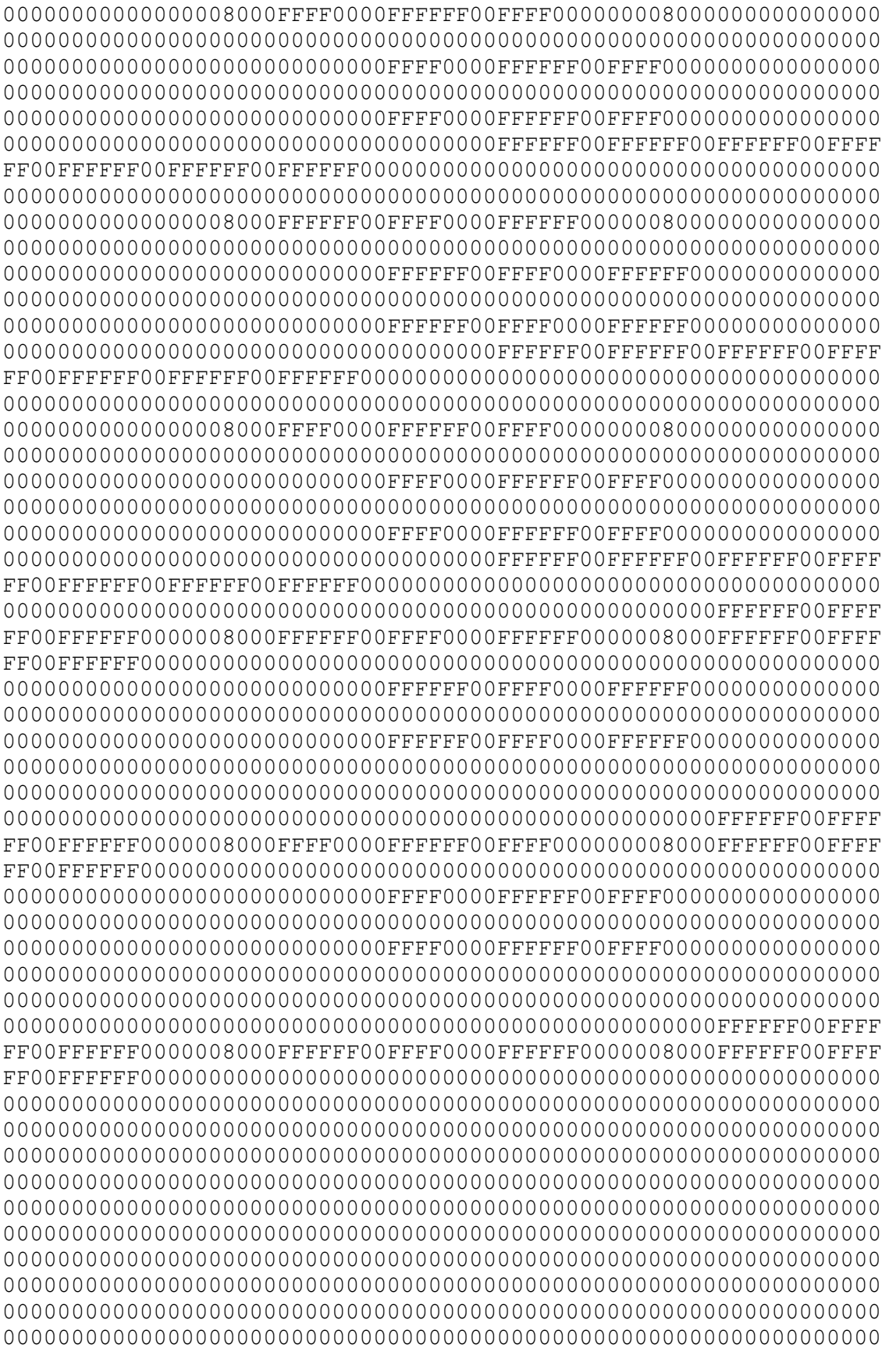
```
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000008686860000000000000000000000000086868600000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000008686860000000000000000000000000086868600000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000008686
8600000000008686860000000000868686000000000086868600000000008686
8600000000008686860000000000000000000000000086868600000000008686
8600000000008686860000000000868686000000000086868600000000008686
8600000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

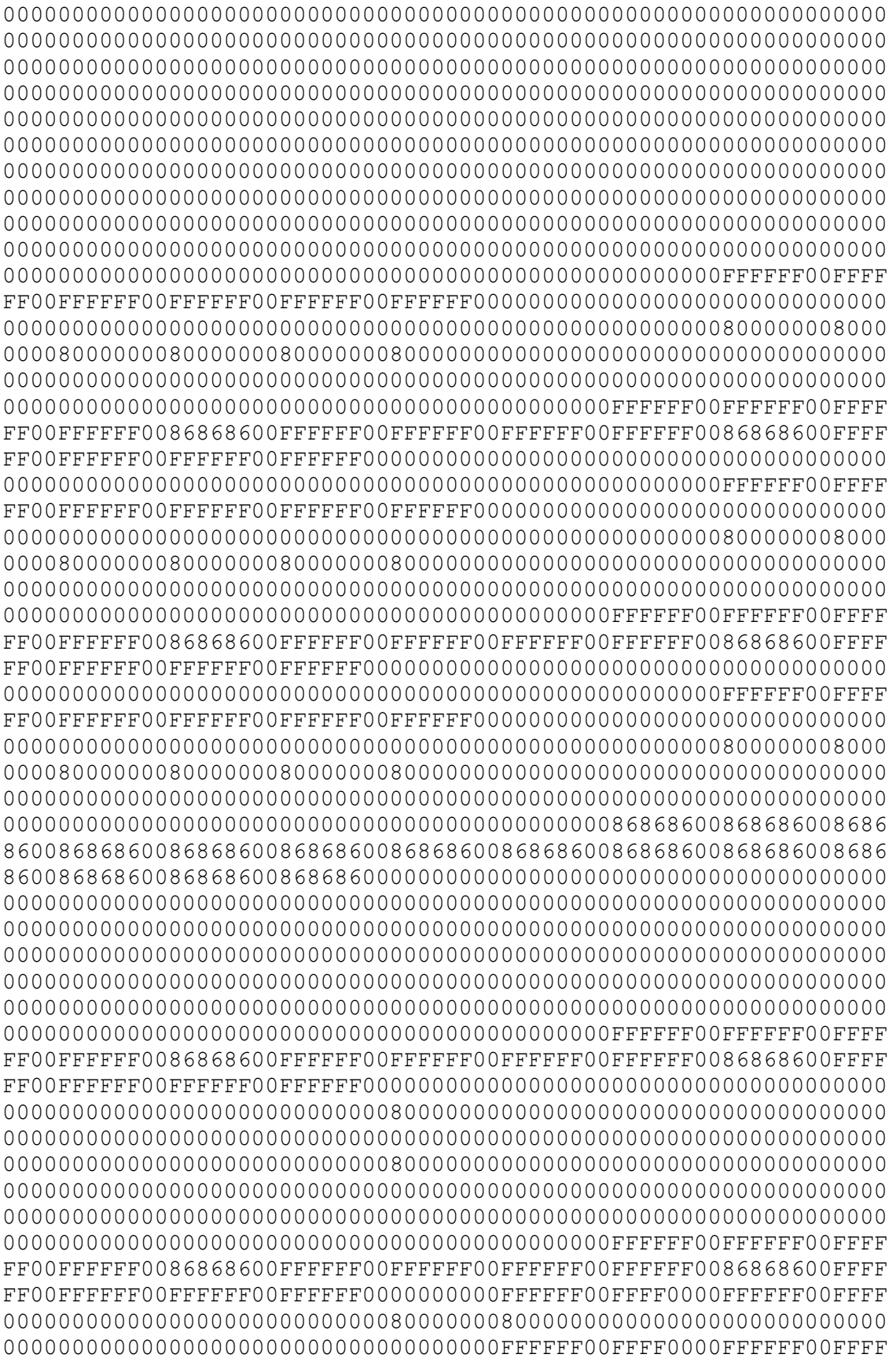


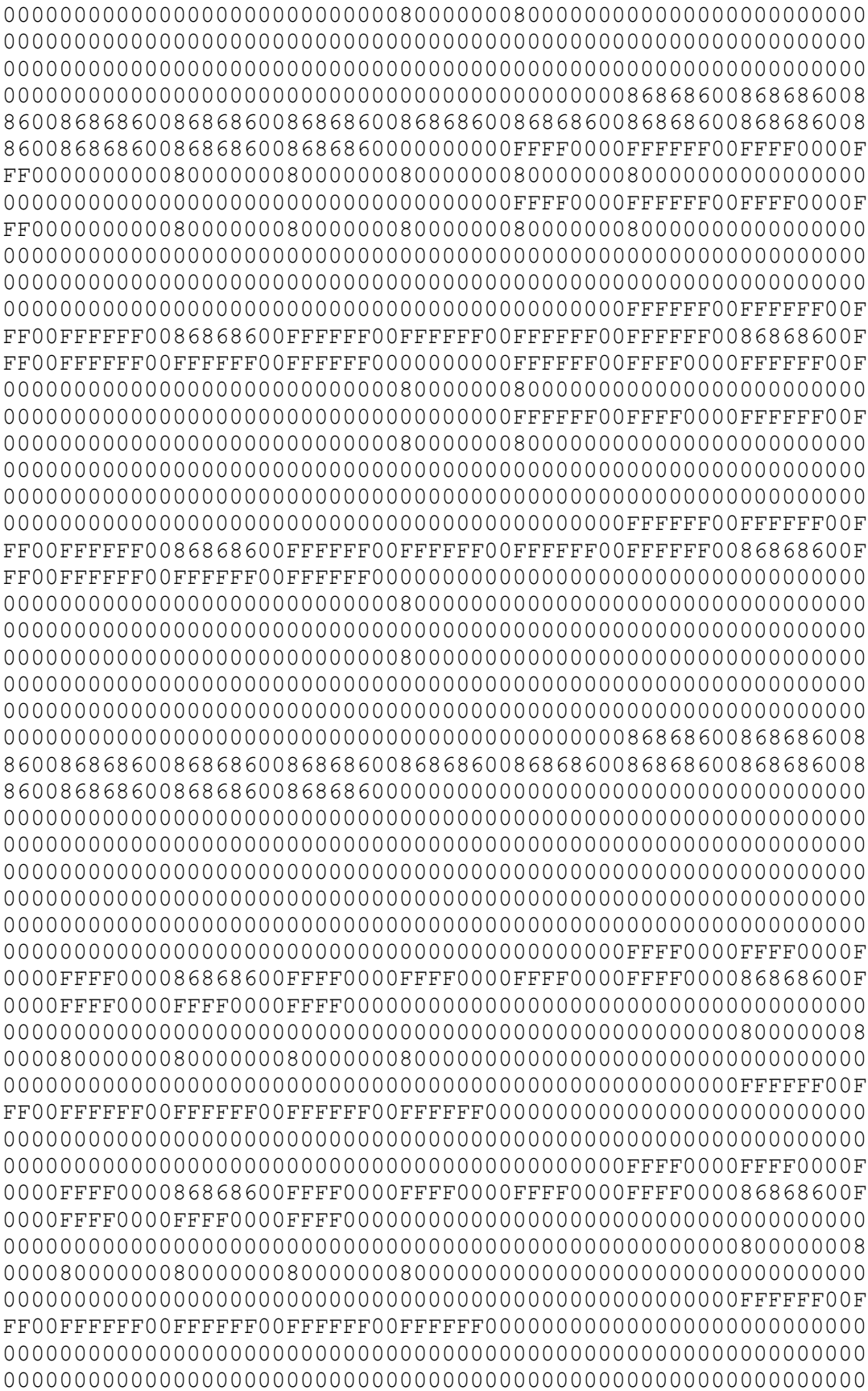


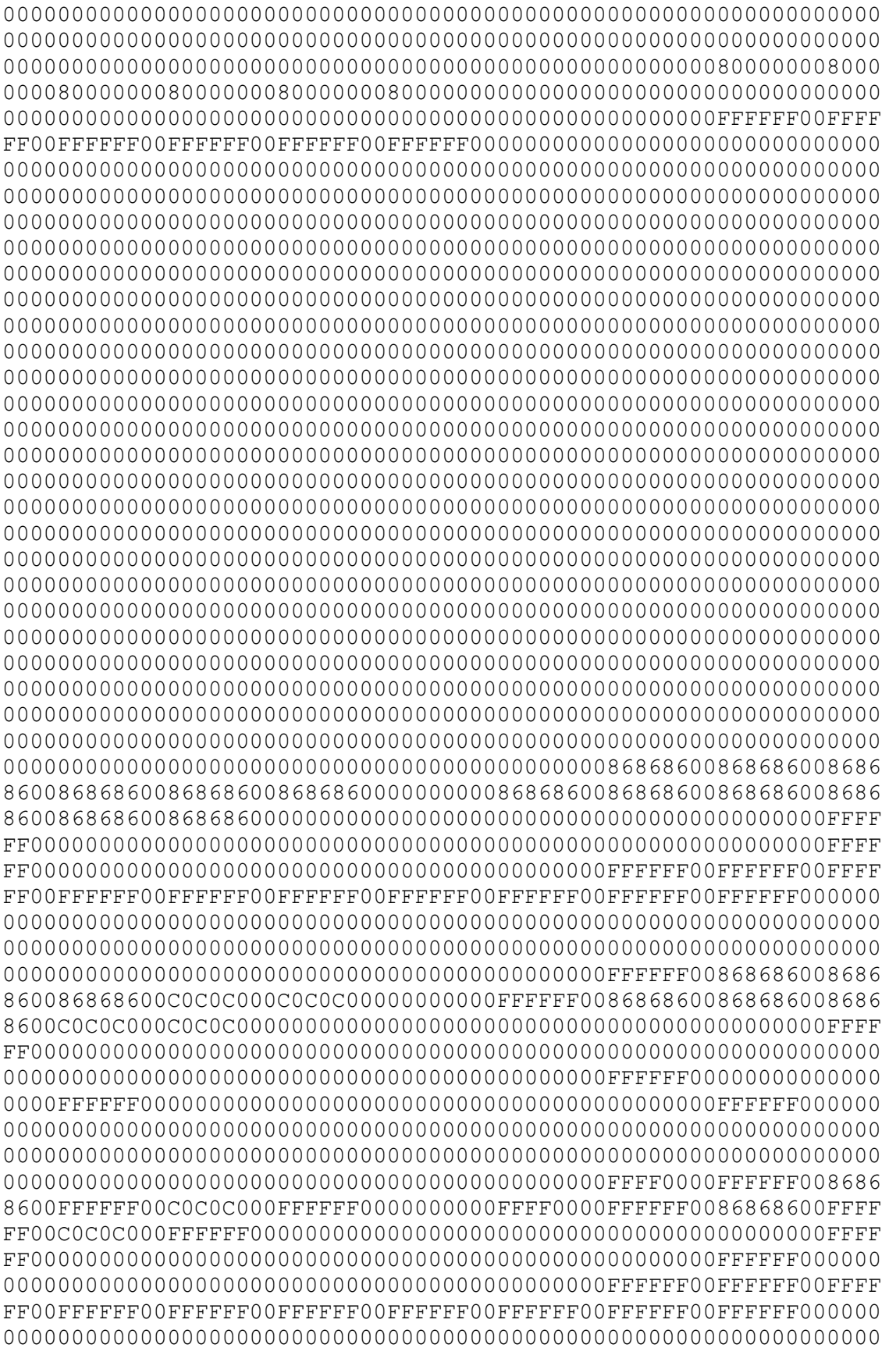
```
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000080000000800086868600000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000800000008000868686000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000008000
0008000000008000000000000000000000000000000000000000000000008000
000800000000800000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000008000
0008000000008000000000000000000000000000000000000000000000008000
000800000000800000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000080000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000008000
0008000000008000000000000000000000000000000000000000000000008000
000800000000800000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF000000000000000000FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000008000000080000000800000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

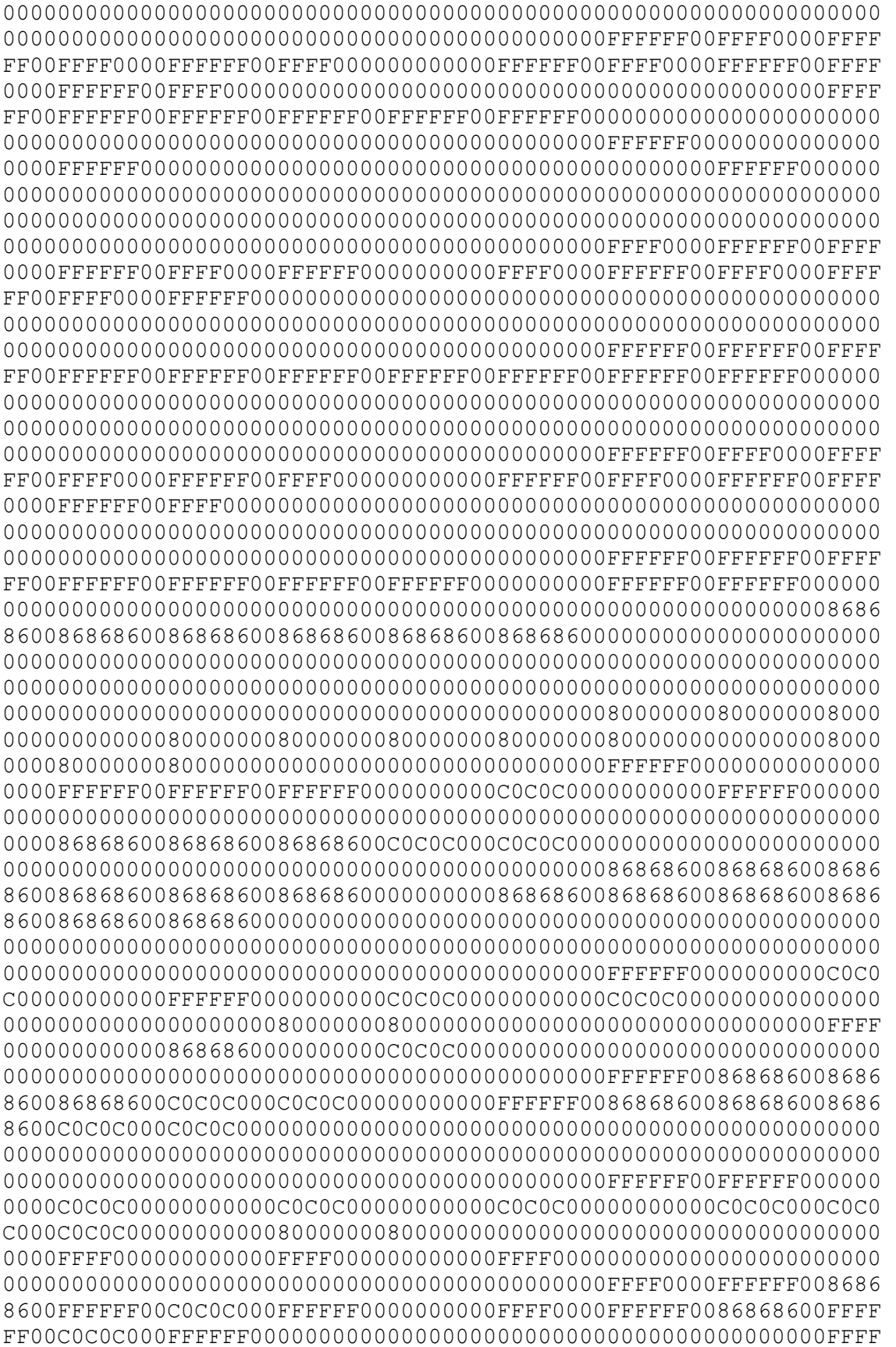


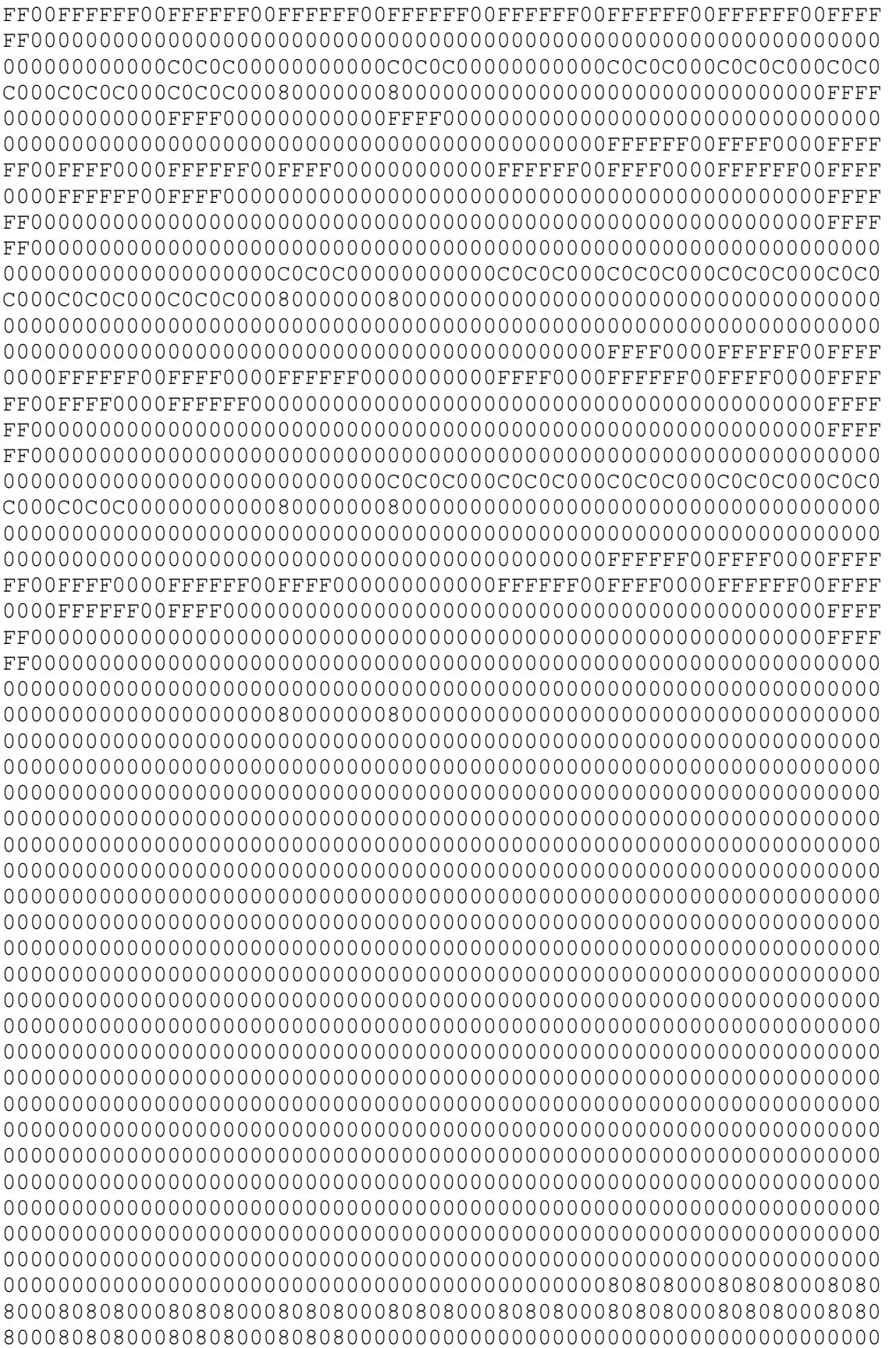


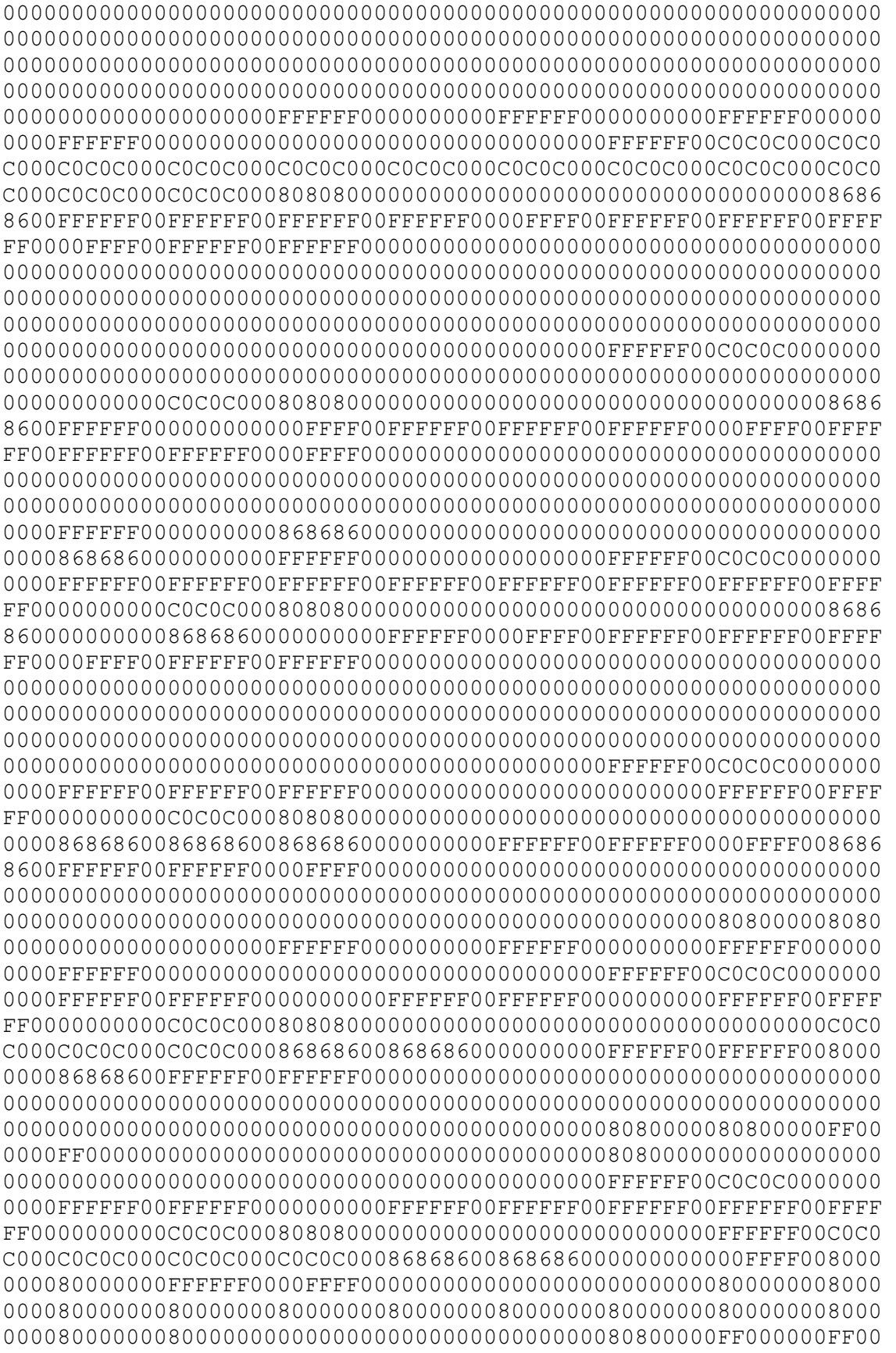




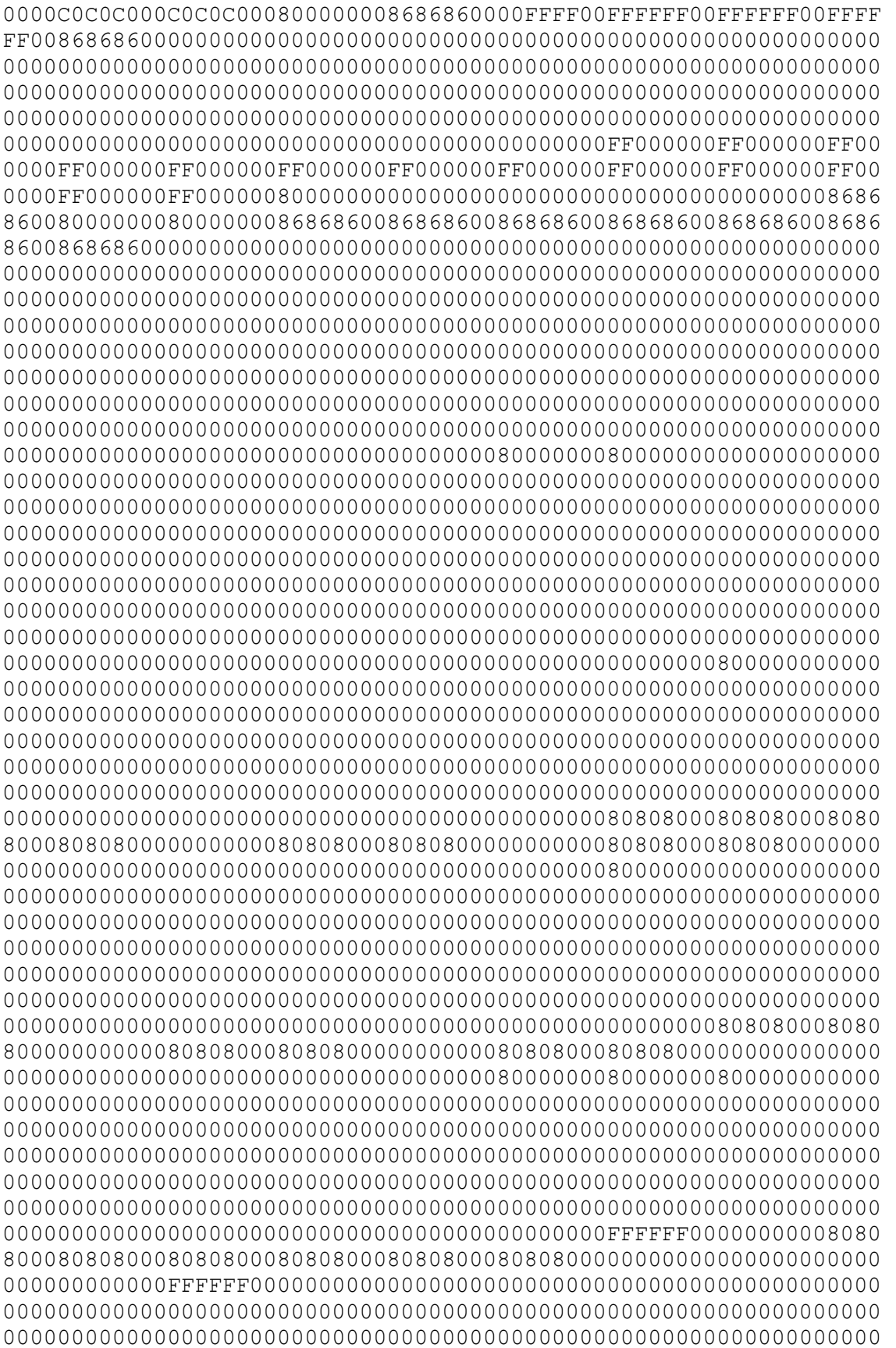


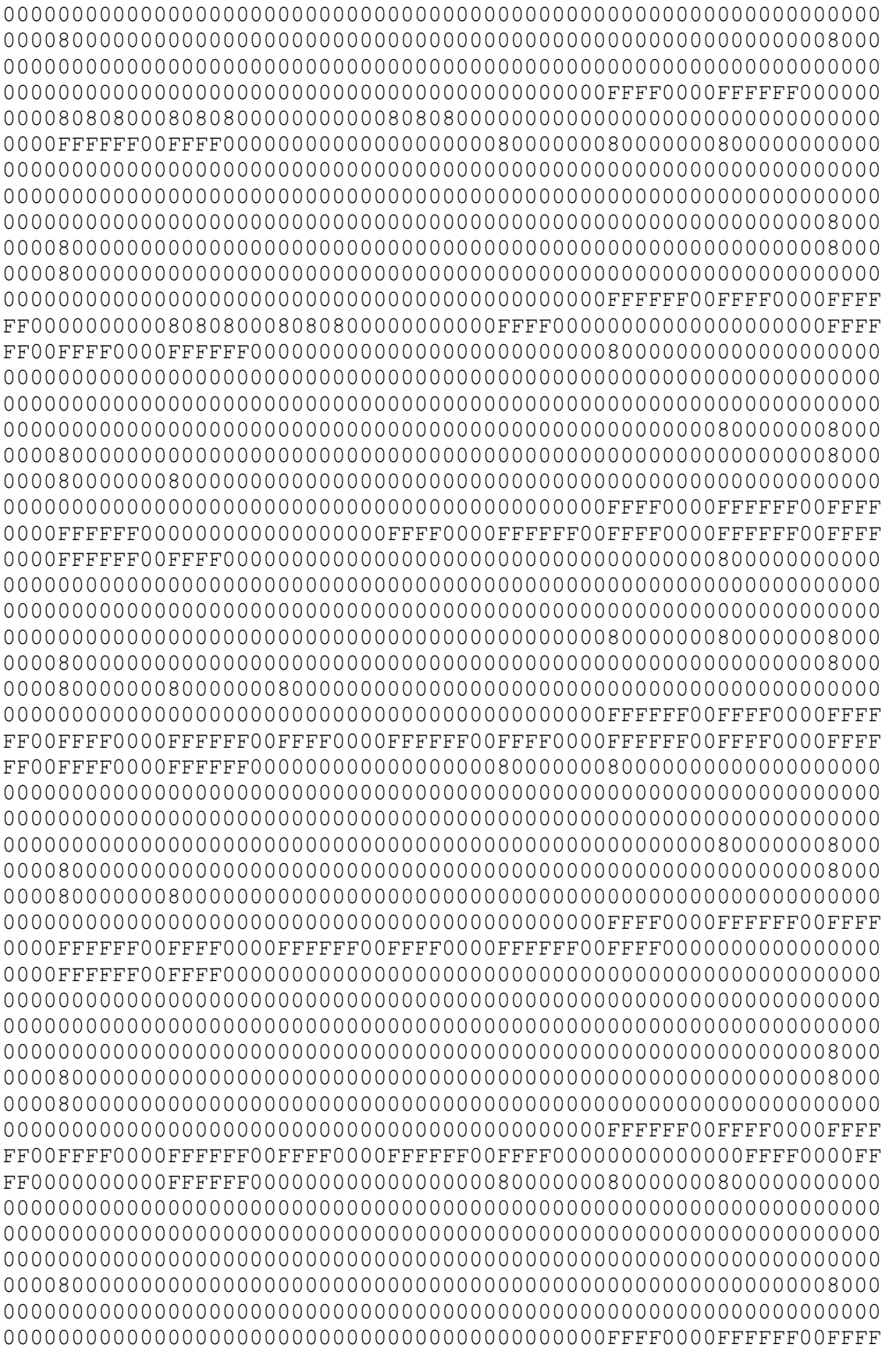


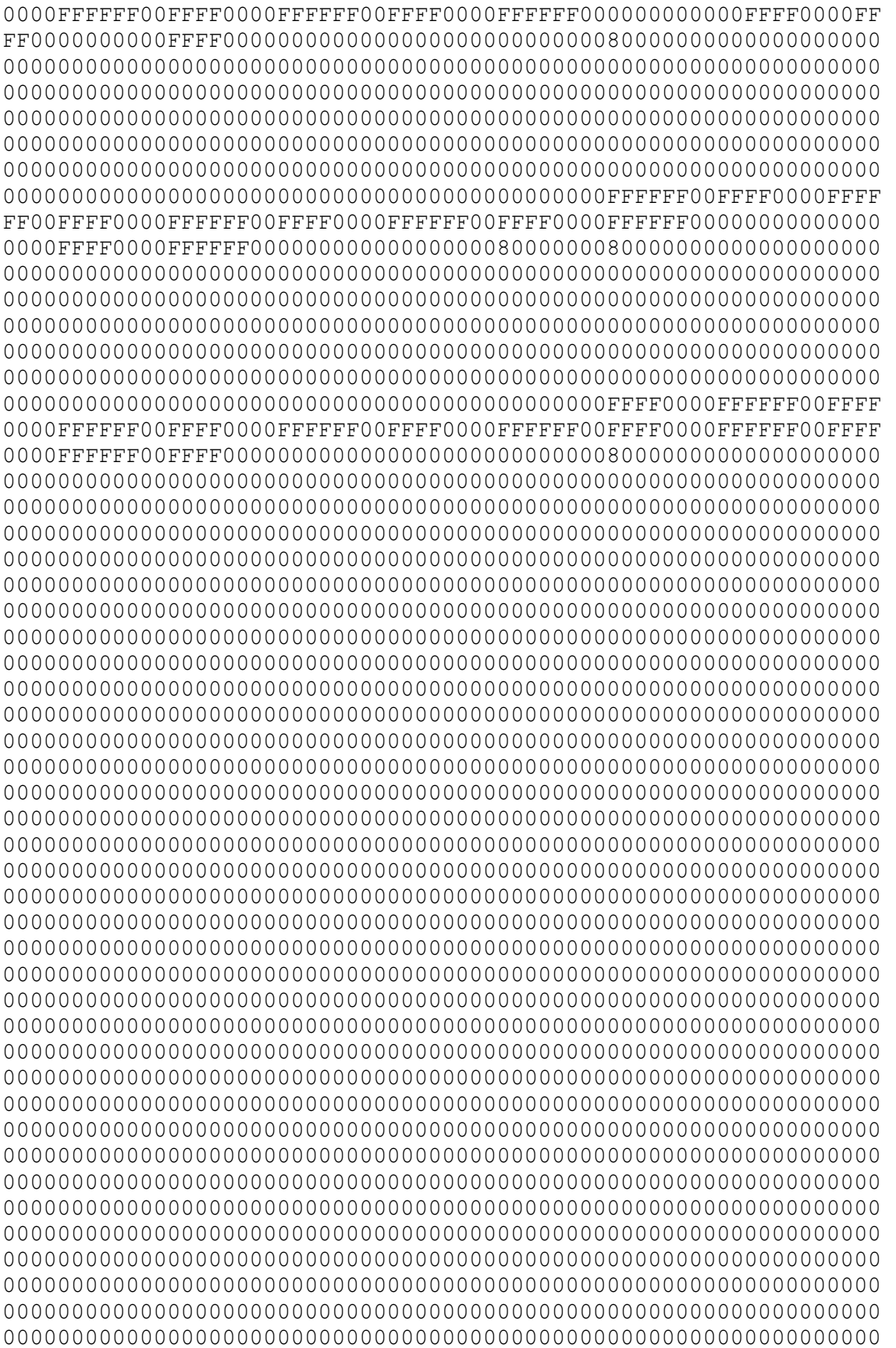


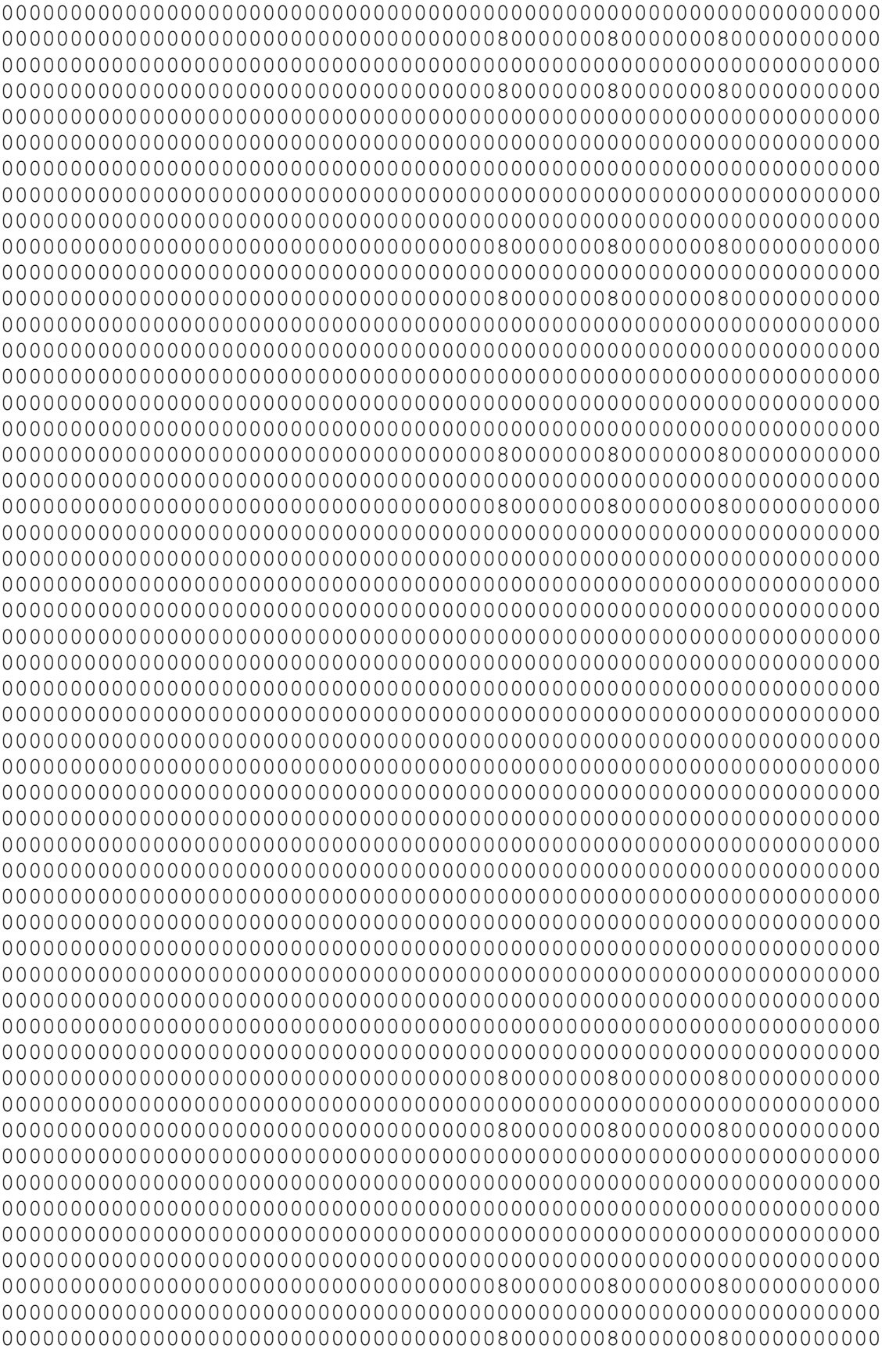

0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000FFFFFF0000000000FFFFFF0000000000FFFFFF000000
0000FFFFFF0000000000000000000000000000000000FFFFFF00C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0
C000C0C0C000C0C0C000808080000000000000000000000000000000000008686
8600FFFFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFF
FF0000FFFF00FFFFFF00FFFFFF0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000000000000000000000000000000000000000000000000000000000000000000000
000000000000C0C0C000808080000000000000000000000000000000000008686
8600FFFFFF000000000000FFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF00FFFF
FF00FFFFFF00FFFFFF0000FFFF0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000FFFFFF0000000000868686000000000000000000000000000000000000000000
0000868686000000000000FFFFFF0000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000000000000000000008686
8600000000008686860000000000FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFF
FF0000FFFF00FFFFFF00FFFFFF0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF00FFFFFF0000000000000000000000000000FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000000000000000000000C0C0
C000C0C0C000C0C0C000868686008686860000000000FFFFFF00FFFFFF008000
000086868600FFFFFF00FFFFFF0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000080800000808000000FF00
0000FF0000000000000000000000000000000000000080800000000000000000
0000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000FFFFFF00C0C0
C000C0C0C000C0C0C000C0C0C000868686008686860000000000FFFF008000
0000800000000FFFFFF0000FFFF0000000000000000000000000800000008000
0000800000008000000080000000800000008000000080000000800000008000
0000800000008000000000000000000000000000080800000FF000000FF00

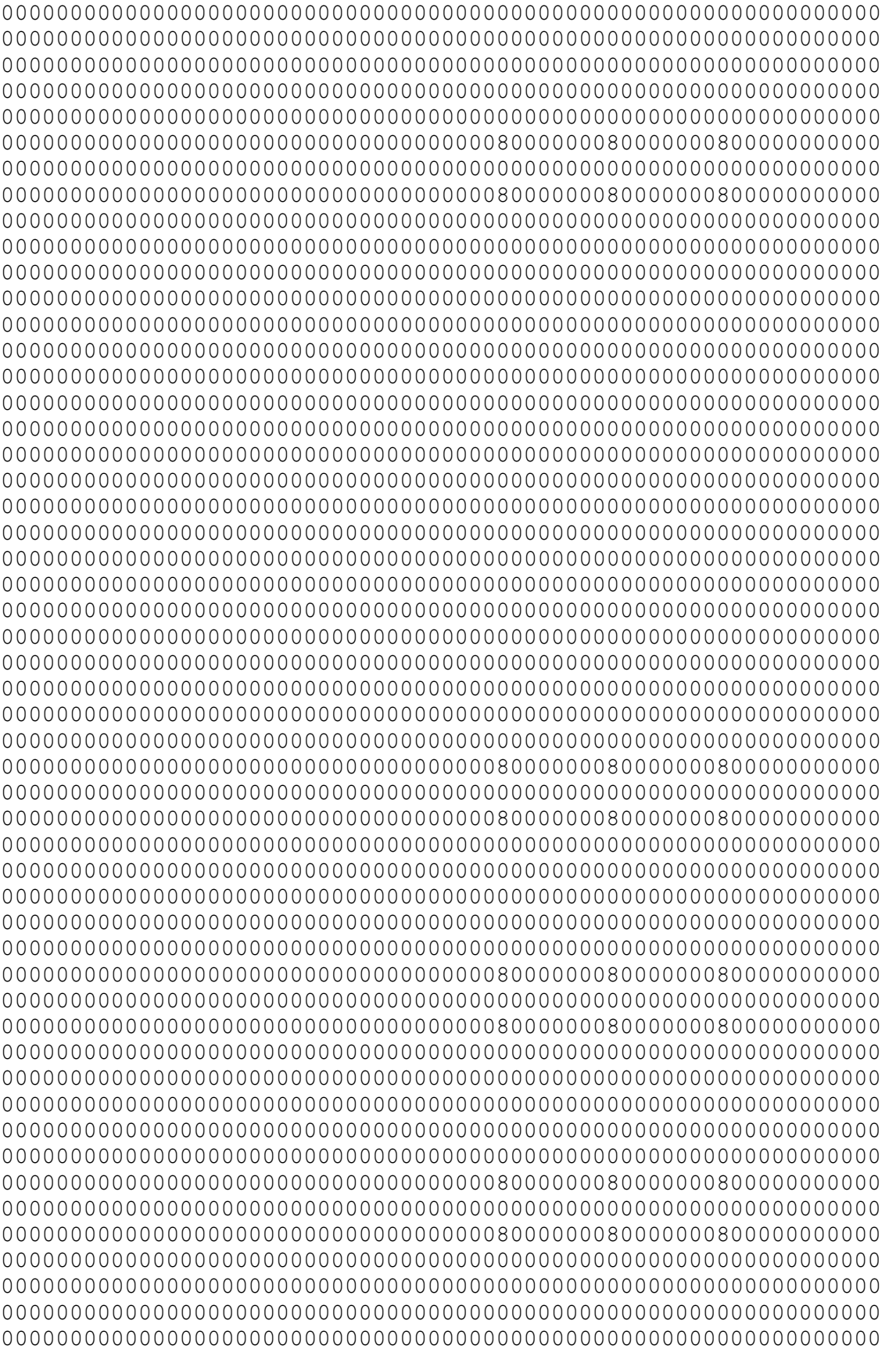
0000FF0000000080000000800000008000000080000086868600808000000000
00000000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF00FFFFFF0000000000000000000000FFFFFF00FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000000C0C0C000FFFFFF00FFFF
FF00C0C0C000C0C0C000C0C0C000C0C0C000868686008686860080000000800000
00008000000FFFFFF00FFFFFF0000000000000000000000000000800000008000
00008000000080000000800000008000000080000000800000008000000080000
00008000000080000000000000000000000000000000FF000000FF0000000080
00000800000008000000080000008000000080000FF000000808000000000
00000000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF0000000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000000000C0C0C000FFFF
FF00FFFFFF00C0C0C00000000000C0C0C000C0C0C000868686008000000080000
00008000000FFFFFF0000FFFF0000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FF0000000080000000800
00000800000008000000080000008000000086868600800000008080000000000
00000000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF0000000000000000000000000000000000FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000000000000000000000C0C0
C000FFFFFF0000000000800000000000000C0C0C000800000008000000008000
00086868600FFFFFF00FFFFFF0000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FF000000FFFF00000080
00000800000800000000080000000800000FF000000008000008080000000000
00000000000000000000000000000000000000000000FFFFFF00C0C0C0000000
0000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF0000000000C0C0C000808080000000000000000000000000000000000000000000
0000C0C0C000FFFFFF0080000000C0C0C000868686008000000080000000868 6
8600FFFFFF00FFFFFF00FFFFFF0000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000868686 00FF000000FFFF
FF00FFFF0000FF000000FF000000FF000000FF000000008000008080000000000
00000000000000000000000000000000000000000000FFFFFF00C0C0C0000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000C0C0C000808080000000000000000000000000000000000008000
00000000000C0C0C000800000008686860000000008686 600FFFFFF00FFFF
FF00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000FFFFFF00FFFF
0000FFFFFF0000800000008000000080000000800000808000000000000000000
00000000000000000000000000000000000000000000FFFFFF00C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0
C000C0C0C000C0C0C000808080000000000000000000000000000000000000008000
0000C0C0C0000000000080000000000000086868600FFFFFF0000FFFF00FFFF
FF0086868600C0C0C000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000008686
860086868600FFFF0000008000000080000080800000000000000000000000000000
00000000000000000000000000000000000000000000FF000000800000008000
00008000000080000000800000008000000080000000800000008000000080000
00008000000080000000800000000000000000000000000000000000000008000



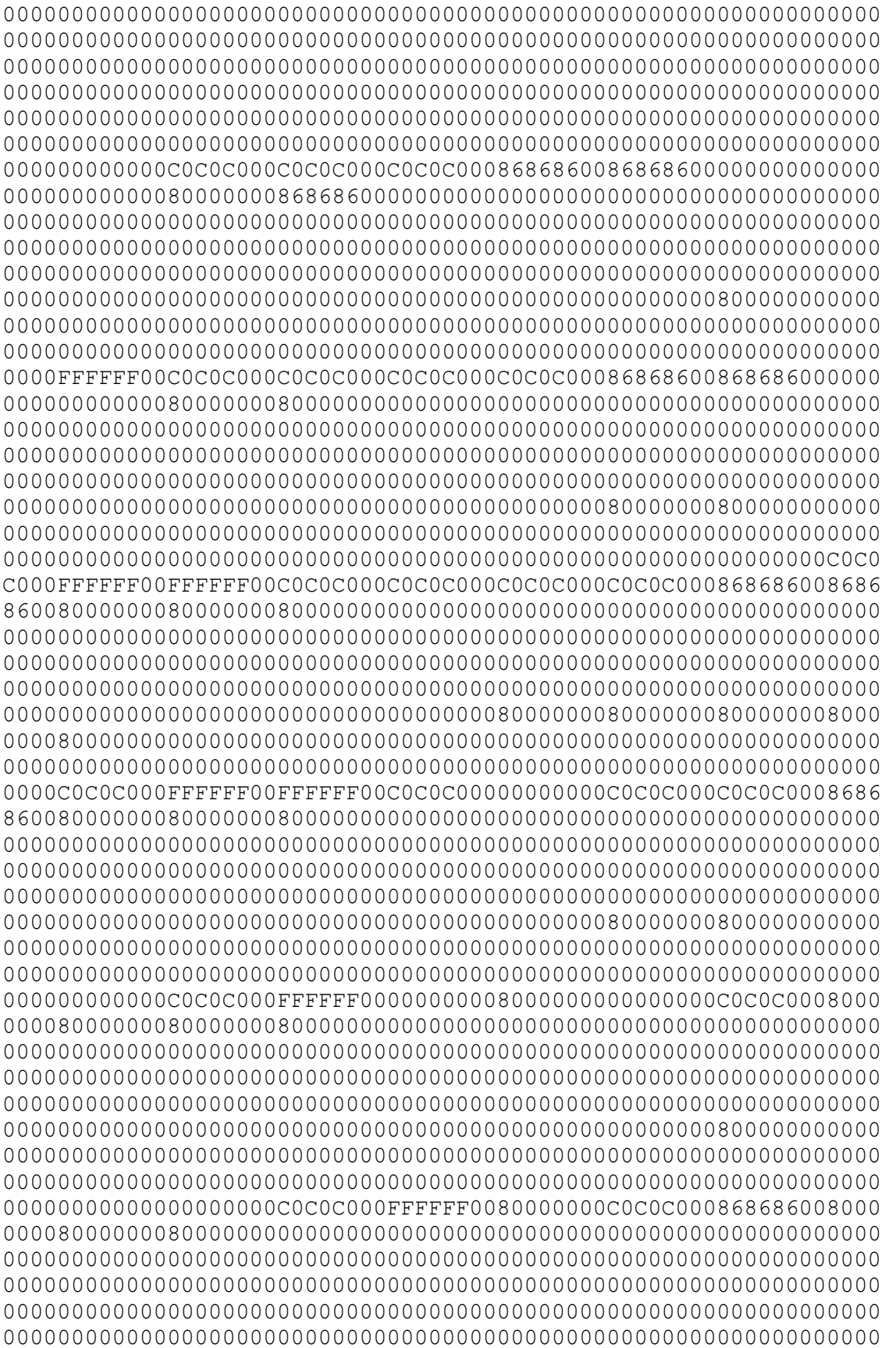


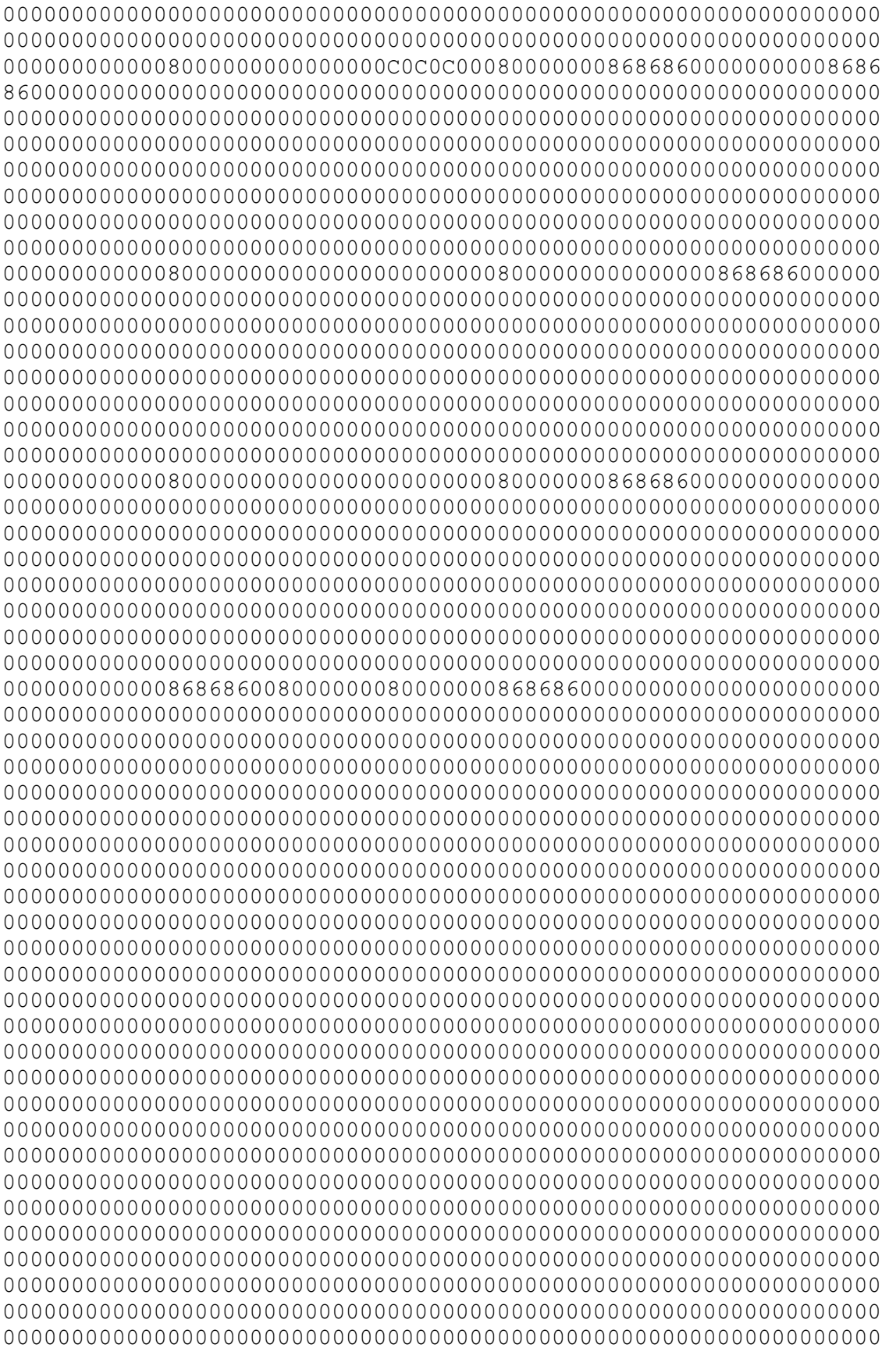


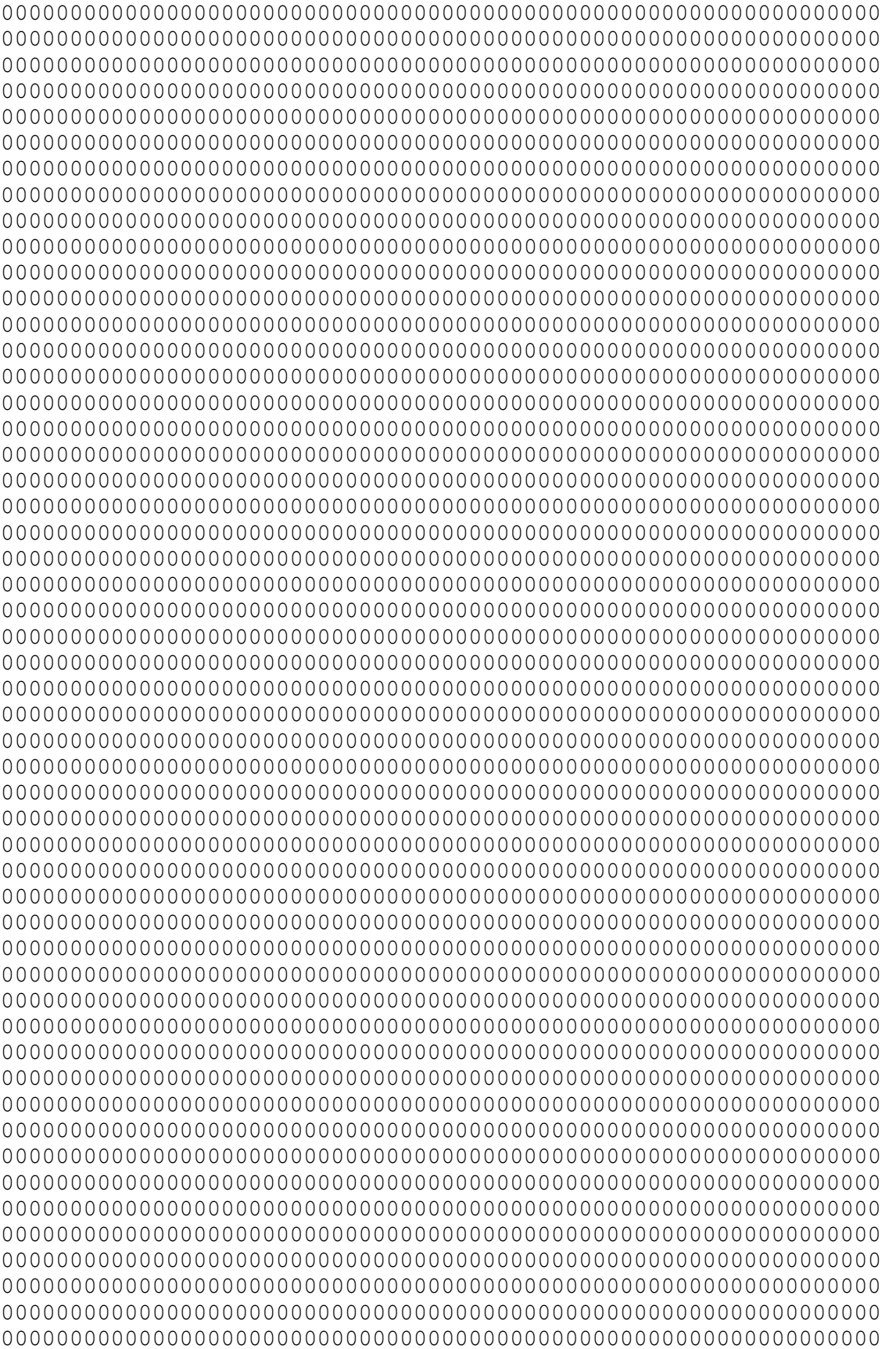


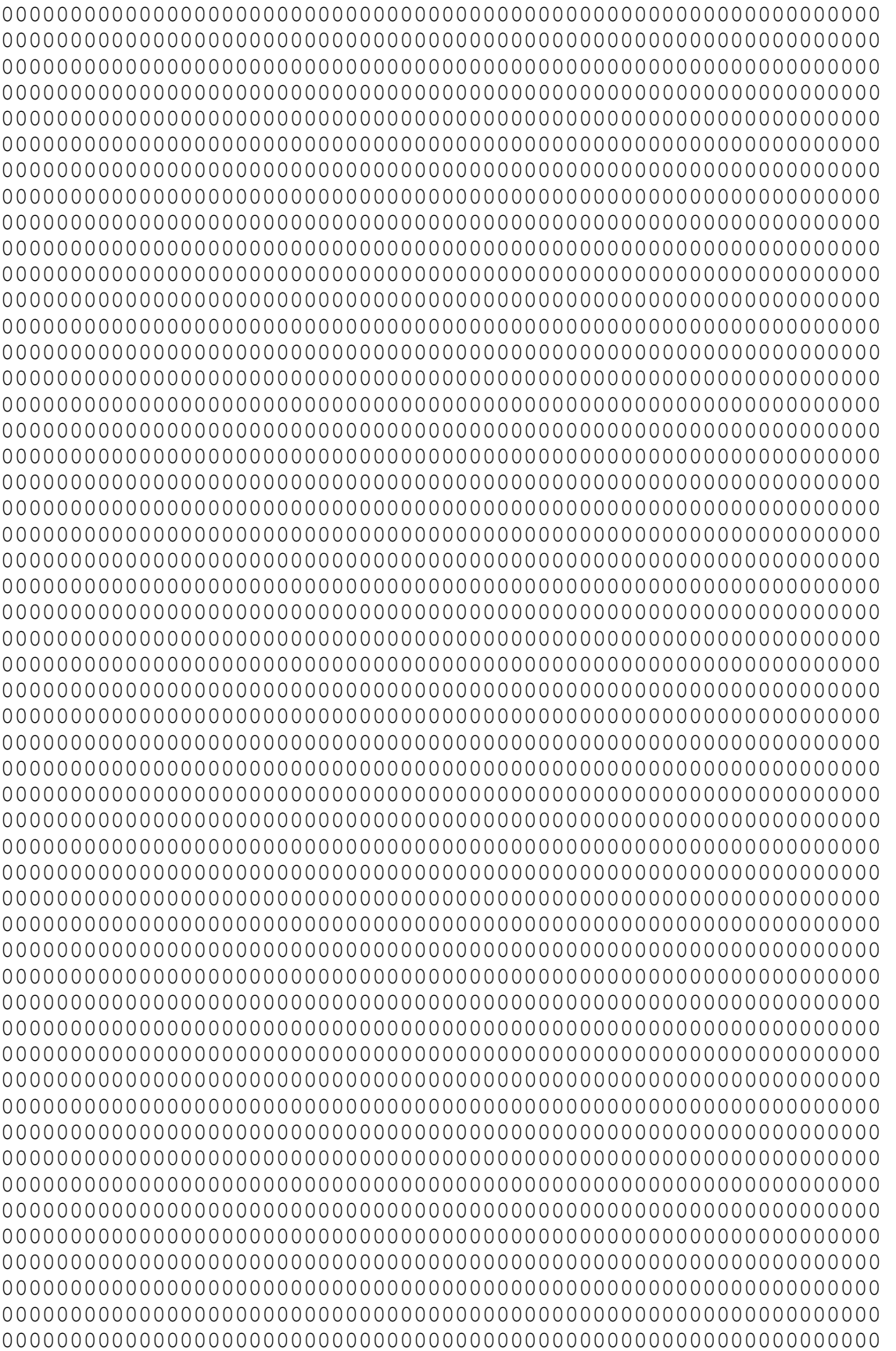


```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000868686000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000086868600868686008686860000000000000000000000
0000000000008686860000000000000000000000000000000000000000000000
```

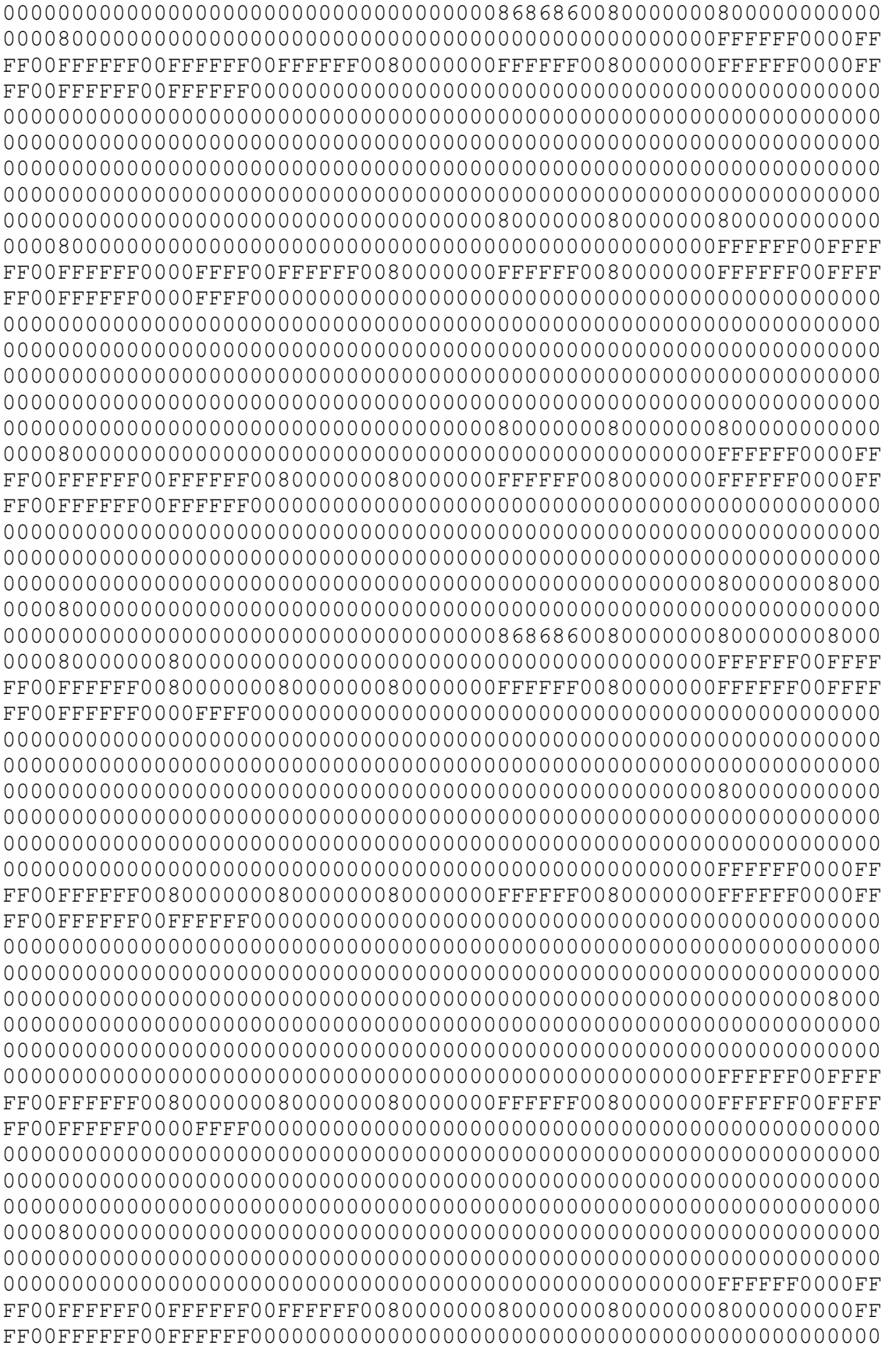




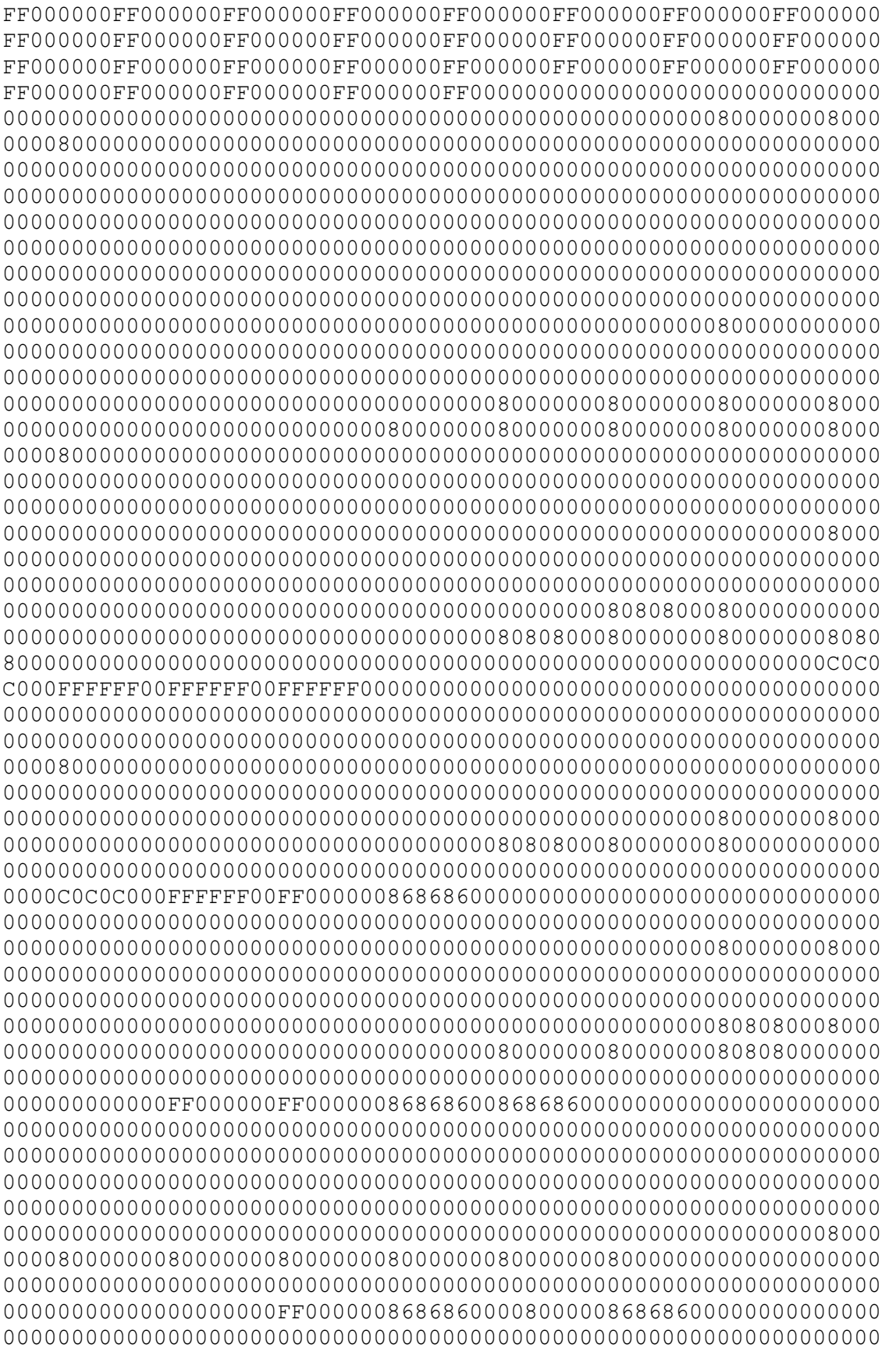


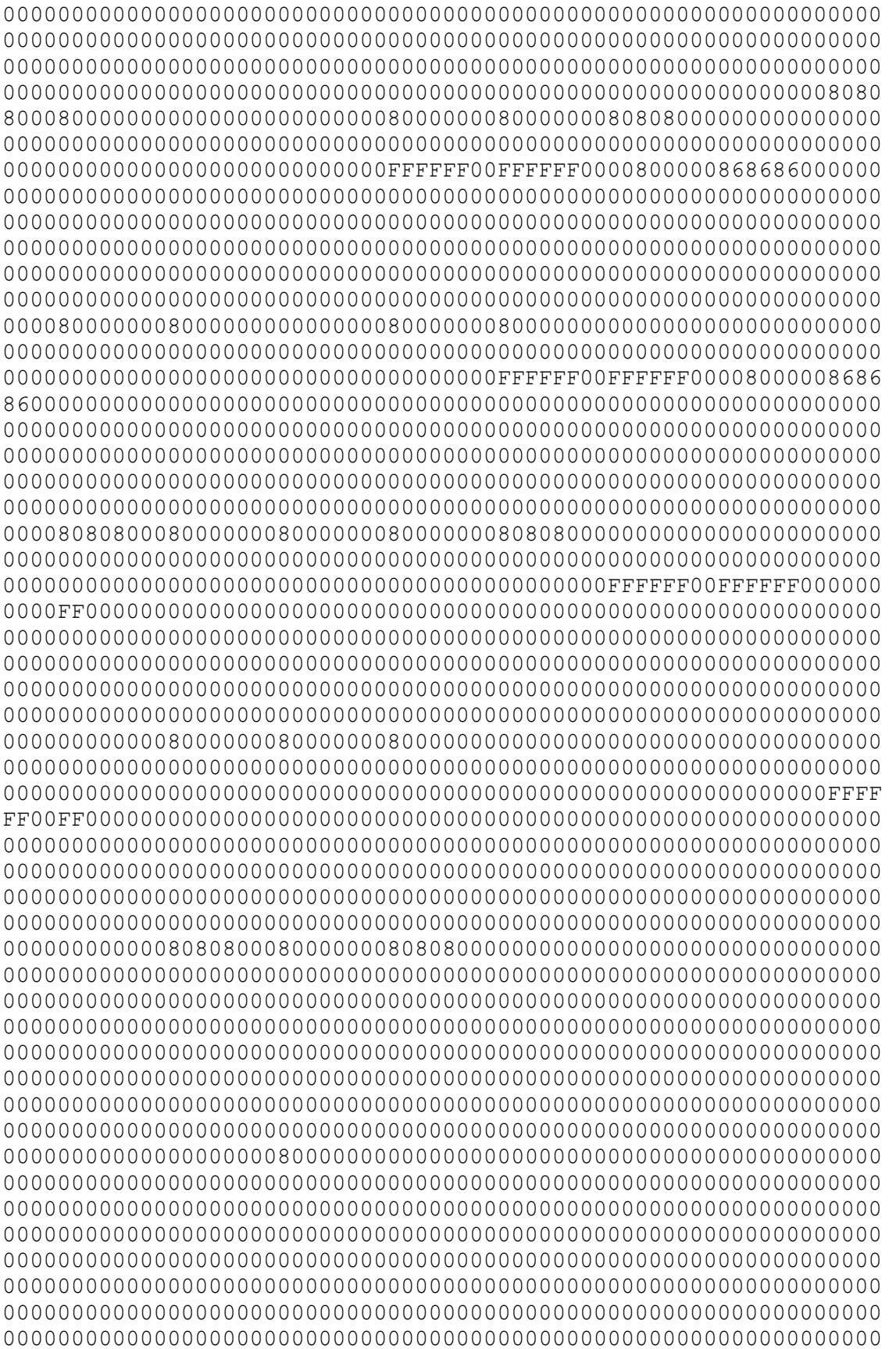


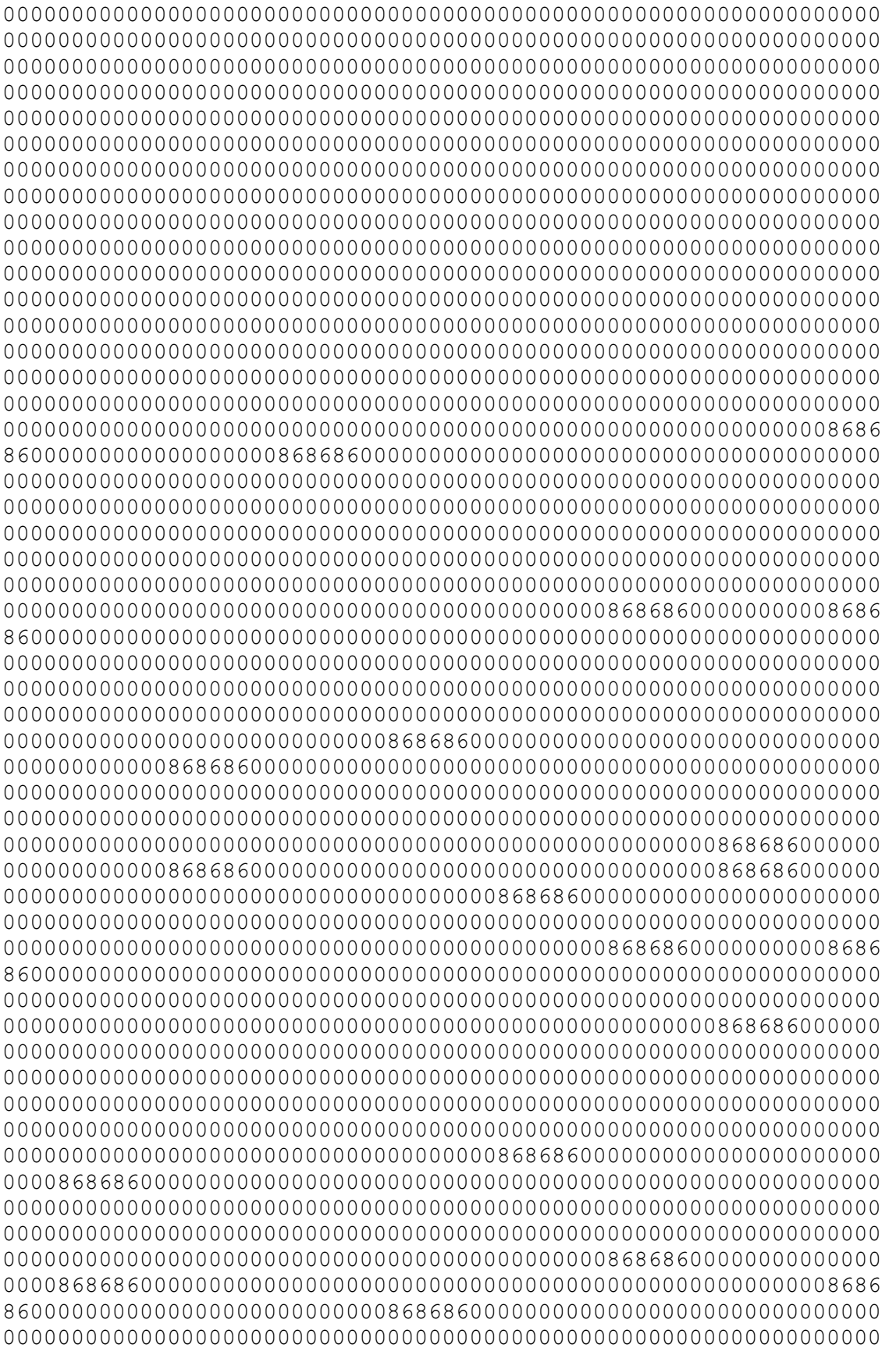
```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000800000000000
0000800000000000000000000000000000000000000000000000FFFFFF0000FF
FF00FFFFFF00FFFFFF00FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFFFF0000FF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000800000000000
0000800000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF00FFFFFF00FFFF
FF00FFFFFF0000FFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000800000000000
0000800000000000000000000000000000000000000000000000FFFFFF0000FF
FF00FFFFFF00FFFFFF00FFFFFF0080000000FFFFFF0080000000FFFFFF0000FF
FF00FFFFFF00FFFFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000800000000000
0000800000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF0000FFFF00FFFFFF0080000000FFFFFF0080000000FFFFFF00FFFF
FF00FFFFFF0000FFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

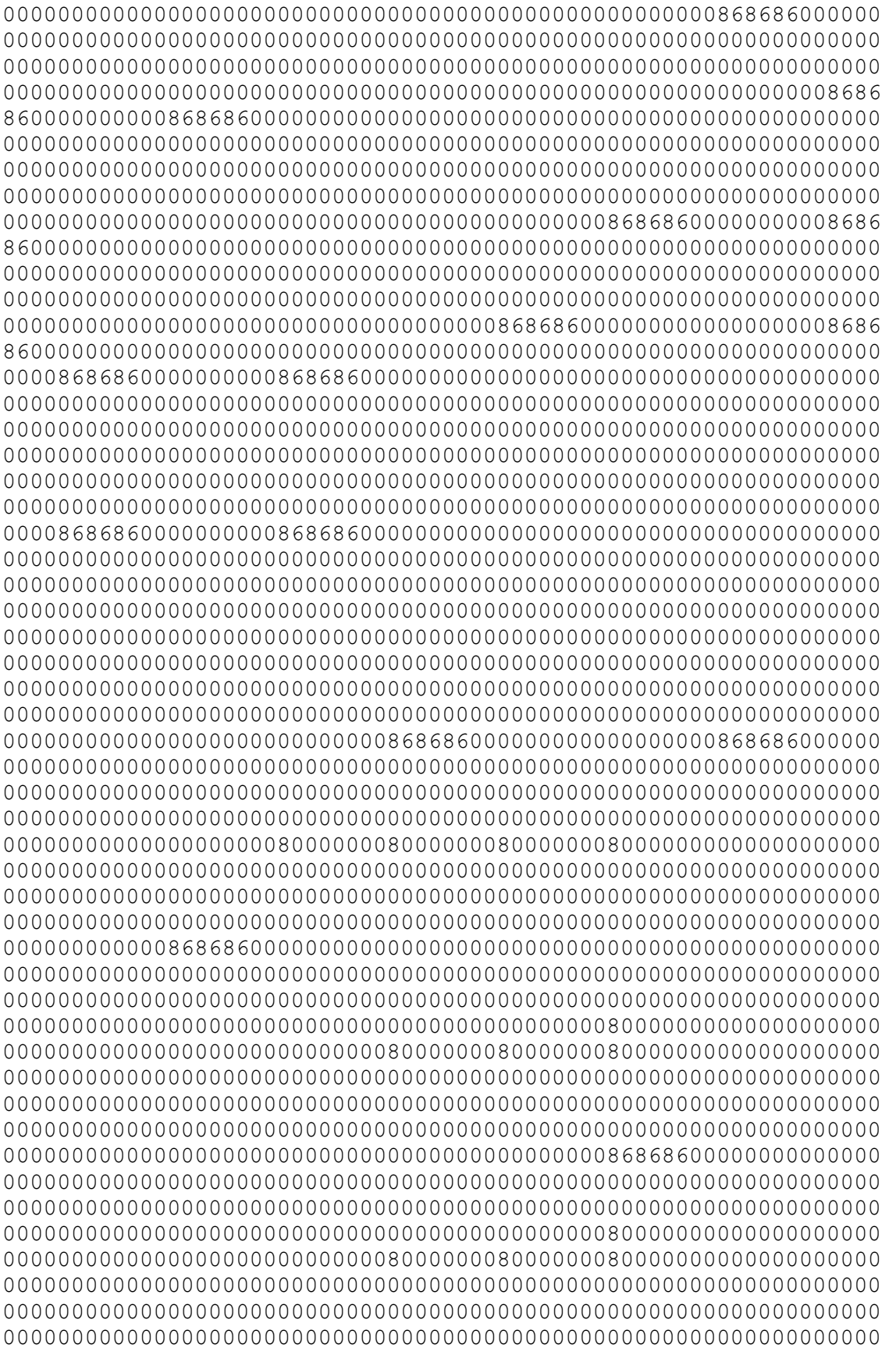


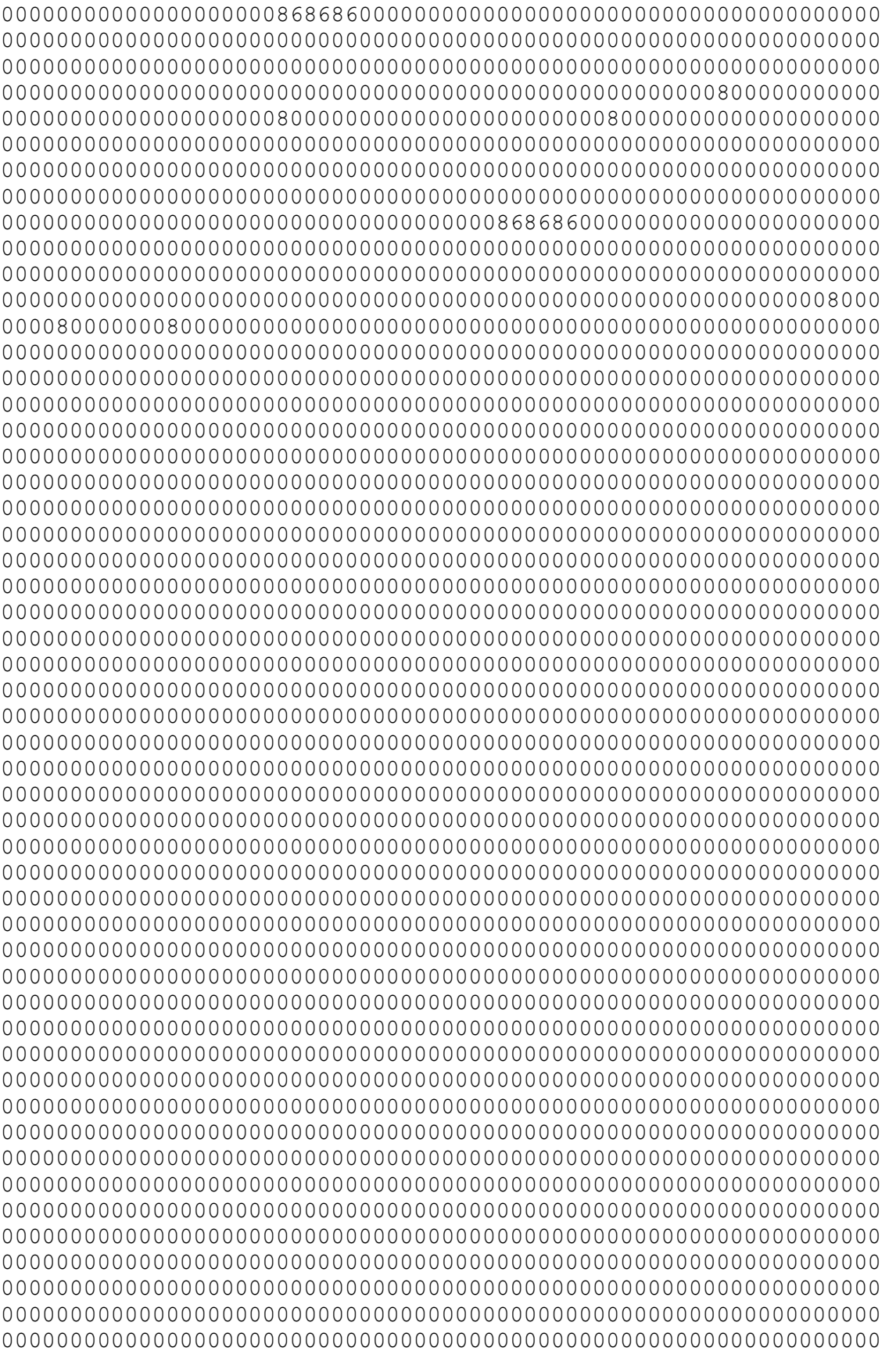
```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000800000008000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF0000FFFF00FFFFFF00FFFFFF00FFFFFF0000FFFF00FFFFFF00FFFF
FF00FFFFFF0000FFFF0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000FF000000FF000000FF000000
FF000000FF000000FF000000FF000000FF000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000FF000000FF000000FF000000
```

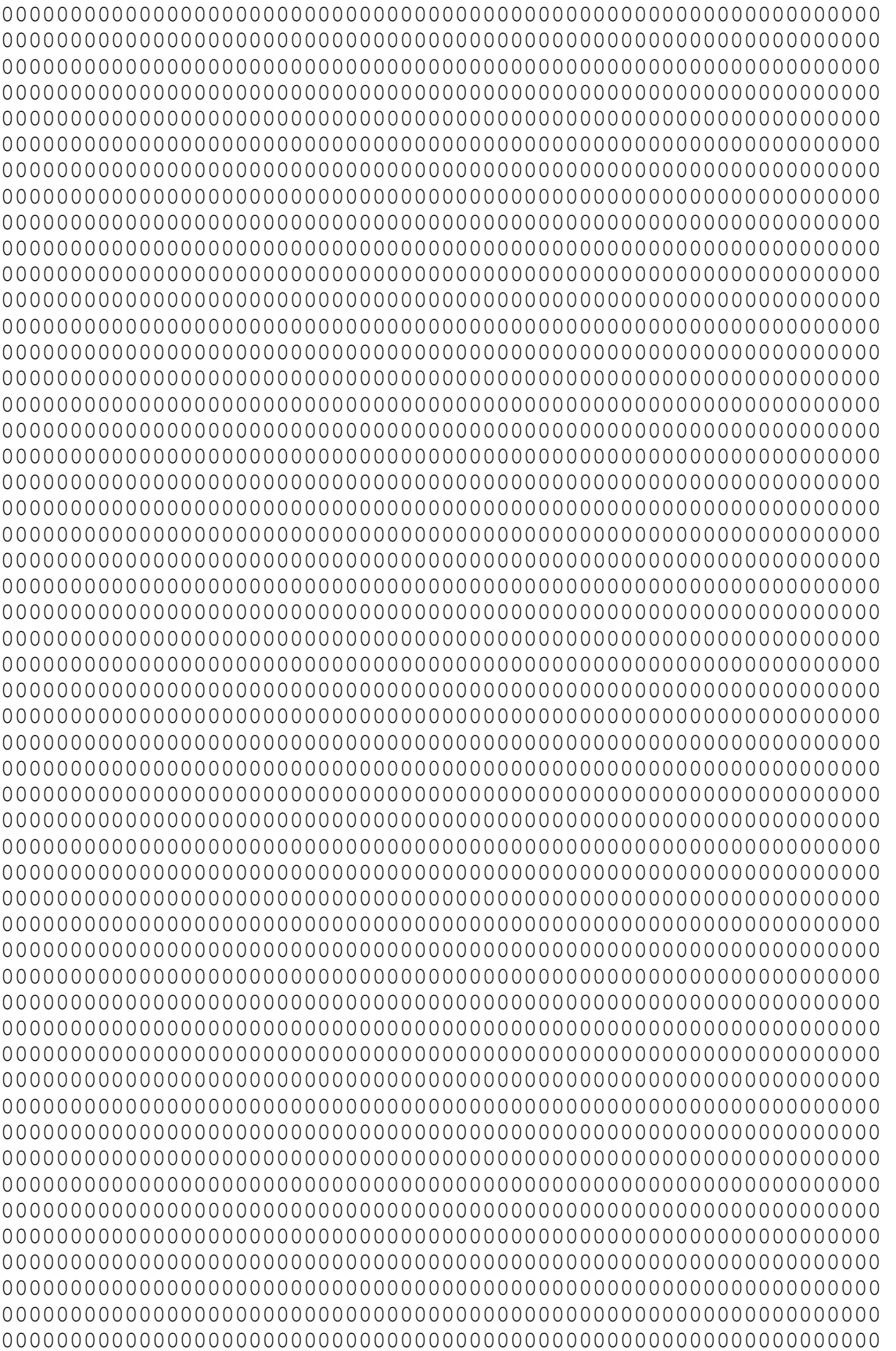


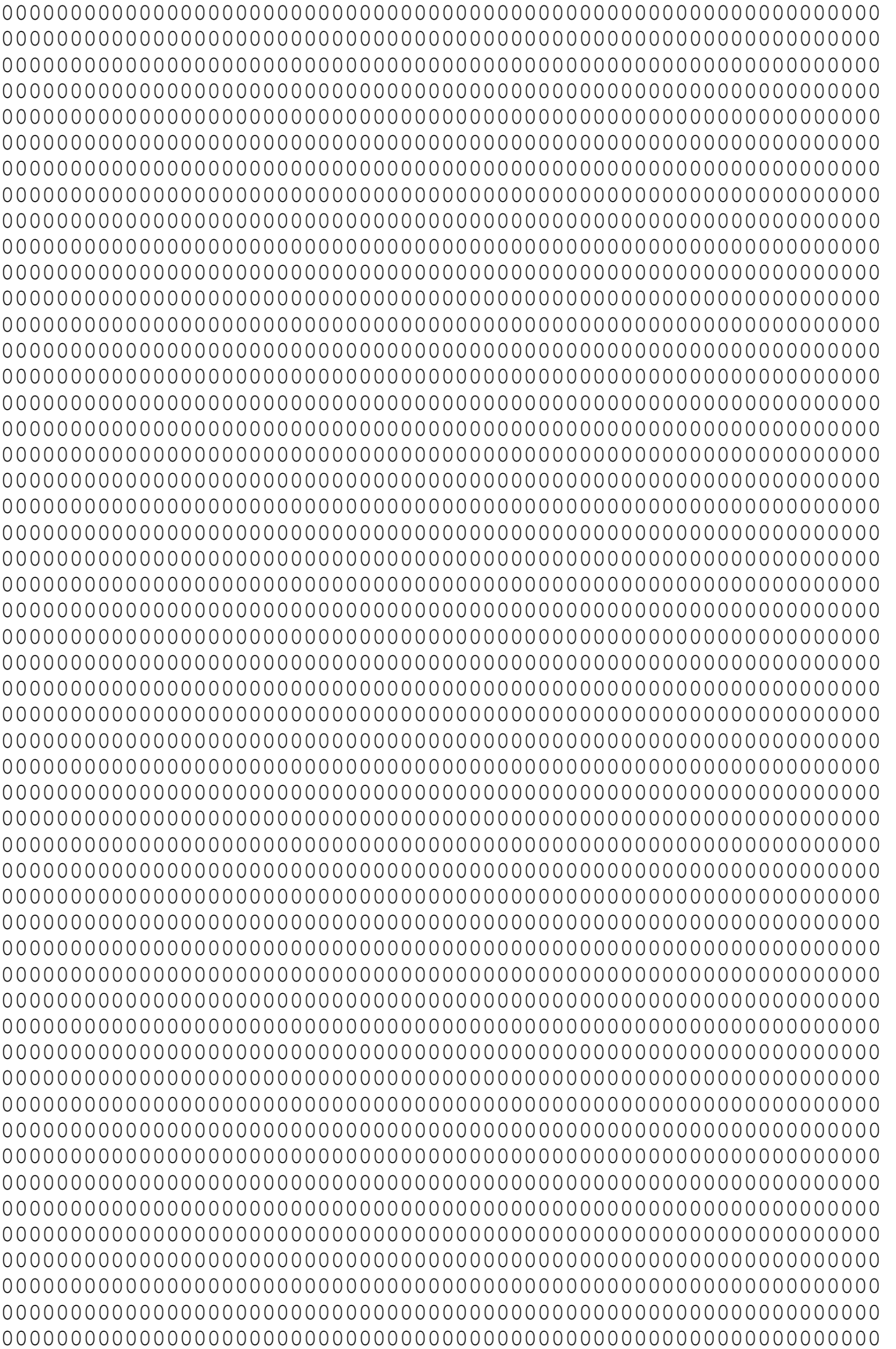


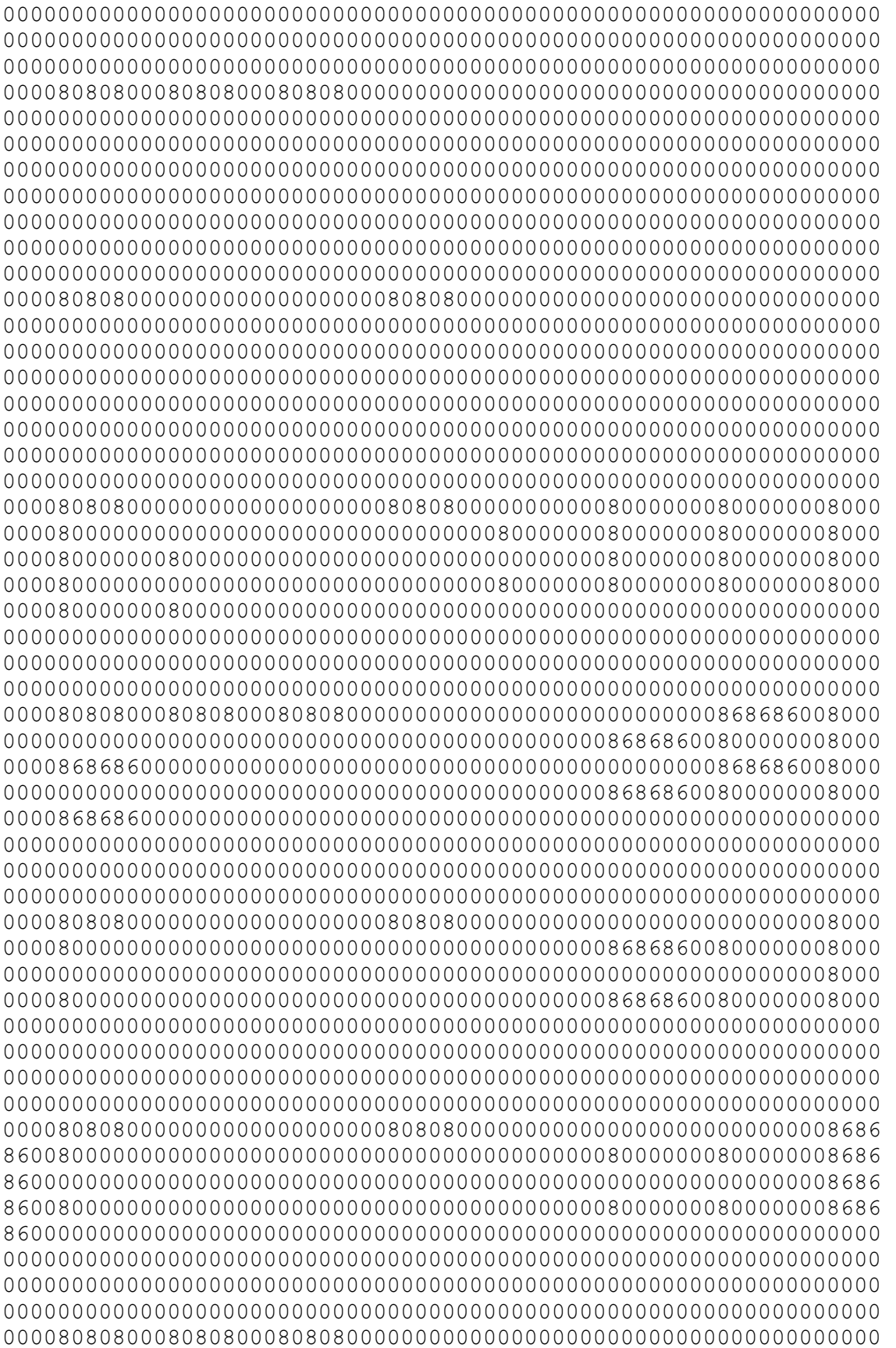


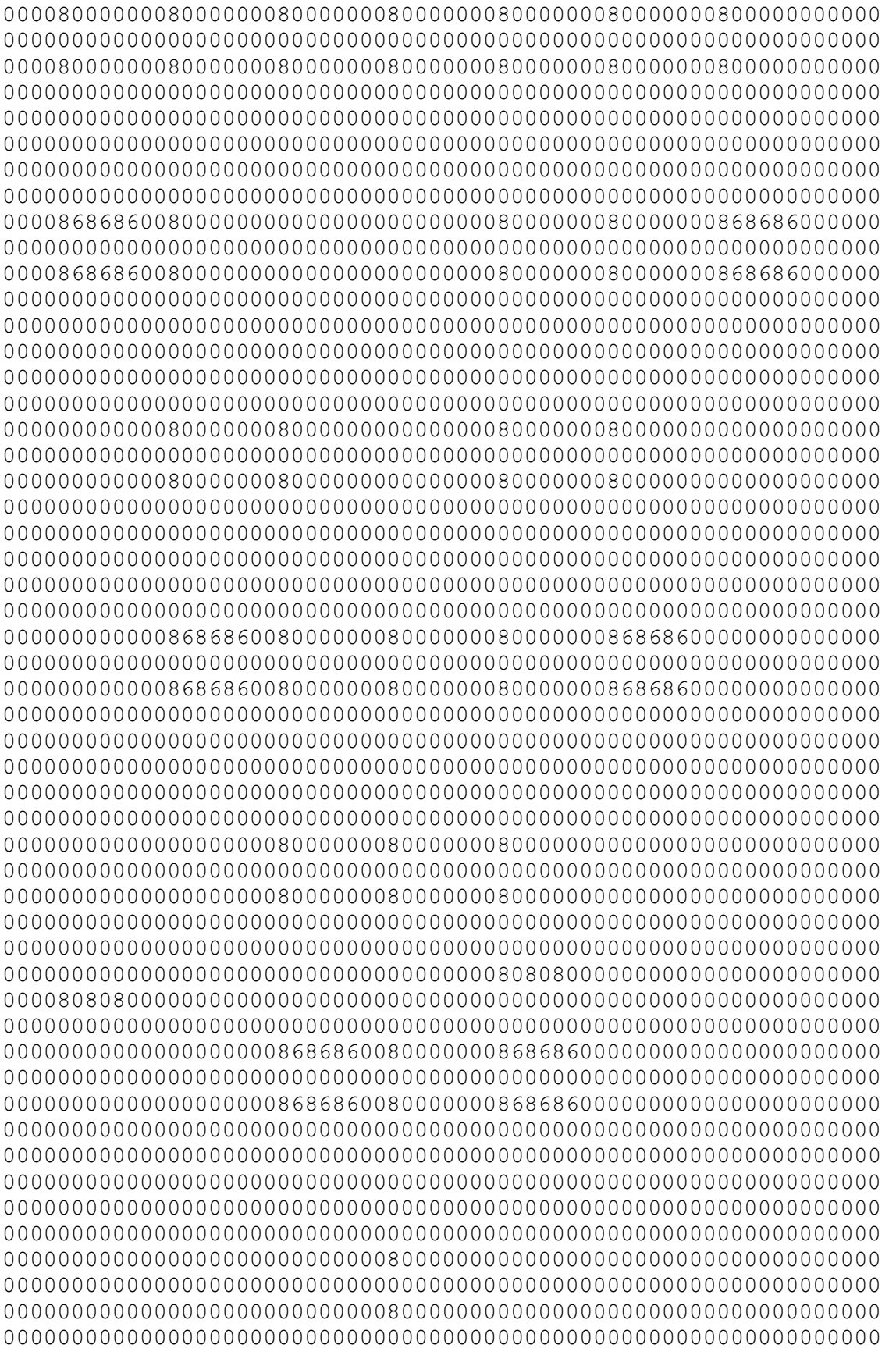


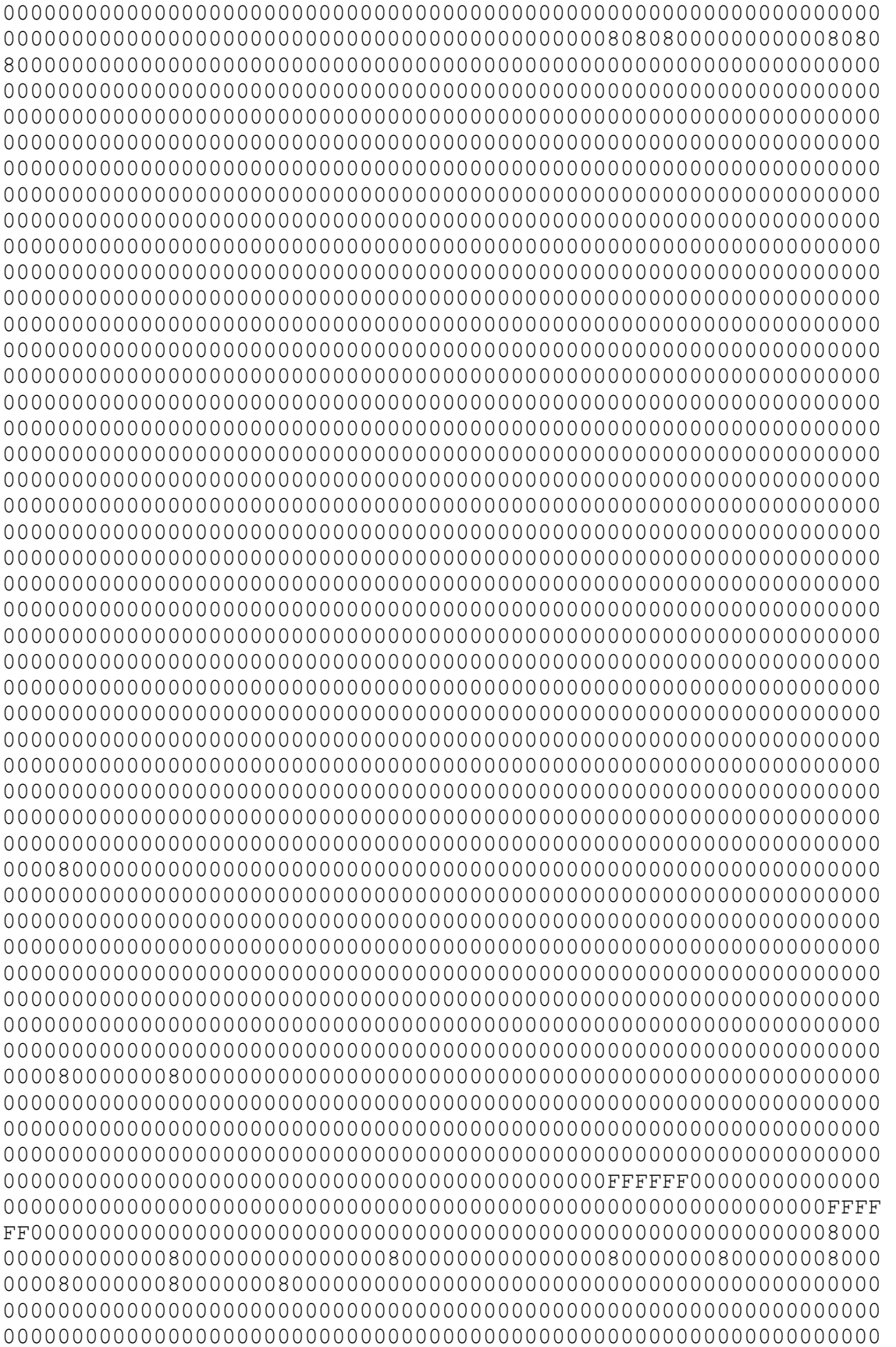


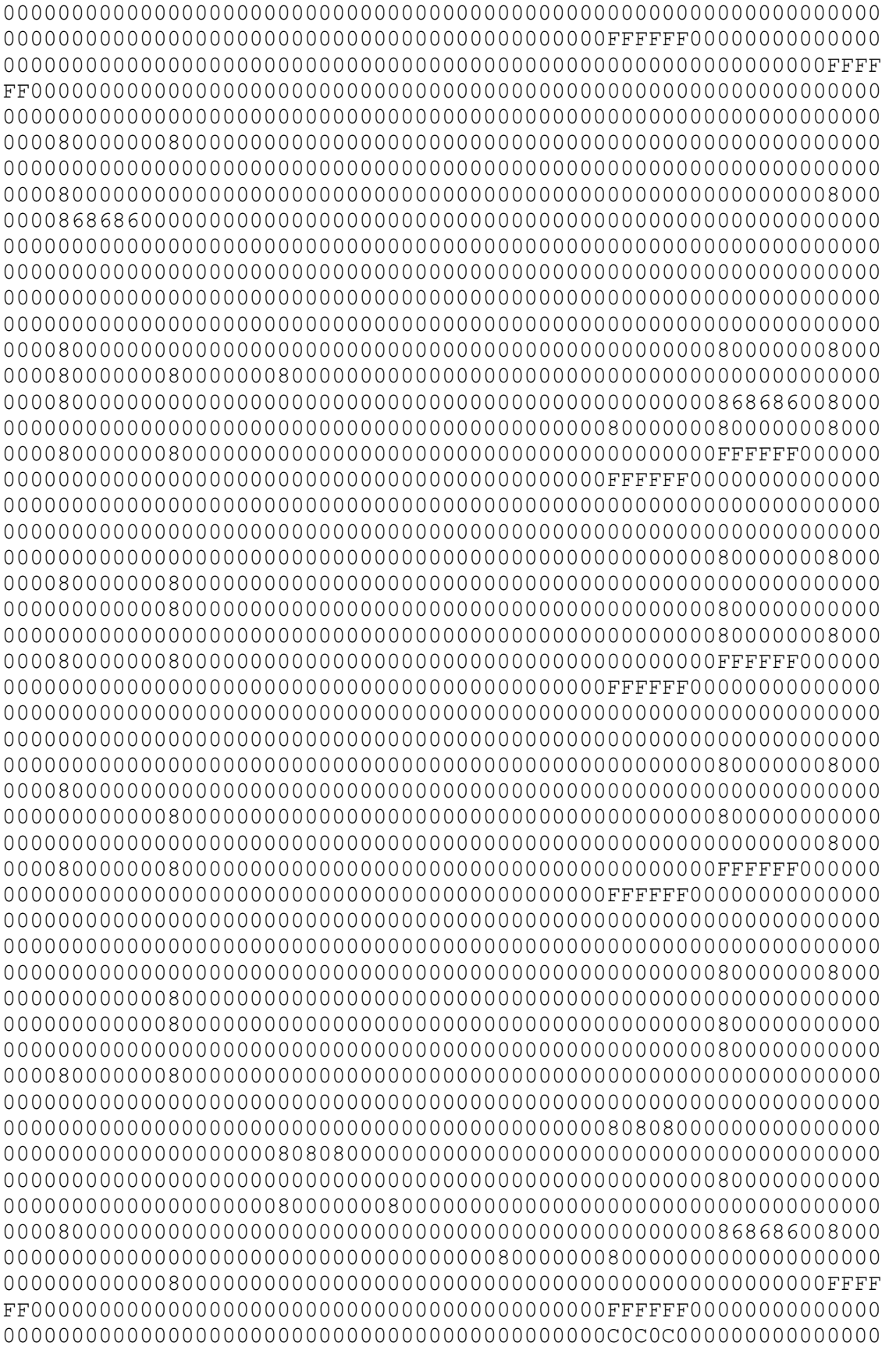


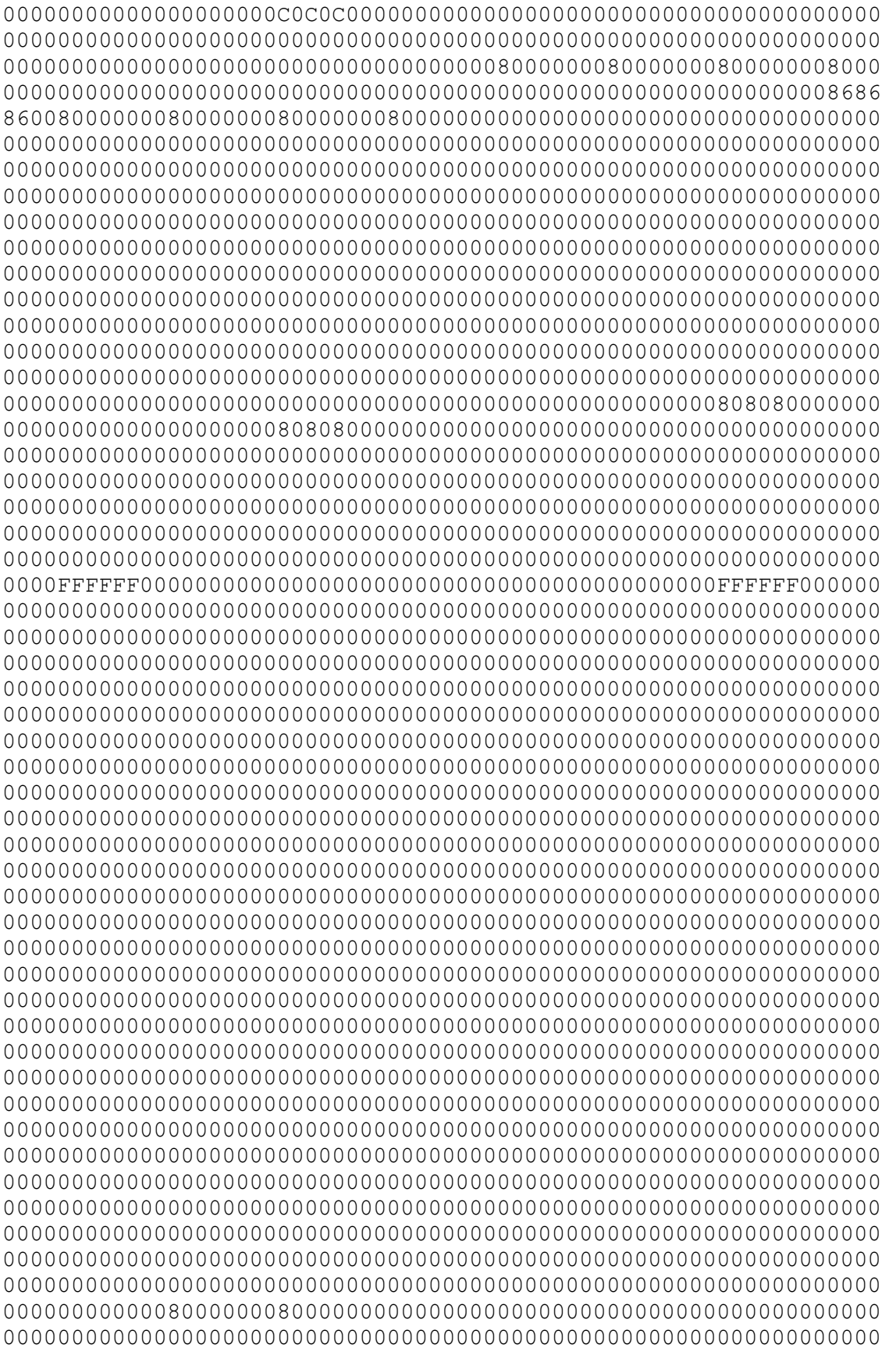




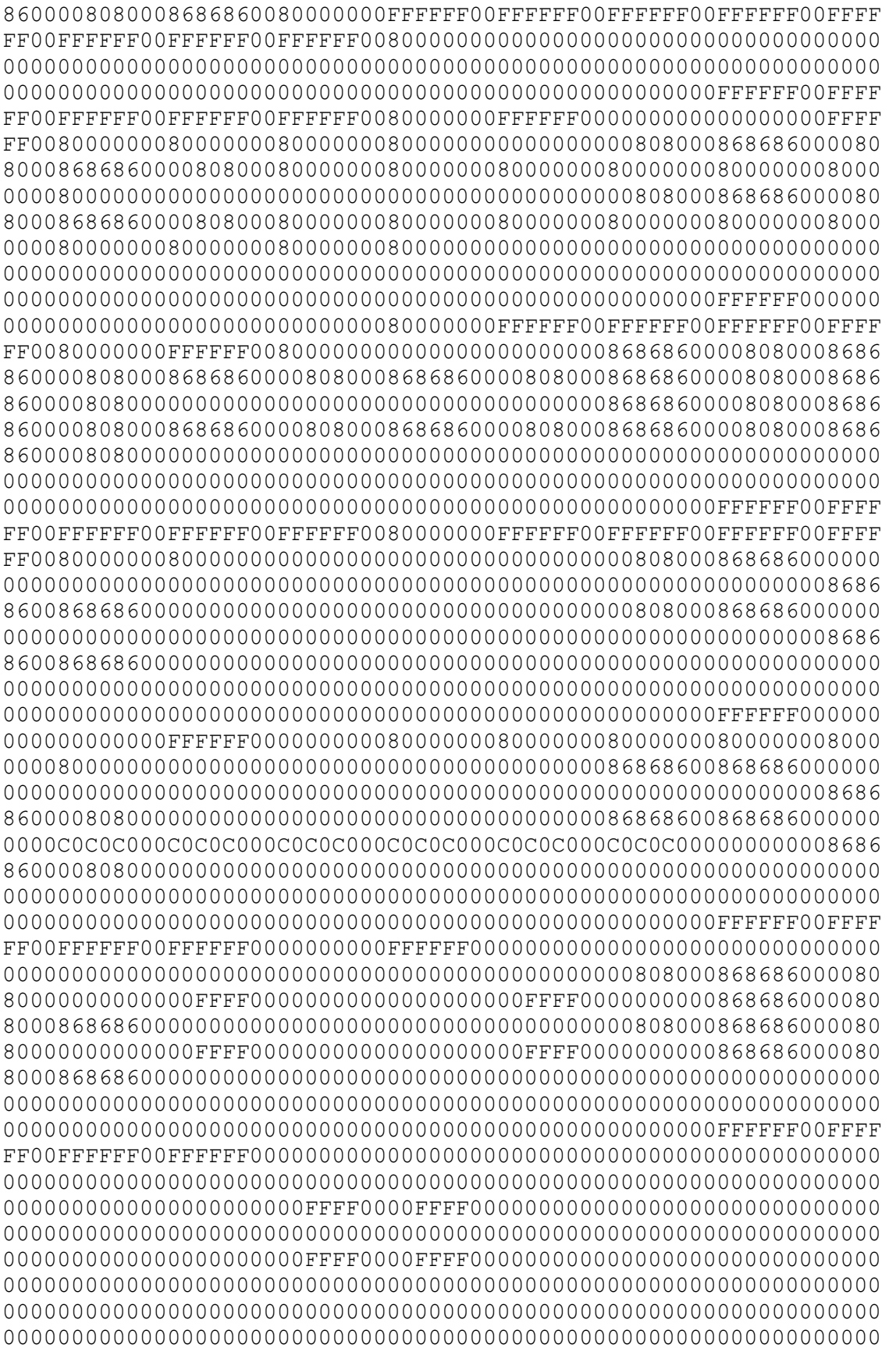


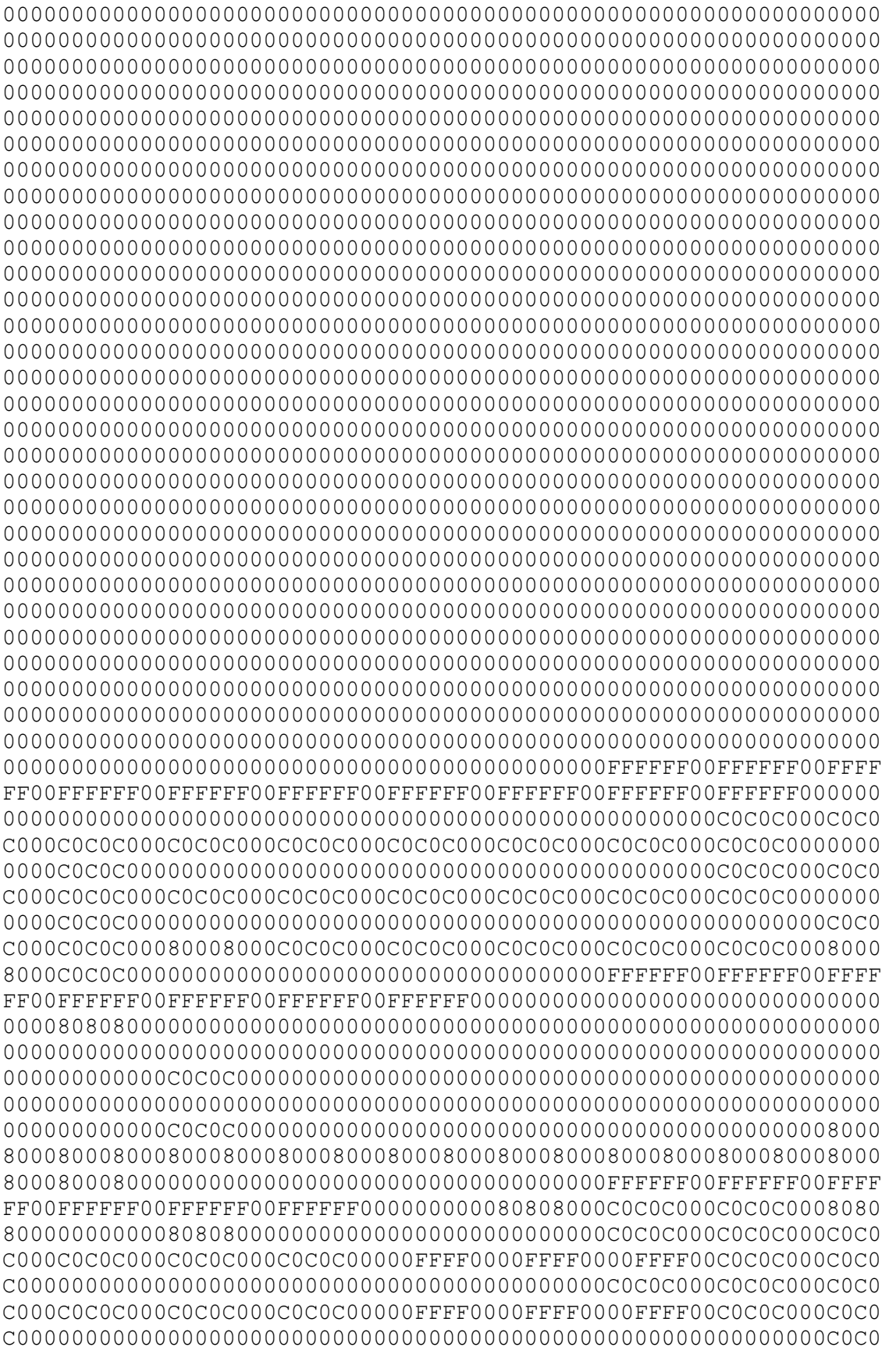


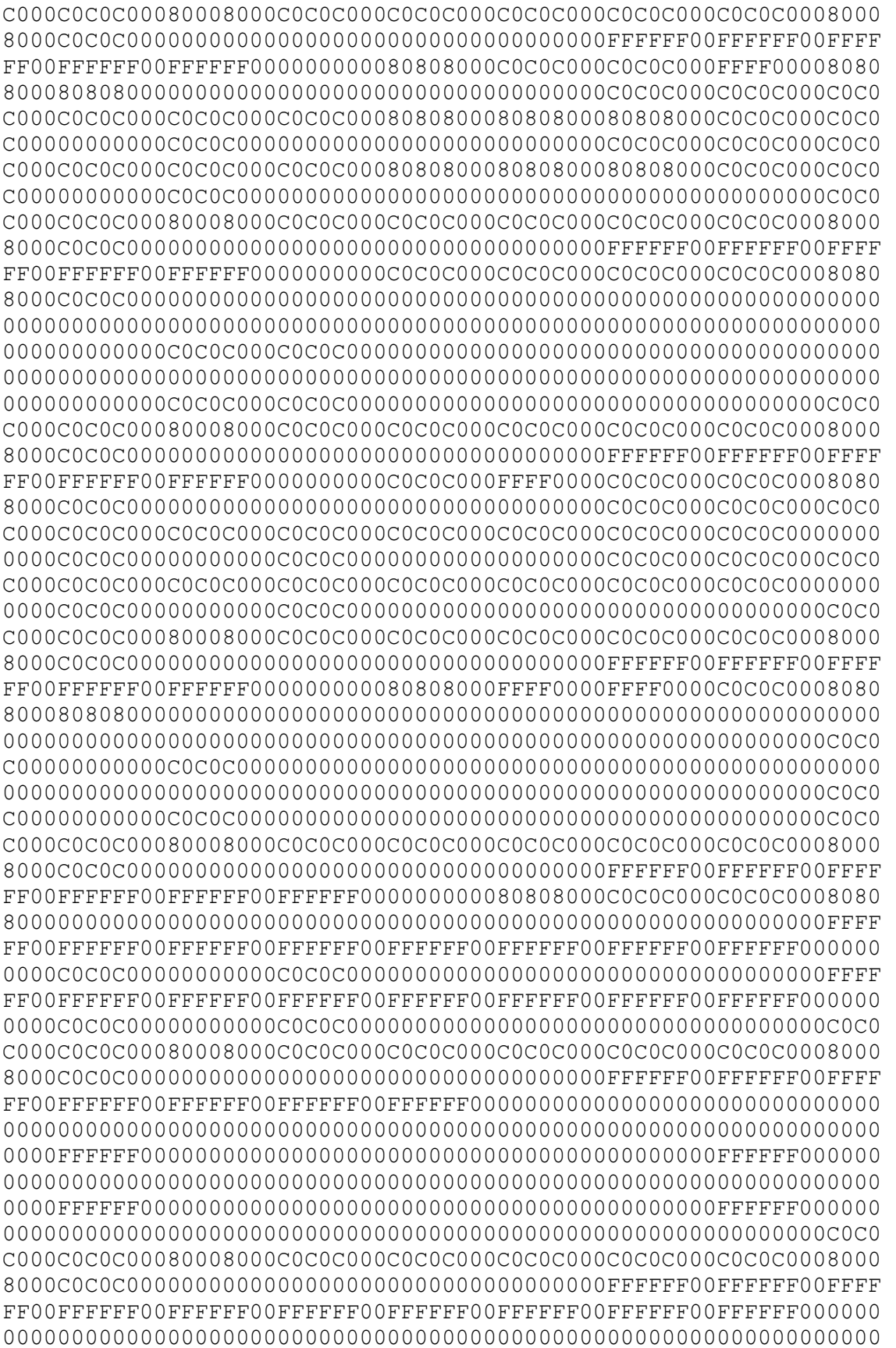


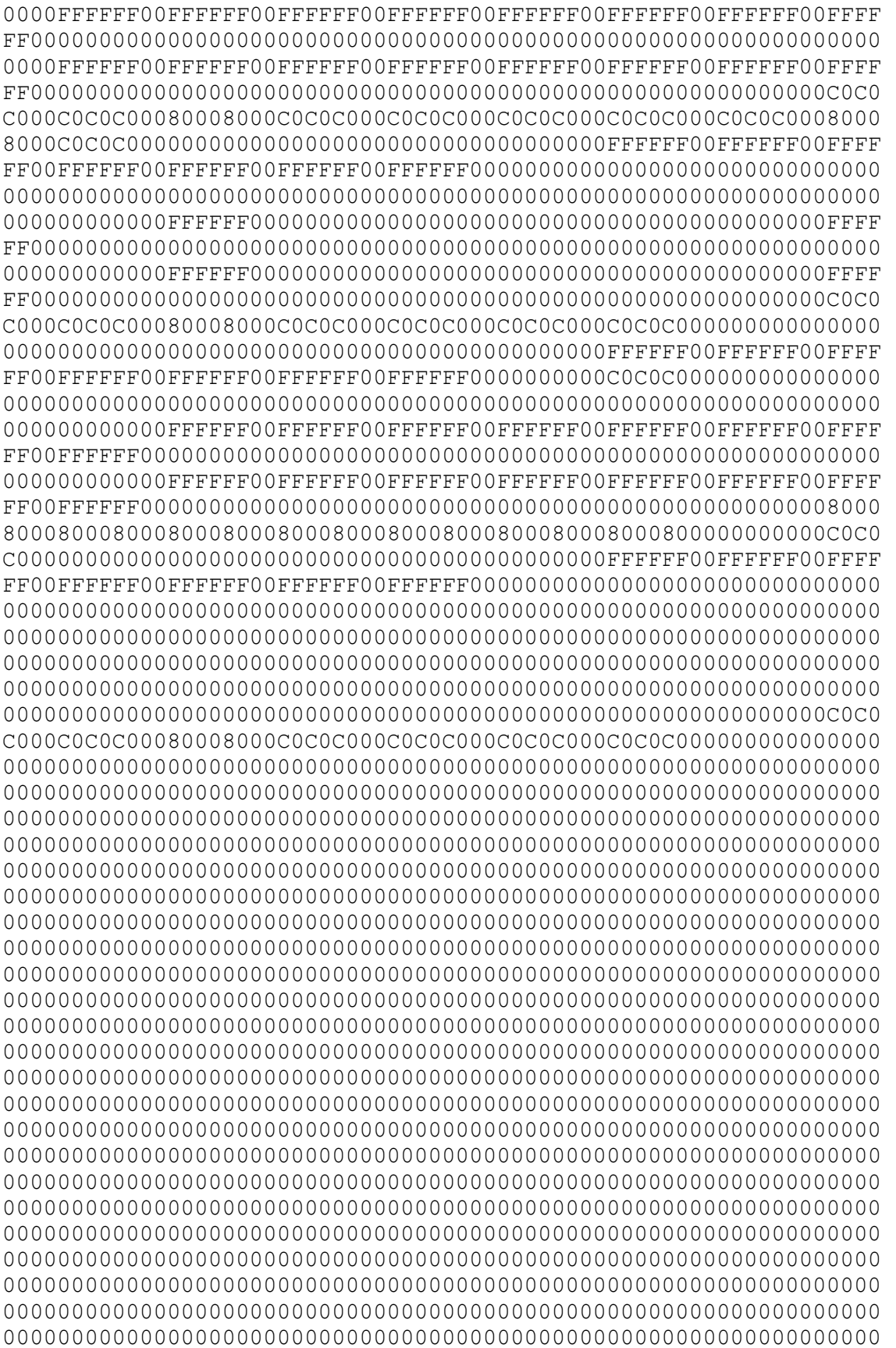


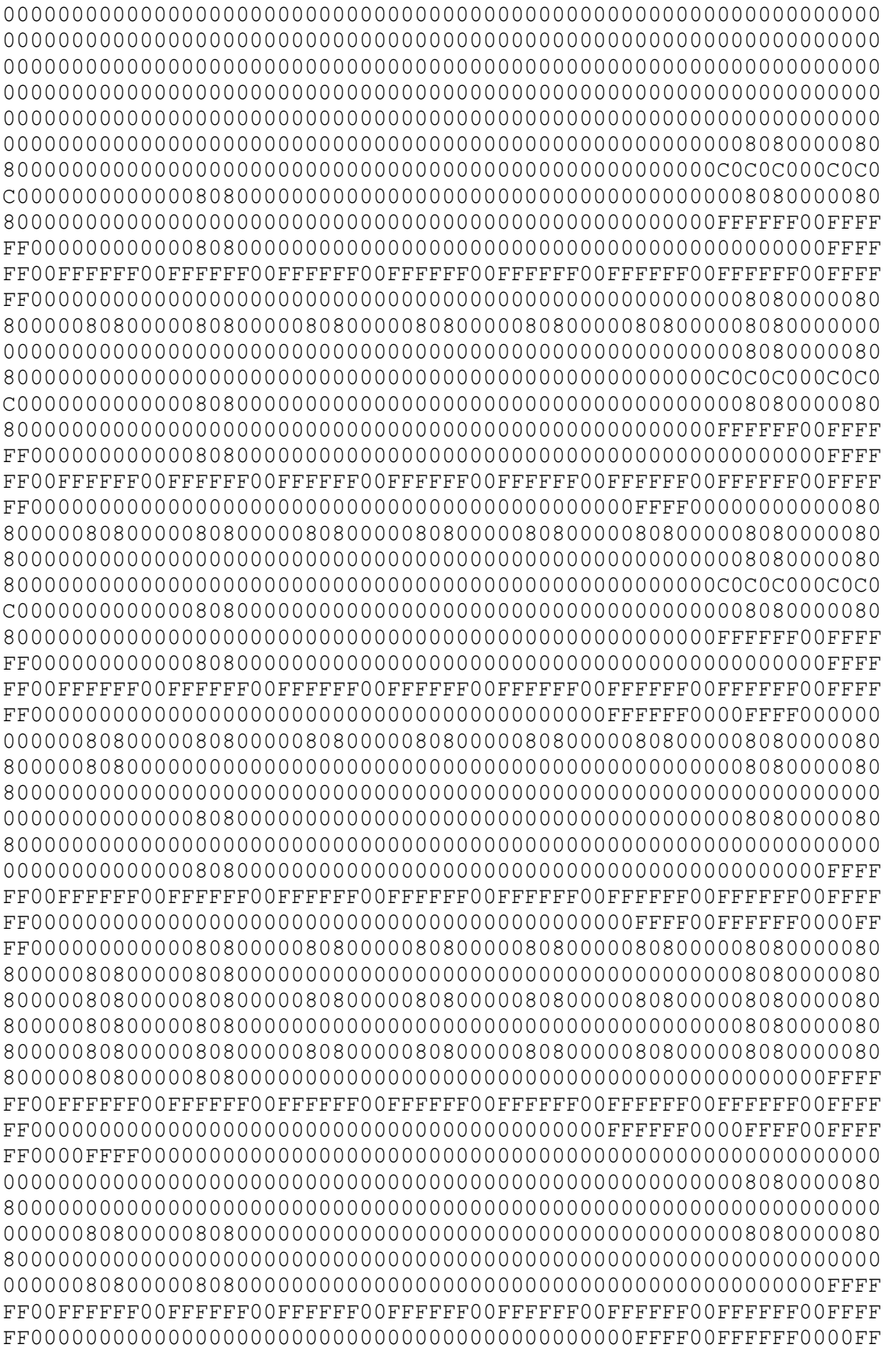
```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000800000008000000080000000800000008000000080000
0000800000008000000080000000800000000000000000000000000000000000
0000000000000000000800000008000000080000000800000008000000080000
0000800000008000000080000000800000000000000000000000000000000000
0000800000000000000000000000800000008000000080000000800000008000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000800000008000000080000000800000008000000
0000800000008000000080000000800000000000000000000000000000000000
0000000000000000000800000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000000000000000000000000000000000
0000000000000000000800000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000000000000000000000000000000000
0000800000000000000000000000800000000000000800000000000000000000
0000800000000000000000000000000000000000000000000000000000000000
0000000000000000000000000800000000FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000000000008686860000808000868 6
8600008080008686860080000000FFFFFF0000000000000000000000000000000
0000000000000000000FFFFFF0080000000000000008686860000808000868 6
8600008080008686860080000000FFFFFF00FFFFFF00FFFFFF00000000000000
0000FFFFFF00FFFFFF00FFFFFF0080000000000000000000000000000000000000
0000800000000000000000000000800000000000000800000000000000000000
0000800000000000000000000000000000000000000000000000000000000000
0000000000000000000000000800000000FFFFFF0000000000000000000000
0000000000000000000FFFFFF0080000000000000000808000868686000080
8000868686000080800080000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000000000000808000868686000080
8000868686000080800080000000FFFFFF00000000000000000000000000FFFF
0000FFFFFF00FFFFFF00FFFFFF0080000000000000000000000000000000000000
0000000000008000000080000000800000000000000800000000000000000000
0000800000000000000000000000000000000000000000000000000000000000
0000000000000000000000000800000000FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000000000008686860000808000868 6
8600008080008686860080000000FFFFFF00000000000000000000000000FFFF
FF0080000000800000008000000080000000000000008686860000808000868 6
8600008080008686860080000000FFFFFF00FFFFFF00FFFFFF00000000000000
0000FFFFFF00FFFFFF00FFFFFF0080000000000000000000000000000000000000
0000800000000000000000000000800000000000000800000000000000000000
0000000000000000000000000000000000000000000000000000FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000FFFFFF0000000000000000000000
0000000000000000000FFFFFF0080000000000000000808000868686000080
8000868686000080800080000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF0080000000FFFFFF0080000000000000000000000000808000868686000080
8000868686000080800080000000FFFFFF0000000000000000000000000000000
0000000000000000000FFFFFF0080000000000000000000000000000000000000
0000000000000000000000000800000000000000800000000000000000000000
0000000000000000000000000000000000000000000000FFFFFF000000
0000000000000000000000000000000800000000FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF0080000000000000008686860000808000868 6
8600008080008686860080000000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF0080000000800000000000000000000000000000008686860000808000868 6
```

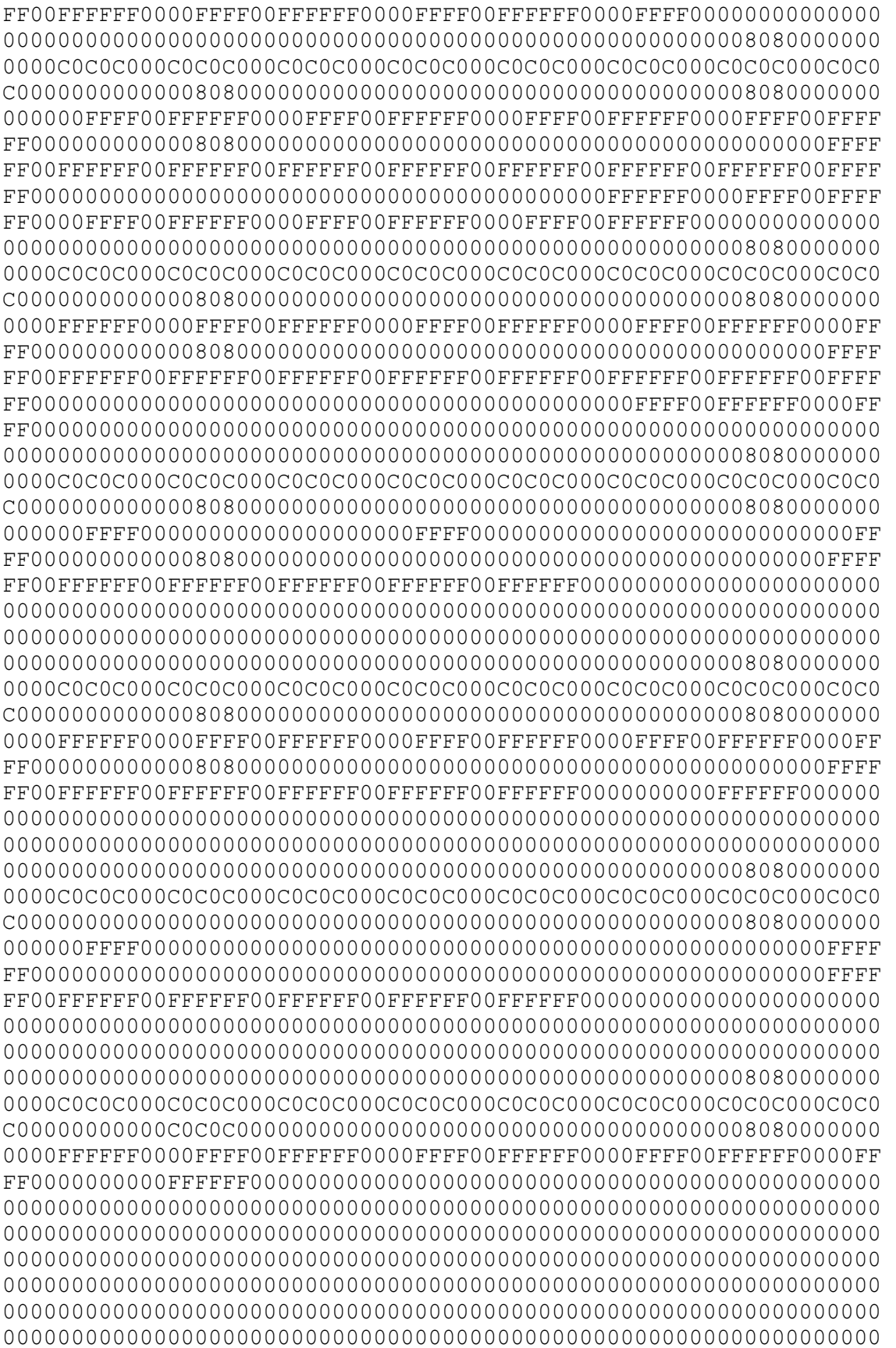












```
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000424D3E000000000000003E000000
28000000400000007001000001000100000000000800B000000000000000000000
00000000000000000000000FFFFFF00FFE3000000000000FBC2000000000000
FBC0000000000000FBC0000000000000FB80000000000000F80F000000000000
F81F000000000000F01F000000000000E21F000000000000E01F000000000000
000F000000000000C07F0000000000000080FF0000000000000BFF000000000000
1BFF0000000000003BFF000000000000FFFFFFDFFC00F8FFFC00FFCFFC00F07F
FC00FFC7FC00E23FFC0000030000C41FFC00000100018A0FEC0000000003910F
E400000100007018FE00000300007441F000000070023803F0001000F0001D21F
0003001F0000E00F0007007F0023F187000F00FF0023FBC3E3FF01FF0023FFE1
E7FF03FF0007FFF0EFFFFFFF003FFFF8FFFFFFFF8000FFFF000F00000000FFFC
000F6EEE7FFCF3FF000F6EEE67FCE1F9000F60004004E1F3000F6EEE7FFCF0E3
000F6EEE3FFCF847000F60000000FC0F000F6EEE0001FE1F00046EEEFFFFFC0F
00000000FC3FF84F00006FFEFC1FE0E3F8006FFEF80FC1F1FC000000FE3FC7F8
FE04FFFFFE3FFFFFFFFFFFFFFE7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
AAAB80038003AAABFFFFFFFFBFFBFFFFBFFBBFFBBFFBBFFBFFFFFFFFBFFBFFFF
BFFBBFFBBFFBBFFBFFFFFFFFBFFBFFFFBFFBBFFBBFFBBFFBFFFFFFFFBFFBFFFF
BFFBBFFBBFFBBFFBFFFFFFFFBFFBFFFFBFFBBFFBBFFBBFFBFFFFFFFFBFFBFFFF
8003AAAB8003AAABFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBFFDBFFDFFFFFFFF
00000000AAABAAAB80018001BFFFFFFB80018001BFFBBFFB80018001BFFFFFFB
80018001BFFBBFFB80018001BFFFFFFB80018001BFFBBFFB80018001BFFFFFFB
80018001BFFBBFFB80018001BFFFFFFB80018001BFFBBFFB80018001BFFFFFFB
00000000AAABAAABBFFDBFFDFFFFFFFFFFFFFFFFFFFFFFFF000000000000BFFD
00000000000000000000000000008001000000000000080010000000000008001
00000000000080010000000000080010000000000008001000000000000800 1
00000000000080010000000000080010000000000008001000000000000800 1
FE3FFFFFFFFF0000FF1FFFFFFFFFBFFDFFFFFFFFFFFFFFFFFC1C1C1C1FFFFFFFF
C1C1C1C100FFFEFFC1C1C1C100FFFC7FC1C1C1C100FFF83FFF7FFF7F00FFFEFF
FE3FFE3F001BFEFFFC1FFC1F0019F83FFF7FFF7F0000F83FFF7FFF7F0019F83F
FC1FFC1F001BF83FFC1FFC1F00FFF83FFC1FFC1F00FF8003FC1FFC1F00FF8003
FC1FFC1F00FF8003FC1FFC1FFFFF8003FFFFFFFFFFFFFFFFFFFFFFFFF80FF80
FEDF0000FF80FF80FEDF0000FF80FF80FEDF0000FF80FF80FEDF0000FF80FF80
FEDF000006FF06FFF8DF0000067F067FF0DF0000003F003FE0DF0000067F067F
E0DF000006FF06FFE0DF0000FF80FF80F0DF0000FF80FF80F8070000FF80FF80
FFFF0000FF80FF80FFFFFFFFFF80FF80FFFFFFFFFFFFFFFFF0001FFFF000FFFFF
0001CFE7000FC0030001CF87000FFFFF0001CF8F000FC0030001C01F000FFFFF
0001C03F000FC0230001FFFF000FC03F00018823000FD0230001FFFF0004CABF
0001C0070000D5230001C0070000CABF0001CFE7F800C0230001CFE7FC00FFFF
0001CFE7FE04C0030001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8438001E000
FFFFF5158001E000FFFFE8028001E000FFFFE0E08001E000FFFFC8028001E000
FFFF85158001C000FFFF000380018000C003000F80010000C003000F80018000
FFFF000F8001C000FFFF000F8001E000FFFF000F8001E000FFFF801F8001E001
FFFFC03F8001E003FFFFE07F8001E007FFFFFFFFFFFFFFFF3FFFFFFFFFF0001
```

DFFFFFFFFFFF000DB803F5FFFFD700191FFFF5FFFFD70031FFFFF5F7EFD70061
1FFFF5E7E7D70001BFFFF5C7E3D70001D803E587E19700013FFFC5C7E3170001
FFFFC5E7E71700011FFFC5F7EF170001BFFFF0FFFFC300013803FFFFFFFF0001
BFFFFFFFFFFF0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFF1FFF1FFFFFFFFFFF18031803FFFFFFFF1FFF1FFFFFFF8003FFFFFFFF
8003FFFFFFFFFFFFFFFFFF1FFF1FFFFFFFFFFF180318038003800031FFF1FFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1FFF1FFFFFFFFFFF18031803
FFFFFFFF1FFF1FFFFFFFFFFFFFFFFFFFFFFFFFFF0000FFFFFFFFFEFF0000FFFF
8007FFFF0000FFFFFFFFC27F0000FFFF8001FFFFFC7FFFFFFFFFC200F83BFFFF
EE07FFFFF019FFFFE607DE07E009800383FF9E07C001FFFFE60107FFE0018003
EE019E00F001FFFFFFFFDE00F803FFFF8007FFFFF80FFFFFFFFFC200FA1FFFFF
8001FFFFFB3FFFFFFFFFFEFFF87FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFC007C007C0070003FFFFFFFFFFFFFFFFC03FF807F83F0003
FFFFFFFFFFFFFFFFC007C007C0070003FFFFFFFFFFFFFFFFC03FF807F01F0003
FFFFFFFFFFFFFFFFC007C007C0070003FFFFFFFFFFFFFFFFC03FF807F83F0003
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8001FFFF
FFFF81D7800181FFFFFFFFD78001FFFFE67FFFD78001843FF0FF81D78001FFFF
F9FFFF178001821FF0FFFF178001FFF7E67F811780018461FFC7FF038001FFF5
FFDFFFFF8001821DFFEF80038001FFFFFFF7FFFF8001841FFFCFFFFF8001FFFF
FFFFF80380018107FFFFFFFFFFFFFFFFFFFF00000000FFFFFFFF00000000FFFF
FFFF00000000FFFFFFFF00000000FFC7FC1FFFFFFFFFFFDFF9CF0E07C1FFFFEF
F9CF9F0FC0FFFFF7F9CFCF1FE07FE64FF9CFCF1FF03FF0FFF9CFE03FF81FF9FF
F9CFE63FFC0FF0FFF9CFF27FFE07E67FFC1FF07FFF03FFFFFFFFF8FFFF83FFFF
F007F8FFFFC7FFFFFFFFFDFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0FE0FFFFFC71
FFFF9FE1FFFFFEFBFFFF8FE3C47F9603FFFFCFC3CE7F6F277DEEC007E0FF6F27
7D6EE787E4FF8F8F8331E38FF1FF6F8FBB15F30FF1FF9FDFD73BF11FFBFFFFFF
D77BF81FFFFFFC3FEFFFF83FFFFFBE3FEFFFFC3FFFEFBE3FFFFFFC60FFC7DDBF
FFFFFE71FF83E3FFFFFFFFFBFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFE00FFFFFFFFFFFFFFFFFFFFFFF81FFF83FFC1FFFFFE3FFF39FF9CF77BE
F1FFF39FF9CF75BEF8FFF39FFFCF8CC1FC7FF39FC001AC5DFE3FF39FF9FFDCEB
FF1FF39FF9CFDDEBFF8FF39FF9CFFFF7FF03E10FFC1FFFF7FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF71FFFFFFFFFFF
FF36FFFFFFFFFFFF51687038703FFFFDF31CF87CF87F00FFF76E78FE78FF8C7
BFF6E78FE78FF8C7FFD1F01FF01FF8C7DFFFF31FF31FF80FFFBFF93FF93FF8C7
77FFF83FF83FF8C777BFFC7FFC7FF8C707FFFC7FFC7FF00FAFD7FEFFFEFFFFFF
8FE7FFFFFFFFFFFDFC7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7
FFFFFFFF07C1FFF3FFFFFFFF07C1EA81FFFFFFFF07C1FFF3FFF7E7FF0101FFF7
C1F7CF830001FFFFC3FBDFC302011C7FC7FBDFE302011C7FCBFBDFD38003087F
DCF7CF3BC107087FFF0FE0FFC107007FFFFFFFFFE38F80FFFFFFFFFFE38FC9FF
FFFFFFFFE38FC9FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9FFFFFFFC00FC00
F6CFFE0080008000F6B7FE0000000000F6B7FE0000000000F8B7800000000000
FE8F800000010000FE3F800000030000FF7F800000030000FE3F800100030003
FEBF800300030003FC9F80070FC30003FDDF807F00030003FDDF80FF80078007
FDDF81FFF87FF87FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000CC007C007C003
000880038003C003000100010001C003000300010001C003000300010001C003
000300000000C003000300000000C003000380008000C0030007C000C000C003
000FE001E001C003000FE007E007C003000FF007F007C003001FF003F003C007
003FF803F803C00F007FFFFFFFFFC01FFFFFFFFFFFFFFFFFFFFFFFFC001C001
C007001F80018001C007000F80018001C007000780018001C007000380018001
C007000180018001C007000080018001C007001F80018001C007001F80018001
C007001F80018001C0078FF180018001C00FFFF980018001C01FFF7580018001
C03FFF8F80018001FFFFFFFFFFFFFFFF0000000000000000000000000000000000

```
    000000000000}
  end
  object ActionList1: TActionList
    Images = ImageList1
    Left = 56
    Top = 176
    object rvActionEvent1: TrvActionEvent
      Category = 'RVE Custom'
      Caption = 'rvActionEvent1'
      Control = DBRichViewEdit1
    end
    object rvActionNew1: TrvActionNew
      Category = 'RVE File'
      Caption = '&New'
      Hint = 'New Document'
      ImageIndex = 0
      ShortCut = 16462
      Control = DBRichViewEdit1
    end
    object rvActionOpen1: TrvActionOpen
      Category = 'RVE File'
      Caption = '&Open...'
      Hint = 'Open Document'
      ImageIndex = 1
      ShortCut = 16463
      Control = DBRichViewEdit1
    end
    object rvActionSave1: TrvActionSave
      Category = 'RVE File'
      Caption = '&Save'
      Hint = 'Save'
      ImageIndex = 2
      ShortCut = 16467
      Control = DBRichViewEdit1
    end
    object rvActionExport1: TrvActionExport
      Category = 'RVE File'
      Caption = '&Export...'
      Hint = 'Export To File'
      ImageIndex = 85
      Control = DBRichViewEdit1
      ImagePrefix = 'img'
      SaveOptions = []
    end
    object rvActionSaveAs1: TrvActionSaveAs
      Category = 'RVE File'
      Caption = 'Save &As...'
      Hint = 'Save As...'
      ImageIndex = 3
      Control = DBRichViewEdit1
    end
    object rvActionPrintPreview1: TrvActionPrintPreview
```

```
    Category = 'RVE File'
    Caption = 'P&rint Preview...'
    Hint = 'Print Preview'
    ImageIndex = 4
    Control = DBRichViewEdit1
  end
  object rvActionPrint1: TrvActionPrint
    Category = 'RVE File'
    Caption = '&Print...'
    Hint = 'Print Document'
    ImageIndex = 5
    ShortCut = 16464
    Control = DBRichViewEdit1
  end
  object rvActionQuickPrint1: TrvActionQuickPrint
    Category = 'RVE File'
    Caption = '&Print'
    Hint = 'Quick Print'
    ImageIndex = 6
    Control = DBRichViewEdit1
  end
  object rvActionPageSetup1: TrvActionPageSetup
    Category = 'RVE File'
    Caption = 'Pa&ge Setup...'
    Hint = 'Page Setup'
    ImageIndex = 7
  end
  object rvActionCut1: TrvActionCut
    Category = 'RVE Edit'
    Caption = 'Cu&t'
    Hint = 'Cut'
    ImageIndex = 8
    ShortCut = 16472
    Control = DBRichViewEdit1
  end
  object rvActionCopy1: TrvActionCopy
    Category = 'RVE Edit'
    Caption = '&Copy'
    Hint = 'Copy'
    ImageIndex = 9
    ShortCut = 16451
    Control = DBRichViewEdit1
  end
  object rvActionPaste1: TrvActionPaste
    Category = 'RVE Edit'
    Caption = '&Paste'
    Hint = 'Paste'
    ImageIndex = 10
    ShortCut = 16470
    Control = DBRichViewEdit1
  end
  object rvActionPasteAsText1: TrvActionPasteAsText
```

```
    Category = 'RVE Edit'
    Caption = 'Paste as &Text'
    Hint = 'Past As Text'
    ImageIndex = 11
    ShortCut = 24662
    Control = DBRichViewEdit1
  end
  object rvActionPasteSpecial1: TrvActionPasteSpecial
    Category = 'RVE Edit'
    Caption = 'Paste &Special...'
    Hint = 'Paste Special'
    ImageIndex = 86
    Control = DBRichViewEdit1
  end
  object rvActionUndo1: TrvActionUndo
    Category = 'RVE Edit'
    Caption = '&Undo'
    Hint = 'Undo'
    ImageIndex = 12
    ShortCut = 16474
    Control = DBRichViewEdit1
  end
  object rvActionRedo1: TrvActionRedo
    Category = 'RVE Edit'
    Caption = '&Redo'
    Hint = 'Redo'
    ImageIndex = 13
    ShortCut = 16473
    Control = DBRichViewEdit1
  end
  object rvActionSelectAll1: TrvActionSelectAll
    Category = 'RVE Edit'
    Caption = 'Select &All'
    Hint = 'Select All'
    ShortCut = 16449
    Control = DBRichViewEdit1
  end
  object rvActionFind1: TrvActionFind
    Category = 'RVE Edit'
    Caption = '&Find...'
    Hint = 'Find'
    ImageIndex = 14
    ShortCut = 16454
    Control = DBRichViewEdit1
  end
  object rvActionFindNext1: TrvActionFindNext
    Category = 'RVE Edit'
    Caption = 'Find &Next'
    Hint = 'Find Next'
    ImageIndex = 15
    ShortCut = 114
    Control = DBRichViewEdit1
```

```
    end
    object rvActionReplace1: TrvActionReplace
      Category = 'RVE Edit'
      Caption = '&Replace...'
      Hint = 'Replace'
      ImageIndex = 16
      ShortCut = 16456
      Control = DBRichViewEdit1
    end
    object rvActionCharCase1: TrvActionCharCase
      Category = 'RVE Edit'
      Caption = 'Character Case'
      Hint = 'Change Characer Case'
      ShortCut = 8306
      Control = DBRichViewEdit1
    end
    object rvActionFonts1: TrvActionFonts
      Category = 'RVE Text'
      Caption = '&Font...'
      Hint = 'Fonts'
      ImageIndex = 17
      Control = DBRichViewEdit1
    end
    object rvActionFontEx1: TrvActionFontEx
      Category = 'RVE Text'
      Caption = '&Font...'
      ImageIndex = 18
      Control = DBRichViewEdit1
    end
    object rvActionFontBold1: TrvActionFontBold
      Category = 'RVE Text'
      Caption = '&Bold'
      Hint = 'Bold'
      ImageIndex = 19
      ShortCut = 16450
      Control = DBRichViewEdit1
    end
    object rvActionFontItalic1: TrvActionFontItalic
      Category = 'RVE Text'
      Caption = '&Italic'
      Hint = 'Italic'
      ImageIndex = 20
      ShortCut = 16457
      Control = DBRichViewEdit1
    end
    object rvActionFontUnderline1: TrvActionFontUnderline
      Category = 'RVE Text'
      Caption = '&Underline'
      Hint = 'Underline'
      ImageIndex = 21
      ShortCut = 16469
      Control = DBRichViewEdit1
```

```
    end
    object rvActionFontStrikeout1: TrvActionFontStrikeout
      Category = 'RVE Text'
      Caption = '&Strike out'
      Hint = 'Strikeout'
      ImageIndex = 22
      Control = DBRichViewEdit1
    end
    object rvActionFontGrow1: TrvActionFontGrow
      Category = 'RVE Text'
      Caption = '&Grow Font'
      Hint = 'Grow Font'
      ImageIndex = 23
      Control = DBRichViewEdit1
    end
    object rvActionFontShrink1: TrvActionFontShrink
      Category = 'RVE Text'
      Caption = 'S&hrink Font'
      Hint = 'Shrink Font'
      ImageIndex = 24
      Control = DBRichViewEdit1
    end
    object rvActionFontGrowOnePoint1: TrvActionFontGrowOnePoint
      Category = 'RVE Text'
      Caption = 'G&row Font By One Point'
      ImageIndex = 25
      Control = DBRichViewEdit1
    end
    object rvActionFontShrinkOnePoint1: TrvActionFontShrinkOnePoint
      Category = 'RVE Text'
      Caption = 'Shri&nk Font By One Point'
      ImageIndex = 26
      Control = DBRichViewEdit1
    end
    object rvActionFontAllCaps1: TrvActionFontAllCaps
      Category = 'RVE Text'
      Caption = '&All Capitals'
      Hint = 'All Caps'
      ImageIndex = 27
      Control = DBRichViewEdit1
    end
    object rvActionFontOverline1: TrvActionFontOverline
      Category = 'RVE Text'
      Caption = '&Overline'
      Hint = 'Overline'
      ImageIndex = 28
      Control = DBRichViewEdit1
    end
    object rvActionFontColor1: TrvActionFontColor
      Category = 'RVE Text'
      Caption = 'Text &Color...'
      Hint = 'Font Color'
```

```
    ImageIndex = 29
    Control = DBRichViewEdit1
    CallerControl = ToolButton29
    UserInterface = rvacColorDialog
  end
  object rvActionFontBackColor1: TrvActionFontBackColor
    Category = 'RVE Text'
    Caption = 'Text Bac&kground Color...'
    Hint = 'Highlight'
    ImageIndex = 30
    Control = DBRichViewEdit1
    CallerControl = DBRichViewEdit1
    UserInterface = rvacColorDialog
  end
  object rvActionSubscript1: TrvActionSubscript
    Category = 'RVE Text'
    Caption = 'Subscript'
    Hint = 'Subscript'
    ImageIndex = 31
    Control = DBRichViewEdit1
  end
  object rvActionSuperscript1: TrvActionSuperscript
    Category = 'RVE Text'
    Caption = 'Superscript'
    Hint = 'Superscript'
    ImageIndex = 32
    Control = DBRichViewEdit1
  end
  object rvActionParagraph1: TrvActionParagraph
    Category = 'RVE Paragraph'
    Caption = '&Paragraph...'
    Hint = 'Paragraph'
    ImageIndex = 33
    Control = DBRichViewEdit1
  end
  object rvActionParaBorder1: TrvActionParaBorder
    Category = 'RVE Paragraph'
    Caption = 'Paragraph &Borders and Background...'
    Hint = 'Paragraph Border'
    ImageIndex = 34
    Control = DBRichViewEdit1
  end
  object rvActionWordWrap1: TrvActionWordWrap
    Category = 'RVE Paragraph'
    Caption = '&Word Wrap'
    ImageIndex = 35
    Control = DBRichViewEdit1
  end
  object rvActionAlignLeft1: TrvActionAlignLeft
    Category = 'RVE Paragraph'
    Caption = 'Align &Left'
    Hint = 'Align Left'
```

```
    ImageIndex = 36
    Control = DBRichViewEdit1
  end
  object rvActionAlignRight1: TrvActionAlignRight
    Category = 'RVE Paragraph'
    Caption = 'Align &Right'
    Hint = 'Align Right'
    ImageIndex = 37
    Control = DBRichViewEdit1
  end
  object rvActionAlignCenter1: TrvActionAlignCenter
    Category = 'RVE Paragraph'
    Caption = 'Align &Center'
    Hint = 'Align Center'
    ImageIndex = 38
    Control = DBRichViewEdit1
  end
  object rvActionAlignJustify1: TrvActionAlignJustify
    Category = 'RVE Paragraph'
    Caption = '&Justify'
    Hint = 'Align Justified'
    ImageIndex = 39
    Control = DBRichViewEdit1
  end
  object rvActionIndentInc1: TrvActionIndentInc
    Category = 'RVE Paragraph'
    Caption = '&Increase Indent'
    Hint = 'Increase Indent'
    ImageIndex = 40
    Control = DBRichViewEdit1
    IndentMax = 200
  end
  object rvActionIndentDec1: TrvActionIndentDec
    Category = 'RVE Paragraph'
    Caption = '&Decrease Indent'
    Hint = 'Decrease Indent'
    ImageIndex = 41
    Control = DBRichViewEdit1
  end
  object rvActionParaColor1: TrvActionParaColor
    Category = 'RVE Paragraph'
    Caption = 'Paragraph &Background Color...'
    Hint = 'Paragraph Color'
    ImageIndex = 42
    Control = DBRichViewEdit1
    UserInterface = rvacColorDialog
  end
  object rvActionLineSpacing1001: TrvActionLineSpacing100
    Category = 'RVE Paragraph'
    Caption = '&Single Line Spacing'
    Hint = 'Line Spacing 1'
    ImageIndex = 43
```

```
  Control = DBRichViewEdit1
end
object rvActionLineSpacing1501: TrvActionLineSpacing150
  Category = 'RVE Paragraph'
  Caption = '1.5 Li&ne Spacing'
  Hint = 'Line Spacing 1.5'
  ImageIndex = 44
  Control = DBRichViewEdit1
end
object rvActionLineSpacing2001: TrvActionLineSpacing200
  Category = 'RVE Paragraph'
  Caption = '&Double Line Spacing'
  Hint = 'Line Spacing 2'
  ImageIndex = 45
  Control = DBRichViewEdit1
end
object rvActionClearLeft1: TrvActionClearLeft
  Category = 'RVE Paragraph'
  Caption = 'Clear Text Flow at &Left Side'
  Control = DBRichViewEdit1
end
object rvActionClearRight1: TrvActionClearRight
  Category = 'RVE Paragraph'
  Caption = 'Clear Text Flow at &Right Side'
  Control = DBRichViewEdit1
end
object rvActionClearBoth1: TrvActionClearBoth
  Category = 'RVE Paragraph'
  Caption = 'Clear Text Flow at &Both Sides'
  Control = DBRichViewEdit1
end
object rvActionClearNone1: TrvActionClearNone
  Category = 'RVE Paragraph'
  Caption = '&Normal Text Flow'
  Control = DBRichViewEdit1
end
object rvActionParaList1: TrvActionParaList
  Category = 'RVE List'
  Caption = '&Bullets and Numbering...'
  Hint = 'Paragraph List'
  ImageIndex = 46
  Control = DBRichViewEdit1
end
object rvActionParaBullets1: TrvActionParaBullets
  Category = 'RVE List'
  Caption = '&Bullets'
  Hint = 'Bullets'
  ImageIndex = 47
  Control = DBRichViewEdit1
end
object rvActionParaNumbering1: TrvActionParaNumbering
  Category = 'RVE List'
```

```
    Caption = '&Numbering'
    Hint = 'Numbering'
    ImageIndex = 48
    Control = DBRichViewEdit1
  end
  object rvActionTextRTL1: TrvActionTextRTL
    Category = 'RVE BiDi'
    Caption = 'Right To Left Text'
    Control = DBRichViewEdit1
  end
  object rvActionTextLTR1: TrvActionTextLTR
    Category = 'RVE BiDi'
    Caption = 'Left To Right Text'
    Control = DBRichViewEdit1
  end
  object rvActionParaRTL1: TrvActionParaRTL
    Category = 'RVE BiDi'
    Caption = 'Right To Left'
    ImageIndex = 49
    Control = DBRichViewEdit1
  end
  object rvActionParaLTR1: TrvActionParaLTR
    Category = 'RVE BiDi'
    Caption = 'Left To Right'
    ImageIndex = 50
    Control = DBRichViewEdit1
  end
  object rvActionInsertFile1: TrvActionInsertFile
    Category = 'RVE Insert'
    Caption = '&File...'
    Hint = 'Import File'
    ImageIndex = 84
    Control = DBRichViewEdit1
  end
  object rvActionInsertPicture1: TrvActionInsertPicture
    Category = 'RVE Insert'
    Caption = '&Picture...'
    Hint = 'Insert Picture'
    ImageIndex = 51
    Control = DBRichViewEdit1
    VAlign = rvvaBaseline
    DefaultExt = 'bmp'
  end
  object rvActionInsertHLine1: TrvActionInsertHLine
    Category = 'RVE Insert'
    Caption = '&Horizontal Line'
    Hint = 'Insert Line'
    ImageIndex = 52
    Control = DBRichViewEdit1
  end
  object rvActionInsertHyperlink1: TrvActionInsertHyperlink
    Category = 'RVE Insert'
```

```
    Caption = 'Hypertext &Link...'
    Hint = 'Insert Hyperlink'
    ImageIndex = 53
    Control = DBRichViewEdit1
    SpaceFiller = ' '
  end
  object rvActionInsertSymbol1: TrvActionInsertSymbol
    Category = 'RVE Insert'
    Caption = '&Symbol...'
    Hint = 'Insert Symbol'
    ImageIndex = 54
    Control = DBRichViewEdit1
    AlwaysInsertUnicode = False
  end
  object rvActionInsertText1: TrvActionInsertText
    Category = 'RVE Insert'
    Caption = 'rvActionInsertText1'
    Control = DBRichViewEdit1
  end
  object rvActionColor1: TrvActionColor
    Category = 'RVE Miscellaneous'
    Caption = 'Background &Color...'
    Hint = 'Background|Changes a background color of the document'
    ImageIndex = 55
    Control = DBRichViewEdit1
    Color = clWindow
  end
  object rvActionBackground1: TrvActionBackground
    Category = 'RVE Miscellaneous'
    Caption = '&Background...'
    Hint = 'Change Background'
    ImageIndex = 56
    Control = DBRichViewEdit1
  end
  object rvActionFillColor1: TrvActionFillColor
    Category = 'RVE Miscellaneous'
    Caption = '&Fill Color...'
    Control = DBRichViewEdit1
  end
  object rvActionInsertPageBreak1: TrvActionInsertPageBreak
    Category = 'RVE Miscellaneous'
    Caption = '&Insert Page Break'
    Hint = 'Insert Page Break'
    ImageIndex = 57
    ShortCut = 16397
    Control = DBRichViewEdit1
  end
  object rvActionRemovePageBreak1: TrvActionRemovePageBreak
    Category = 'RVE Miscellaneous'
    Caption = '&Remove Page Break'
    Control = DBRichViewEdit1
  end
```

```
    object rvActionRemoveHyperlinks1: TrvActionRemoveHyperlinks
      Category = 'RVE Miscellaneous'
      Caption = '&Remove Hyperlinks'
      Hint = 'Remove Hyperlink'
      ImageIndex = 89
      Control = DBRichViewEdit1
    end
    object rvActionItemProperties1: TrvActionItemProperties
      Category = 'RVE Miscellaneous'
      Caption = 'Object &Properties...'
      Hint = 'Properties'
      ImageIndex = 58
      ShortCut = 32781
      Control = DBRichViewEdit1
    end
    object rvActionVAlign1: TrvActionVAlign
      Category = 'RVE Miscellaneous'
      Caption = '&Object Position...'
      Hint = 'Align'
      ImageIndex = 59
      Control = DBRichViewEdit1
    end
    object rvActionShowSpecialCharacters1:
TrvActionShowSpecialCharacters
      Category = 'RVE Miscellaneous'
      Caption = '&Nonprinting Characters'
      Hint = 'Show Special Characters'
      ImageIndex = 60
      Control = DBRichViewEdit1
    end
    object rvActionInsertTable1: TrvActionInsertTable
      Category = 'RVE Table'
      Caption = '&Insert Table...'
      Hint = 'Insert Table'
      ImageIndex = 61
    end
    object rvActionTableInsertRowsBelow1: TrvActionTableInsertRowsBelow
      Category = 'RVE Table'
      Caption = 'Insert Row &Below'
      Hint = 'Insert Row Below'
      ImageIndex = 62
      Control = DBRichViewEdit1
      AllowMultiple = True
    end
    object rvActionTableInsertRowsAbove1: TrvActionTableInsertRowsAbove
      Category = 'RVE Table'
      Caption = 'Insert Row &Above'
      Hint = 'Insert Row Above'
      ImageIndex = 63
      Control = DBRichViewEdit1
    end
    object rvActionTableInsertColLeft1: TrvActionTableInsertColLeft
```

```
    Category = 'RVE Table'
    Caption = 'Insert Column &Left'
    Hint = 'Insert Column Left'
    ImageIndex = 64
    Control = DBRichViewEdit1
  end
  object rvActionTableInsertColRight1: TrvActionTableInsertColRight
    Category = 'RVE Table'
    Caption = 'Insert Column &Right'
    Hint = 'Insert Column Right'
    ImageIndex = 65
    Control = DBRichViewEdit1
  end
  object rvActionTableDeleteRows1: TrvActionTableDeleteRows
    Category = 'RVE Table'
    Caption = 'Delete R&ows'
    Hint = 'Delete Row(s)'
    ImageIndex = 66
    Control = DBRichViewEdit1
  end
  object rvActionTableDeleteCols1: TrvActionTableDeleteCols
    Category = 'RVE Table'
    Caption = 'Delete &Columns'
    Hint = 'Delete Column(s)'
    ImageIndex = 67
    Control = DBRichViewEdit1
  end
  object rvActionTableDeleteTable1: TrvActionTableDeleteTable
    Category = 'RVE Table'
    Caption = '&Delete Table'
    Hint = 'Delete Table'
    ImageIndex = 68
    Control = DBRichViewEdit1
  end
  object rvActionTableMergeCells1: TrvActionTableMergeCells
    Category = 'RVE Table'
    Caption = '&Merge Cells'
    Hint = 'Merge Cells'
    ImageIndex = 69
    Control = DBRichViewEdit1
  end
  object rvActionTableSplitCells1: TrvActionTableSplitCells
    Category = 'RVE Table'
    Caption = '&Split Cells...'
    Hint = 'Split Cells'
    ImageIndex = 70
    Control = DBRichViewEdit1
  end
  object rvActionTableSelectTable1: TrvActionTableSelectTable
    Category = 'RVE Table'
    Caption = 'Select &Table'
    ShortCut = 32780
```

```
      Control = DBRichViewEdit1
    end
    object rvActionTableSelectRows1: TrvActionTableSelectRows
      Category = 'RVE Table'
      Caption = 'Select Ro&ws'
      Control = DBRichViewEdit1
    end
    object rvActionTableSelectCols1: TrvActionTableSelectCols
      Category = 'RVE Table'
      Caption = 'Select Col&umns'
      Control = DBRichViewEdit1
    end
    object rvActionTableSelectCell1: TrvActionTableSelectCell
      Category = 'RVE Table'
      Caption = 'Select C&ell'
      Control = DBRichViewEdit1
    end
    object rvActionTableCellVAlignTop1: TrvActionTableCellVAlignTop
      Category = 'RVE Table'
      Caption = 'Align Cell To The &Top'
      Hint = 'Align Cell - Top'
      ImageIndex = 71
      Control = DBRichViewEdit1
    end
    object rvActionTableCellVAlignMiddle1:
TrvActionTableCellVAlignMiddle
      Category = 'RVE Table'
      Caption = 'Align Cell To The &Middle'
      Hint = 'Align Cell - Middle'
      ImageIndex = 72
      Control = DBRichViewEdit1
    end
    object rvActionTableCellVAlignBottom1:
TrvActionTableCellVAlignBottom
      Category = 'RVE Table'
      Caption = 'Align Cell To The &Bottom'
      Hint = 'Align Cell - Bottom'
      ImageIndex = 73
      Control = DBRichViewEdit1
    end
    object rvActionTableCellVAlignDefault1:
TrvActionTableCellVAlignDefault
      Category = 'RVE Table'
      Caption = '&Default Cell Vertical Alignment'
      Control = DBRichViewEdit1
    end
    object rvActionTableCellLeftBorder1: TrvActionTableCellLeftBorder
      Category = 'RVE Table'
      Caption = '&Left Border'
      Hint = 'Cell Border - Left'
      ImageIndex = 74
      Control = DBRichViewEdit1
```

```
    end
    object rvActionTableCellRightBorder1: TrvActionTableCellRightBorder
      Category = 'RVE Table'
      Caption = '&Right Border'
      Hint = 'Cell Border - Right'
      ImageIndex = 75
      Control = DBRichViewEdit1
    end
    object rvActionTableCellTopBorder1: TrvActionTableCellTopBorder
      Category = 'RVE Table'
      Caption = '&Top Border'
      Hint = 'Cell Border - Top'
      ImageIndex = 76
      Control = DBRichViewEdit1
    end
    object rvActionTableCellBottomBorder1:
TrvActionTableCellBottomBorder
      Category = 'RVE Table'
      Caption = '&Bottom Border'
      Hint = 'Cell Border - Bottom'
      ImageIndex = 77
      Control = DBRichViewEdit1
    end
    object rvActionTableCellAllBorders1: TrvActionTableCellAllBorders
      Category = 'RVE Table'
      Caption = '&All Borders'
      Hint = 'Cell Border - All'
      ImageIndex = 78
      Control = DBRichViewEdit1
    end
    object rvActionTableCellNoBorders1: TrvActionTableCellNoBorders
      Category = 'RVE Table'
      Caption = '&No Borders'
      Hint = 'Cell Border - None'
      ImageIndex = 79
      Control = DBRichViewEdit1
    end
    object rvActionTableProperties1: TrvActionTableProperties
      Category = 'RVE Table'
      Caption = 'Table &Properties...'
      Hint = 'Table Properties'
      ImageIndex = 80
      Control = DBRichViewEdit1
    end
    object rvActionTableGrid1: TrvActionTableGrid
      Category = 'RVE Table'
      Caption = 'Show &Grid Lines'
      Hint = 'Table Grid'
      ImageIndex = 81
    end
    object DataSetPost1: TDataSetPost
      Category = 'Dataset'
```

```
      Caption = 'P&ost'
      Hint = 'Save to Database'
      ImageIndex = 82
    end
    object DataSetCancel1: TDataSetCancel
      Category = 'Dataset'
      Caption = '&Cancel'
      Hint = 'Cancel Changes'
      ImageIndex = 83
    end
  end
  object RVStyle1: TRVStyle
    TextStyles = <
      item
        StyleName = 'Normal text'
        FontName = 'Times New Roman'
        Size = 12
        Unicode = True
      end
      item
        StyleName = 'Heading'
        FontName = 'Times New Roman'
        Size = 12
        Style = [fsBold]
        Color = clBlue
        Unicode = True
      end
      item
        StyleName = 'Subheading'
        FontName = 'Times New Roman'
        Style = [fsBold]
        Color = clNavy
        Unicode = True
      end
      item
        StyleName = 'Keywords'
        FontName = 'Times New Roman'
        Style = [fsItalic]
        Color = clMaroon
        Unicode = True
      end
      item
        StyleName = 'Jump 1'
        FontName = 'Arial'
        Style = [fsUnderline]
        Color = clGreen
        Jump = True
        Unicode = True
      end
      item
        StyleName = 'Jump 2'
        FontName = 'Arial'
```

```
        Style = [fsUnderline]
        Color = clGreen
        Jump = True
        Unicode = True
      end>
    ParaStyles = <
      item
        StyleName = 'Paragraph Style'
        Tabs = <>
      end
      item
        StyleName = 'Centered'
        Alignment = rvaCenter
        Tabs = <>
      end>
    ListStyles = <>
    InvalidPicture.Data = {
```

```
      07544269746D617036100000424D361000000000000036000000280000002000
      0000200000000100200000000000001000000000000000000000000000000000
      0000808080008080800080808000808080008080800080808000808080008080
      8000808080008080800080808000808080008080800080808000808080008080
      8000808080008080800080808000808080008080800080808000808080008080
      8000808080008080800080808000808080008080800080808000808080008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
```

```
FF00FFFFFF000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF000000FF000000FF00FFFF
FF00FFFFFF000000FF000000FF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF000000FF000000
FF000000FF000000FF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF000000
FF000000FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF000000FF000000
FF000000FF000000FF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF000000FF000000FF00FFFF
FF00FFFFFF000000FF000000FF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
```

```
      FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF008080800080808000808080008080800080808000808080008080
      800080808000808080008080800080808000808080008080800080808000FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
      FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
      8000808080008080800080808000808080008080800080808000808080008080
      8000808080008080800080808000808080008080800080808000808080008080
      8000808080008080800080808000808080008080800080808000808080008080
      8000808080008080800080808000808080008080800080808000808080008080
      8000}
    StyleTemplates = <>
    Left = 48
    Top = 120
  end
  object RVPrint1: TRVPrint
    PreviewCorrection = True
    LeftMarginMM = 20
    RightMarginMM = 20
```

```
    TopMarginMM = 20
    BottomMarginMM = 20
    Left = 200
    Top = 176
  end
  object RVAPopupMenu1: TRVAPopupMenu
    Images = ImageList1
    ActionList = ActionList1
    Left = 128
    Top = 176
  end
  object RVAControlPanel1: TRVAControlPanel
    DialogFontName = 'MS Sans Serif'
    DefaultControl = DBRichViewEdit1
    RVFFilter = 'RichView Files (*.rvf)|*.rvf'
    DefaultExt = 'rvf'
    RVFormatTitle = 'RichView Format'
    DefaultFileName = 'Untitled.rvf'
    RVPrint = RVPrint1
    Language = 'English (US)'
    Left = 184
    Top = 120
  end
  object RVOfficeConverter1: TRVOfficeConverter
    Left = 272
    Top = 176
  end
  object rd: TReplaceDialog
    Left = 344
    Top = 176
  end
end
```

CaseIndicator.pas

```
unit CaseIndicator;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, ComCtrls, DB, ZAbstractRODataset,
ZAbstractDataset, ZDataset;

type
  TCaseIndicatorForm = class(TForm)
    TreeView1:                          TTreeView;

    CaseQuery:                          TZQuery;
    CaseQueryidcase:                    TIntegerField;
    CaseQuerycase_number:               TWideStringField;
```

```
    RegExQuery:                          TZQuery;
    RegExQueryidregex_pattern:           TIntegerField;
    RegExQuerypattern_name:              TWideStringField;

    IndicatorQuery:                      TZQuery;
    IndicatorQueryunique_matched_text:   TMemoField;

    LinkQuery:                           TZQuery;
    LinkQuerycase_number:                TWideStringField;
    LinkQueryidcase:                     TIntegerField;

    procedure FormClose                 (Sender:TObject;
                                         var Action:TCloseAction);
    procedure FormCreate                (Sender: TObject);
    procedure TreeView1MouseDown        (Sender: TObject;
                                         Button: TMouseButton;
                                         Shift: TShiftState; X, Y:
Integer);
  private
    IndicatorCaseID:  Integer;
    procedure LoadTreeview;
    { Private declarations }
  public
    constructor CreateWithID
(AOwner:TApplication;CaseID:Integer);
    class procedure OpenIndicatorWithID(ID:  Integer);
    { Public declarations }
  protected
    procedure CreateParams              (var Params: TCreateParams);
override;
  end;

var
  CaseIndicatorForm: TCaseIndicatorForm;

implementation

uses Thesis_Main;

{$R *.dfm}

procedure TCaseIndicatorForm.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

constructor TCaseIndicatorForm.CreateWithID(AOwner: TApplication;
  CaseID: Integer);
begin
```

```
   inherited Create(AOwner);
   IndicatorCaseID := CaseID;
end;

procedure TCaseIndicatorForm.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
   CaseQuery.Close;
   Action := caFree;
end;

procedure TCaseIndicatorForm.FormCreate(Sender: TObject);
begin
   CaseQuery.ParamByName('id').Value := IndicatorCaseID;
   LoadTreeview;
end;

procedure TCaseIndicatorForm.LoadTreeview;
var
   rootNode, regexNode, caseNode, linkNode: TTreeNode;
   HoldQuery:  String;
begin
   CaseQuery.Open;
   Caption := Caption + ' - ' + CaseQueryCase_Number.AsString;
   RegExQuery.Open;
   Treeview1.Items.BeginUpdate;
   rootNode := Treeview1.Items.Add(nil, CaseQuerycase_number.AsString);
   while Not RegExQuery.EOF do
   begin
      regexNode := Treeview1.Items.AddChild(rootNode,
                                    RegExQuerypattern_name.AsString);
      IndicatorQuery.SQL.Clear;
      IndicatorQuery.SQL.BeginUpdate;
      IndicatorQuery.SQL.Add('select unique_matched_text');
      IndicatorQuery.SQL.Add('from unique_matched_text');
      IndicatorQuery.SQL.Add('where');
      IndicatorQuery.SQL.Add('idcase = '+ CaseQueryidCase.AsString);
      IndicatorQuery.SQL.Add('and');
      IndicatorQuery.SQL.Add('idregex_pattern = ' +
                                  RegexQueryidRegex_Pattern.AsString);
      IndicatorQuery.SQL.Add('order by');
      IndicatorQuery.SQL.Add('unique_matched_text');
      IndicatorQuery.SQL.EndUpdate;
      IndicatorQuery.Open;
      while Not IndicatorQuery.EOF do
      begin
         caseNode := Treeview1.Items.AddChild(regexNode,

IndicatorQueryunique_matched_text.AsString);
         HoldQuery := LinkQuery.SQL.Text;
         LinkQuery.SQL.Text := StringReplace(
                           LinkQuery.SQL.Text, ':txt', QuotedStr(
```

```
                        IndicatorQueryunique_matched_text.AsString),
                        [rfReplaceAll]);
         LinkQuery.SQL.Text := StringReplace(LinkQuery.SQL.Text,
                           ':cs', QuotedStr(
                           CaseQuerycase_number.AsString),
                           [rfReplaceAll]);
         LinkQuery.Open;
         while Not LinkQuery.EOF do
         begin
            linkNode := Treeview1.Items.AddChild(caseNode,
                                    LinkQuerycase_number.AsString);
            LinkNode.Data := Pointer(LinkQueryidcase.AsInteger);
            LinkQuery.Next;
         end;
         LinkQuery.Close;
         LinkQuery.SQL.Text := HoldQuery;
         IndicatorQuery.Next;
      end;
      IndicatorQuery.Close;
      RegExQuery.Next;
   end;
   Treeview1.Items.EndUpdate;
   RegExQuery.Close;
   Treeview1.Selected := Treeview1.Items.GetFirstNode;
   Treeview1.Selected.Expand(False);
end;

class procedure TCaseIndicatorForm.OpenIndicatorWithID(ID: Integer);
var
   F, Child: TCaseIndicatorForm;
   I:        Integer;
begin
   F := Nil;
   for I := Screen.FormCount - 1 DownTo 0 do
   begin
      if (Screen.Forms[I] is TCaseIndicatorForm) then
      begin
         F := Screen.Forms[I] As TCaseIndicatorForm;
         if F.CaseQueryidcase.AsInteger <> ID then
            F := Nil
         else
            Break;
      end;
   end;
   if F = Nil then
   begin
      Child := TCaseIndicatorForm.CreateWithID(Application, ID);
      Child.Show;
   end
   else
   begin
      if F.WindowState = wsMinimized then
```

```
         F.WindowState := wsNormal;
      F.BringToFront;
   end;
end;

procedure TCaseIndicatorForm.TreeView1MouseDown(Sender: TObject;
                                                Button: TMouseButton;
                                                Shift: TShiftState;
                                                X, Y: Integer);
var
   hts:  THitTests;
begin
   hts                := TreeView1.GetHitTestInfoAt(X, Y);
   Treeview1.Selected := Treeview1.GetNodeAt(X, Y);
   if Button = mbMiddle then
   begin
      if (htOnItem in hts) then
         if Treeview1.Selected.Level = 3 then
         begin
            screen.Cursor := crSQLWait;
            try
               TCaseIndicatorForm.OpenIndicatorWithID(Integer(
                                             TreeView1.Selected.Data));
            finally
               screen.Cursor := crDefault;
            end;
         end;
   end
end;

end.
```

CaseIndicator.dfm

```
object CaseIndicatorForm: TCaseIndicatorForm
  Left = 0
  Top = 0
  Caption = 'Case Indicators'
  ClientHeight = 410
  ClientWidth = 530
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCreate = FormCreate
  PixelsPerInch = 96
```

```
  TextHeight = 13
  object TreeView1: TTreeView
    Left = 0
    Top = 0
    Width = 530
    Height = 410
    Align = alClient
    Constraints.MinHeight = 410
    Constraints.MinWidth = 530
    HotTrack = True
    Indent = 19
    TabOrder = 0
    OnMouseDown = TreeView1MouseDown
  end
  object CaseQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select idcase, case_number'
      'from case_file'
      'where idcase = :id')
    Params = <
      item
        DataType = ftUnknown
        Name = 'id'
        ParamType = ptUnknown
      end>
    Left = 480
    Top = 24
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'id'
        ParamType = ptUnknown
      end>
    object CaseQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object CaseQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
  end
  object RegExQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select idregex_pattern, pattern_name'
      'from regex_pattern'
      'order by pattern_name')
    Params = <>
    Left = 480
```

```
    Top = 72
    object RegExQueryidregex_pattern: TIntegerField
      FieldName = 'idregex_pattern'
      Required = True
    end
    object RegExQuerypattern_name: TWideStringField
      FieldName = 'pattern_name'
      Required = True
      Size = 45
    end
  end
  object IndicatorQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select unique_matched_text'
      'from unique_matched_text'
      'where'
      '   idcase = :idcs and'
      '   idregex_pattern = :regid'
      'order by'
      '   unique_matched_text')
    Params = <
      item
        DataType = ftUnknown
        Name = 'idcs'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'regid'
        ParamType = ptUnknown
      end>
    Left = 480
    Top = 128
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'idcs'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'regid'
        ParamType = ptUnknown
      end>
    object IndicatorQueryunique_matched_text: TMemoField
      FieldName = 'unique_matched_text'
      Required = True
      BlobType = ftMemo
    end
  end
  object LinkQuery: TZQuery
```

```
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select case_file.idcase, case_number'
      'from unique_matched_text, case_file'
      'where '
      '   unique_matched_text.idyear = case_file.idyear and'
      '   unique_matched_text.idcase = case_file.idcase and'
      '   unique_matched_text = :txt and'
      '   case_number <> :cs'
      'order by'
      '   case_number')
    Params = <
      item
        DataType = ftUnknown
        Name = 'txt'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'cs'
        ParamType = ptUnknown
      end>
    Left = 480
    Top = 192
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'txt'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'cs'
        ParamType = ptUnknown
      end>
    object LinkQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object LinkQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
  end
end
```

AllIndicator.pas

```
unit AllIndicator;
```

```
interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ComCtrls, ZAbstractRODataset,
ZAbstractDataset, ZDataset, StdCtrls, ExtCtrls, Buttons;

type
  TIndicator_Form = class(TForm)
    TreeView1:                      TTreeView;

    CaseQuery:                      TZQuery;
    CaseQueryidyear:                TIntegerField;
    CaseQueryidcase:                TIntegerField;
    CaseQuerycase_number:           TWideStringField;

    MatchQuery:                     TZQuery;
    MatchQueryunique_matched_text: TMemoField;
    MatchQueryidregex_pattern:      TIntegerField;

    RegexQuery: TZQuery;
    RegexQueryidregex_pattern:      TIntegerField;
    RegexQuerypattern_name:         TWideStringField;

    LinkCaseQuery:                  TZQuery;
    LinkCaseQuerycase_number:       TWideStringField;
    LinkCaseQueryidcase:            TIntegerField;

    Panel1:                         TPanel;
    Edit1:                          TEdit;
    Label1:                         TLabel;
    SpeedButton1:                   TSpeedButton;

    procedure FormCreate          (Sender: TObject);
    procedure FormClose           (Sender: TObject;
                                   var Action: TCloseAction);
    procedure TreeView1Expanding  (Sender: TObject; Node: TTreeNode;
                                   var AllowExpansion: Boolean);
    procedure TreeView1MouseDown  (Sender: TObject;
                                   Button: TMouseButton;
                                   Shift: TShiftState; X, Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
  private
    { Private declarations }
    procedure LoadIndicatorTreeView;
  protected
    procedure CreateParams        (var Params: TCreateParams);
override;
  public
    { Public declarations }
  end;

var
```

```
  Indicator_Form: TIndicator_Form;

implementation

uses Thesis_Main, CaseIndicator;

{$R *.dfm}

procedure TIndicator_Form.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

procedure TIndicator_Form.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   Action := caFree;
end;

procedure TIndicator_Form.FormCreate(Sender: TObject);
begin
   RegexQuery.Open;
   LoadIndicatorTreeView;
   RegexQuery.Close;
end;

procedure TIndicator_Form.LoadIndicatorTreeView;
var
   rootNode, regexNode:  TTreeNode;
begin
   TreeView1.Items.BeginUpdate;
   Treeview1.Items.Clear;
   rootNode := Treeview1.Items.AddFirst(nil,
                                    'Open Cases To View Indicators');
   RegExQuery.First;
   while Not RegExQuery.EOF do
   begin
      regexNode := Treeview1.Items.AddChild(rootNode,
                                     RegExQuerypattern_name.AsString);
      regexNode.Data := pointer(RegExQueryidRegex_pattern.AsInteger);
      Treeview1.Items.AddChild(regexNode, 'TMP');
      RegExQuery.Next;
   end;
   Treeview1.Selected := Treeview1.Items.GetFirstNode;
   Treeview1.Selected.Expand(False);
   Treeview1.Items.EndUpdate;
end;

procedure TIndicator_Form.SpeedButton1Click(Sender: TObject);
begin
```

```
   RegexQuery.Open;
   LoadIndicatorTreeView;
   RegexQuery.Close;
end;

procedure TIndicator_Form.TreeView1Expanding(Sender: TObject;
                                             Node: TTreeNode;
                                             var AllowExpansion:
                                             Boolean);
var
   tn, MatchNode, CaseNode:  TTreeNode;
begin
   tn := Node.getFirstChild;
   if ((tn <> nil) and (tn.Text = 'TMP')) then
   begin
      Treeview1.Items.BeginUpdate;
      screen.Cursor := crSQLWait;
      try
         tn.Delete;
         MatchQuery.SQL.Clear;
         MatchQuery.SQL.BeginUpdate;
         MatchQuery.SQL.Add('select');
         MatchQuery.SQL.Add('distinct '+
                            'unique_matched_text,idregex_pattern');
         MatchQuery.SQL.Add('from unique_matched_text');
         MatchQuery.SQL.Add('where idregex_pattern = ');
         MatchQuery.SQL.Add(IntToStr(Integer(Node.Data)));
         if Edit1.Text <> '' then
         begin
            MatchQuery.SQL.Add('and unique_matched_text like ');
            if (Pos('%', Edit1.Text) = 0) then
               MatchQuery.SQL.Add('"%'+Edit1.Text+'%"')
            else
               MatchQuery.SQL.Add('"'+Edit1.Text+'"');
         end;
         MatchQuery.SQL.Add('order by unique_matched_text');
         MatchQuery.SQL.EndUpdate;
         MatchQuery.Open;
         while Not MatchQuery.EOF do
         begin
            MatchNode := Treeview1.Items.AddChild(Node,
                         MatchQueryunique_matched_text.AsString);
            LinkCaseQuery.SQL.Clear;
            LinkCaseQuery.SQL.BeginUpdate;
            LinkCaseQuery.SQL.Add('select case_file.idcase, '+
                                  'case_number');
            LinkCaseQuery.SQL.Add('from unique_matched_text, '+
                                  'case_file');
            LinkCaseQuery.SQL.Add('where');
            LinkCaseQuery.SQL.Add('unique_matched_text.idcase = '+
                                  case_file.idcase and');
            LinkCaseQuery.SQL.Add('unique_matched_text.idyear = '+
```

```
                                'case_file.idyear and');
            LinkCaseQuery.SQL.Add('unique_matched_text = ');
            LinkCaseQuery.SQL.Add(quotedstr(MatchNode.Text));
            LinkCaseQuery.SQL.Add('order by case_number');
            LinkCaseQuery.SQL.EndUpdate;
            LinkCaseQuery.Open;
            while Not LinkCaseQuery.EOF do
            begin
               CaseNode := Treeview1.Items.AddChild(MatchNode,
                                  LinkCaseQueryCase_Number.AsString);
               CaseNode.Data := Pointer(LinkCaseQueryidcase.AsInteger);
               LinkCaseQuery.Next;
            end;
            LinkCaseQuery.Close;
            MatchQuery.Next;
         end;
         MatchQuery.Close;
         Treeview1.Items.EndUpdate;
      finally
         screen.Cursor := crDefault;
      end;
   end;
end;

procedure TIndicator_Form.TreeView1MouseDown(Sender: TObject;
                                             Button: TMouseButton;
                                             Shift: TShiftState; X, Y:
Integer);
var
   hts:  THitTests;
begin
   hts                := TreeView1.GetHitTestInfoAt(X, Y);
   Treeview1.Selected := Treeview1.GetNodeAt(X, Y);
   if Button = mbMiddle then
   begin
      if (htOnItem in hts) then
         if Treeview1.Selected.Level = 3 then
         begin
            screen.Cursor := crSQLWait;
            try
               TCaseIndicatorForm.OpenIndicatorWithID(Integer(
                                             TreeView1.Selected.Data));
            finally
               screen.Cursor := crDefault;
            end;
         end;
   end
end;

end.
```

AllIndicator.dfm

```
object Indicator_Form: TIndicator_Form
  Left = 0
  Top = 0
  ActiveControl = TreeView1
  Caption = 'Indicators'
  ClientHeight = 392
  ClientWidth = 634
  Color = clBtnFace
  Constraints.MinHeight = 350
  Constraints.MinWidth = 650
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 13
  object TreeView1: TTreeView
    Left = 0
    Top = 0
    Width = 634
    Height = 338
    Align = alClient
    HotTrack = True
    Indent = 19
    ReadOnly = True
    TabOrder = 0
    OnExpanding = TreeView1Expanding
    OnMouseDown = TreeView1MouseDown
    ExplicitHeight = 260
  end
  object Panel1: TPanel
    Left = 0
    Top = 338
    Width = 634
    Height = 54
    Align = alBottom
    BevelEdges = [beTop]
    BevelKind = bkTile
    BevelOuter = bvLowered
    TabOrder = 1
    ExplicitTop = 260
    DesignSize = (
      634
      52)
    object Label1: TLabel
```

```
    Left = 24
    Top = 8
    Width = 69
    Height = 13
    Caption = 'Filter Indictors'
  end
  object SpeedButton1: TSpeedButton
    Left = 586
    Top = 22
    Width = 23
    Height = 22
    Hint = 'Reset Indicator View'
    Anchors = [akRight, akBottom]
    Glyph.Data = {
      36030000424D3603000000000000360000002800000010000000100000000100
      1800000000000003000061 0B0000610B0000000000000000000000FF00FFFF00FF
      B89F8E4E2E1C4E2E1C4E2E1C4E2E1C4E2E1C4E2E1C4E2E1C4E2E1C4E2E1C4E2E
      1C4E2E1C4E2E1CFF00FFFF00FFFF00FFB89F8EE9DED8A48B7AA48B7AA48B7AA4
      8B7AA48B7AA48B7AA48B7AA48B7AA48B7AA48B7A4E2E1CFF00FFFF00FFFF00FF
      B89F8EEDE3DEE9DED8E5D9D3E2D4CCDECEC6DAC9C0D7C4BAD3BFB4D0BAADD0BA
      ADA48B7A4E2E1CFF00FFFF00FFFF00FFB89F8EF0E9E5EDE3DE008100E5D9D329
      9D27299D2756B453D7C4BAD3BFB4D0BAADA48B7A4E2E1CFF00FFFF00FFFF00FF
      B89F8EF3EEEBF0E9E500810000810000810000810000810056B453D7C4BAD3BF
      B4A48B7A4E2E1CFF00FFFF00FFFF00FFB89F8EF7F3F2F3EEEB00810033BD6B00
      8100E5D9D3E2D4CC00810056B453D7C4BAA48B7A4E2E1CFF00FFFF00FFFF00FF
      B89F8EFBFAF8F7F3F200810000810000810000810000810E5D9D3E2D4CC008100DAC9
      C0A48B7A4E2E1CFF00FFFF00FFFF00FFB89F8EFFFFFFFFBFAF8F7F3F2F3EEEBF0
      E9E5EDE3DEE9DED8E5D9D3E2D4CCDECEC6A48B7A4E2E1CFF00FFFF00FFFF00FF
      B89F8EFFFFFF008100FCFCFCFAF7F600810000810000810000810000810E7DCD5E3D7
      CFA48B7A4E2E1CFF00FFFF00FFFF00FFB89F8EFFFFFFADD59D008100FCFCFCFA
      F7F600810033BD6B008100EAE1DCE7DCD5A48B7A4E2E1CFF00FFFF00FFFF00FF
      B89F8EFFFFFFFFFFFFADD59D00810000810000810000810000810000810EEE6E2C9AD
      9DA48B7A4E2E1CFF00FFFF00FFFF00FFB89F8EFFFFFFFFFFFFFFFFFFADD59D29
      9D27299D27FAF7F6008100B89F8E4E2E1C4E2E1C4E2E1CFF00FFFF00FFFF00FF
      B89F8EFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCFCFCFAF7F6B89F8EEFE5
      DED0BAAD4E2E1CFF00FFFF00FFFF00FFB89F8EFFFFFFFFFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFCFCFCB89F8ED0BAAD4E2E1CFF00FFFF00FFFF00FFFF00FF
      B89F8EFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB89F8E4E2E
      1CFF00FFFF00FFFF00FFFF00FFFF00FFB89F8EB89F8EB89F8EB89F8EB89F8EB8
      9F8EB89F8EB89F8EB89F8EB89F8EFF00FFFF00FFFF00FFFF00FF}
    ParentShowHint = False
    ShowHint = True
    OnClick = SpeedButton1Click
    ExplicitLeft = 592
    ExplicitTop = 24
  end
  object Edit1: TEdit
    Left = 24
    Top = 25
    Width = 537
    Height = 21
    Anchors = [akLeft, akTop, akRight]
    TabOrder = 0
  end
```

```
  end
  object CaseQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'Select '
      '   idyear,'
      '   idcase,'
      '   case_number'
      'from'
      '   case_file'
      'order by'
      '   case_number desc')
    Params = <>
    Left = 568
    Top = 8
    object CaseQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object CaseQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object CaseQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
  end
  object RegexQuery: TZQuery
    Connection = Scanner_Main_Form.ZConnection1
    SQL.Strings = (
      'select'
      '   idregex_pattern,'
      '   pattern_name'
      'from'
      '   regex_pattern'
      'order by'
      '   pattern_name')
    Params = <>
    Left = 568
    Top = 56
    object RegexQueryidregex_pattern: TIntegerField
      FieldName = 'idregex_pattern'
      Required = True
    end
    object RegexQuerypattern_name: TWideStringField
      FieldName = 'pattern_name'
      Required = True
      Size = 45
    end
  end
```

```
object MatchQuery: TZQuery
  Connection = Scanner_Main_Form.ZConnection1
  SQL.Strings = (
    'select distinct(unique_matched_text), idregex_pattern'
    'from unique_matched_text'
    'where idregex_pattern = :regexid'
    'order by unique_matched_text')
  Params = <
    item
      DataType = ftUnknown
      Name = 'regexid'
      ParamType = ptUnknown
    end>
  Left = 568
  Top = 104
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'regexid'
      ParamType = ptUnknown
    end>
  object MatchQueryunique_matched_text: TMemoField
    FieldName = 'unique_matched_text'
    Required = True
    BlobType = ftMemo
  end
  object MatchQueryidregex_pattern: TIntegerField
    FieldName = 'idregex_pattern'
    Required = True
  end
end
object LinkCaseQuery: TZQuery
  Connection = Scanner_Main_Form.ZConnection1
  SQL.Strings = (
    'select case_file.idcase, case_number'
    'from unique_matched_text, case_file'
    'where'
    '   unique_matched_text.idcase = case_file.idcase and'
    '   unique_matched_text.idyear = case_file.idyear and'
    '   unique_matched_text = :txt')
  Params = <
    item
      DataType = ftUnknown
      Name = 'txt'
      ParamType = ptUnknown
    end>
  Left = 568
  Top = 152
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'txt'
```

```
        ParamType = ptUnknown
      end>
    object LinkCaseQuerycase_number: TWideStringField
      FieldName = 'case_number'
      Required = True
      Size = 10
    end
    object LinkCaseQueryidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
  end
end
```

The Case Viewer application was developed using Delphi Professional 2010, however; to recompile this application, additional components are needed. To connect to the MySQL database, the ZEOS database components version 7.0 were used. These components can be found here: http://zeos.firmos.at/. A database aware version of a date and time picker was utilized that can be found here: http://delphi.about.com/library/bluc/ucvcl.htm. Furthermore, the wordprocessing capabilities built into the application were possible through the use of the TRichView components Version 12.0 that can be found here: http://www.trichview.com/, with an added capability to count the number of paragraphs, words, letters contained within a document, which needs RVWordEnum.pas which can be downloaded from http://www.trichview.com/resources/spell/rvspell.zip

Appendix D – Regular Expression Parser Application Source Code

Expressions.dpr

```
program Expressions;

uses
  Forms,
  Windows,
  dialogs,
  RegExMain in 'RegExMain.pas' {Regular_Expression_Form},
  Filter in 'Filter.pas' {FilterForm},
  XML in 'XML.pas' {XMLForm};

var
   GMutex, LMutex:  THandle;
{$R *.res}

begin
   GMutex := OpenMutex(MUTEX_ALL_ACCESS, False,
                       'Global\901D5C9B-A3A2-4DE9-AF3B-9CEFF5290F5B');
   try
   if (GMutex = 0) then
   begin
      LMutex := CreateMutex(Nil, False,
                            '{38D1BAC7-A11E-4CDE-9306-8A0E9F2FBB30}');
      try
         if GetLastError = ERROR_ALREADY_EXISTS then
         begin
            SendMessage(HWND_BROADCAST, RegisterWindowMessage(
                       '{38D1BAC7-A11E-4CDE-9306-8A0E9F2FBB30}'),0,0);
         end
         else
         begin
            Application.Initialize;
            Application.MainFormOnTaskbar := True;
            Application.Title := 'Expression Parser';
            Application.CreateForm(TRegular_Expression_Form,
                                   Regular_Expression_Form);
            Application.CreateForm(TXMLForm, XMLForm);
            Application.Run;
         end;
      finally
         releasemutex(LMutex);
         closehandle(LMutex);
      end;
   end
   else
      MessageDlg('Scanning Service Is Running'+slinebreak+
                 'Stop the Service before loading Expressions',
mtError, [mbOK],
```

```
                0);
   finally
      releasemutex(GMutex);
      closehandle(GMutex);
   end;
end.
```

RegExMain.pas

```
unit RegExMain;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ZAbstractRODataset, ZAbstractDataset,
ZDataset, ZConnection, DBRV, ComCtrls, StdCtrls, DBCtrls,Mask,
ExtCtrls, RVScroll, RichView, RVStyle, PerlRegEx, RVGetTextW, Buttons,
ZAbstractTable, DCPmd5, DCPcrypt2;

type
  TRegular_Expression_Form = class(TForm)
    Label1:                          TLabel;
    Label2:                          TLabel;
    Label3:                          TLabel;
    Label4:                          TLabel;
    Label5:                          TLabel;
    Label6:                          TLabel;
    BitBtn1:                         TBitBtn;
    BitBtn2:                         TBitBtn;
    BitBtn3:                         TBitBtn;
    BitBtn4:                         TBitBtn;
    TreeView1:                       TTreeView;
    StatusBar1:                      TStatusBar;
    DCP_md51:                        TDCP_md5;
    SpeedButton1:                    TSpeedButton;

    RVStyle1:                        TRVStyle;
    DBRichView1:                     TDBRichView;
    DBNavigator1:                    TDBNavigator;
    DBEdit1:                         TDBEdit;
    DBEdit2:                         TDBEdit;
    DBEdit3:                         TDBEdit;
    DBMemo1:                         TDBMemo;
    DBMemo2:                         TDBMemo;
    DBCheckBox1:                     TDBCheckBox;
    DBCheckBox2:                     TDBCheckBox;
    DBCheckBox3:                     TDBCheckBox;
    DBCheckBox4:                     TDBCheckBox;
    DBCheckBox5:                     TDBCheckBox;
    DBCheckBox6:                     TDBCheckBox;
    DBCheckBox7:                     TDBCheckBox;
```

```
PerlRegEx1:                          TPerlRegEx;

ZConnection1:                        TZConnection;

RegExpDS:                            TDataSource;
RegExpQuery:                         TZQuery;
RegExpQueryidregex_pattern:          TIntegerField;
RegExpQueryinclude_in_export:        TSmallintField;
RegExpQuerypattern:                  TMemoField;
RegExpQuerysubexpression:            TIntegerField;
RegExpQuerypattern_name:             TWideStringField;
RegExpQueryexport_name:              TWideStringField;
RegExpQuerypattern_description:      TMemoField;
RegExpQuerycaseless:                 TSmallintField;
RegExpQuerysingle_line:              TSmallintField;
RegExpQueryanchored:                 TSmallintField;
RegExpQuerymulti_line:               TSmallintField;
RegExpQueryextended:                 TSmallintField;
RegExpQueryungreedy:                 TSmallintField;

YrDS:                                TDataSource;
YrQuery:                             TZQuery;
YrQueryidyear:                       TIntegerField;
YrQueryyear:                         TWideStringField;

CaseDS:                              TDataSource;
CaseQuery:                           TZQuery;
CaseQueryidcase:                     TIntegerField;
CaseQueryidyear:                     TIntegerField;
CaseQuerycase_number:                TWideStringField;
CaseQuerycase_start_date:            TDateField;
CaseQuerycase_close_date:            TDateField;

DocDS:                               TDataSource;
DocQuery:                            TZQuery;
DocQueryidcase_document:             TIntegerField;
DocQueryidyear:                      TIntegerField;
DocQueryidcase:                      TIntegerField;
DocQueryiddocument_type:             TIntegerField;
DocQuerydocument_name:               TWideStringField;
DocQuerydocument_text:               TBlobField;
DocQuerydatetime_last_modified:      TDateTimeField;
DocQuerydatetime_last_scan:          TDateTimeField;
DocQuerydatetime_created:            TDateTimeField;
DocQuerydatetime_last_accessed:      TDateTimeField;

SearchedDocDS:                       TDataSource;
SearchedDocTable:                    TZTable;
SearchedDocTableidregex_pattern:     TIntegerField;
SearchedDocTableidyear:              TIntegerField;
SearchedDocTableidcase:              TIntegerField;
```

```
    SearchedDocTableidcase_document:  TIntegerField;

    MatchedTextDS:                    TDataSource;
    MatchedTextTable:                 TZTable;
    MatchedTextTableidmatched_text:   TIntegerField;
    MatchedTextTableidregex_pattern:  TIntegerField;
    MatchedTextTableidyear:           TIntegerField;
    MatchedTextTableidcase:           TIntegerField;
    MatchedTextTableidcase_document:  TIntegerField;
    MatchedTextTablematched_text:     TMemoField;

    PostProcessQuery:                 TZQuery;
    PostProcessQuerysql_command:      TMemoField;
    PostProcessQueryidpost_process:   TIntegerField;
    PostProcessQueryidregex_pattern:  TIntegerField;

    DeleteQuery:                      TZQuery;
    UniqueQuery:                      TZQuery;
    PostProcessExecuteQuery:          TZQuery;
    RegExpQueryinclude_in_scan:       TSmallintField;
    DBCheckBox8:                      TDBCheckBox;
    UpdateDocCheckBox:                TCheckBox;

    procedure FormCreate             (Sender: TObject);
    procedure FormDestroy            (Sender: TObject);
    procedure FormClose              (Sender: TObject;
                                      var Action: TCloseAction);
    procedure PerlRegEx1Match        (Sender: TObject);
    procedure BitBtn1Click           (Sender: TObject);
    procedure BitBtn2Click           (Sender: TObject);
    procedure BitBtn3Click           (Sender: TObject);
    procedure BitBtn4Click           (Sender: TObject);
    procedure SpeedButton1Click      (Sender: TObject);
    procedure FormCloseQuery         (Sender: TObject;
                                      var CanClose: Boolean);
    procedure FormKeyDown            (Sender: TObject; var Key: Word;
                                      Shift: TShiftState);
  private
    Root:                             TTreeNode;
    RegMatch:                         TTreeNode;
    YrMatch:                          TTreeNode;
    CaseMatch:                        TTreeNode;
    OutStrMatch:                      TTreeNode;
    DocMatch:                         TTreeNode;
    NewYear:                          Boolean;
    NewCase:                          Boolean;
    NewDoc:                           Boolean;
    NewRegEx:                         Boolean;
    Stop:                             Boolean;
    CloseButtonToggledOff:            Boolean;
    procedure UpdateSearchedDocument (REID, YrID, CsID, DcID: Integer);
    procedure UpdateMatchedTextTable (REID, YrID, CsID, DcID: Integer;
```

```
                                        OutString: String);
    procedure FilterRecords;
    procedure CondenseToUnique;
    procedure SetPerlRegExOptions;
    function GetMD5Value              (S:ANSIString): ANSIString;
    procedure Toggle_Close_Button     (Value: Boolean);
    { Private declarations }
  public
    function GetRegexID:              Integer;
    function GetRegexName:            String;
    { Public declarations }
  end;

var
  Regular_Expression_Form:            TRegular_Expression_Form;
  OldWindowProc:                      Pointer;
  MyMsg:                              LongInt;

implementation

{$R *.dfm}

uses XML, Filter;

function NewWindowProc(WindowHandle: HWND; TheMessage: LongInt; ParamW:
LongInt;
                       ParamL: LongInt): LongInt; stdcall;
begin
   if TheMessage = MyMsg then
   begin
      SendMessage(Application.Handle, WM_SYSCOMMAND, SC_RESTORE, 0);
      SetForeGroundWindow(Application.Handle);
      Result := 0;
      Exit;
   end;

Result:=CallWindowProc(OldWindowProc,WindowHandle,TheMessage,ParamW,Par
amL);
end;

procedure TRegular_Expression_Form.BitBtn1Click(Sender: TObject);
var
   SearchStream:  UTF8String;
   bk:            TBookMark;
begin
   if Not(RegexpQuery.State in [dsEdit,dsInsert]) then
   begin
      Toggle_Close_Button(True);
      Treeview1.SetFocus;
      BitBtn1.Enabled := False;
      BitBtn1.Visible := False;
      BitBtn2.Enabled := False;
```

```
BitBtn3.Visible := True;
BitBtn3.Enabled := True;
DeleteQuery.ExecSQL;
SearchedDocTable.Open;
MatchedTextTable.Open;
NewYear := True;
NewCase := True;
NewDoc := True;
NewRegEx := True;
StatusBar1.Panels[0].Text := 'Starting Match!';
Treeview1.Items.BeginUpdate;
Treeview1.Items.Clear;
Root := Treeview1.Items.Add(nil, 'Matches Found');
Treeview1.Items.EndUpdate;
bk := RegExpQuery.GetBookmark;
try
   RegExpQuery.DisableControls;
   try
      yrQuery.Open;
      while (Not (yrQuery.EOF)) and (not (stop)) do
      begin
         caseQuery.SQL.Clear;
         caseQuery.SQL.BeginUpdate;
         caseQuery.SQL.Add('select * from case_file ');
         caseQuery.SQL.Add('where idyear = '+
                           yrQueryidyear.AsString);
         caseQuery.SQL.Add('order by case_number desc');
         caseQuery.SQL.EndUpdate;
         caseQuery.Open;
         while (Not (caseQuery.EOF)) and (not (stop)) do
         begin
            docQuery.SQL.Clear;
            docQuery.SQL.BeginUpdate;
            docQuery.SQL.Add('select * from case_document');
            docQuery.SQL.Add('where idyear ='+
                             caseQueryidyear.AsString +
                             ' and idcase = '+
                             caseQueryidcase.AsString);
            docQuery.SQL.Add('order by document_name');
            docQuery.SQL.EndUpdate;
            docQuery.Open;
            StatusBar1.Panels[0].Text := 'Case:  '+
                                 caseQuerycase_number.AsString +
                                 ' Document:  ' +
                                 docQuerydocument_name.AsString;
            while (Not (docQuery.EOF)) and (not (stop)) do
            begin
               RegexpQuery.First;
               while Not RegexpQuery.EOF do
               begin
                  if
                  RegExpQueryInclude_In_Scan.AsInteger = 1 then
```

```
                    begin
                       StatusBar1.Panels[1].Text :=
                          'Processing Regular Expression Type:  ' +
                           RegExpQueryPattern_Name.AsString;
                       PerlRegEx1.RegEx := UTF8Encode(
                                     RegExpQueryPattern.AsString);
                       SearchStream :=
                              UTF8Encode(GetAllText(DBRichview1));
                       PerlRegEx1.Subject := SearchStream;
                       UpdateSearchedDocument(
                       RegExpQueryidRegEx_Pattern.AsInteger,
                       YrQueryidYear.AsInteger,
                       CaseQueryidCase.AsInteger,
                       DocQueryidcase_document.AsInteger);
                       SetPerlRegExOptions;
                       PerlRegEx1.Match;
                    end;
                    RegExpQuery.Next;
                    NewRegEx := True;
                 end;
                 if UpdateDocCheckBox.Checked then
                 begin
                    docQuery.Edit;
                    docQuerydatetime_last_scan.AsDateTime := Now;
                    docQuery.Post;
                 end;
                 docQuery.Next;
                 newdoc := True;
              end;
              docQuery.Close;
              caseQuery.Next;
              newcase := True;
           end;
           caseQuery.Close;
           yrQuery.Next;
           newyear := true;
        end;
        yrQuery.Close;
     finally
        RegExpQuery.EnableControls;
     end;
     RegExpQuery.GotoBookmark(bk);
finally
   RegExpQuery.GotoBookmark(bk);
end;
if Not Stop then
begin
   FilterRecords;
   CondenseToUnique;
end;
BitBtn2.Enabled := True;
BitBtn1.Visible := True;
```

```
      BitBtn1.Enabled := True;
      BitBtn3.Enabled := False;
      BitBtn3.Visible := False;
      SearchedDocTable.Close;
      Stop := False;
      DeleteQuery.ExecSQL;
      StatusBar1.Panels[0].Text := 'Search Complete!';
      StatusBar1.Panels[1].Text := '';
      Toggle_Close_Button(False);
   end
   else
      MessageDlg('Finish Edits Before Parsing!',mtError, [mbOK],0);
end;

procedure TRegular_Expression_Form.BitBtn2Click(Sender: TObject);
begin
   Close;
end;

procedure TRegular_Expression_Form.BitBtn3Click(Sender: TObject);
begin
   Stop := True;
end;

procedure TRegular_Expression_Form.BitBtn4Click(Sender: TObject);
var
   ChildForm: TXMLForm;
begin
   ChildForm := TXMLForm.Create(Application);
   try
      try
         ChildForm.ShowModal;
      finally
         ChildForm.Free;
      end;
   finally
      if screen.Cursor <> crDefault then
         screen.Cursor := crDefault;
   end;
end;

procedure TRegular_Expression_Form.CondenseToUnique;
begin
   StatusBar1.Panels[0].Text := 'Condencing Records';
   StatusBar1.Panels[1].Text := '';
   MatchedTextTable.Refresh;
   MatchedTextTable.First;
   while (Not (MatchedTextTable.EOF)) and (not (stop)) do
   begin
      Application.ProcessMessages;
      UniqueQuery.SQL.Clear;
      UniqueQuery.SQL.Text :=
```

```
         'insert into unique_matched_text '+
        '(idregex_pattern,idcase,idyear,md5_value,unique_matched_text) ' +
         'values ('+MatchedTextTableidregex_pattern.AsString +','+
                   MatchedTextTableidcase.AsString +','+
                   MatchedTextTableidYear.AsString +","'+
                   String(GetMD5Value(ANSIString(UpperCase(
                   MatchedTextTableMatched_Text.AsString))))+'",'+
                   QuotedStr(
                   MatchedTextTableMatched_Text.AsString)+')';
      try
         UniqueQuery.ExecSQL;
      except
        //Expecting key violation for duplicate matches within the same case
        //thus the key violation error reported by the database is silenced.
      end;
      MatchedTextTable.Next;
   end;
   MatchedTextTable.Close;
end;

procedure TRegular_Expression_Form.FilterRecords;
begin
   PostProcessQuery.Open;
   while Not PostProcessQuery.EOF do
   begin
      PostProcessExecuteQuery.SQL.Clear;
      PostProcessExecuteQuery.SQL.Text :=
                                  PostProcessQuerySQL_Command.AsString;
      PostProcessExecuteQuery.ExecSQL;
      PostProcessQuery.Next;
   end;
   PostProcessQuery.Close;
end;

procedure TRegular_Expression_Form.FormClose(Sender: TObject;
                                             var Action: TCloseAction);
begin
   RegExpQuery.Close;
end;

procedure TRegular_Expression_Form.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
begin
   if RegExpQuery.State in [dsEdit, dsInsert] then
   begin
      case MessageDlg('Save Changes?', mtConfirmation,
                      [mbYes,mbNo,mbCancel],0) of
        mrYes: begin
                  RegExpQuery.Post;
                  CanClose := True;
               end;
        mrNo, mrCancel:  CanClose := False;
```

```
      end;
   end;
end;

procedure TRegular_Expression_Form.FormCreate(Sender: TObject);

 procedure InitializeValues;
 begin
    Top := 0;
    Left := 0;
    CloseButtonToggledOff := False;
    NewCase := True;
    NewDoc := True;
    Stop := False;
    Application.HintHidePause := MaxInt;
 end;

 procedure setRegExOptionHints;
 begin
   DBCheckBox1.Hint :=  //caseless
    'Tries to match the regex without paying attention to case'+slinebreak+
    'If set, ''Bye'' will match ''Bye'', ''bye'', ''BYE'' and'+slinebreak+
    'even ''byE'', ''bYe'', etc. Otherwise, only ''Bye'' will match.'+slinebreak+
    'Equivalent to Perl''s /i modifier.';
   DBCheckBox2.Hint :=   //multiline
    'The ^ (beginning of string) and $ (ending of string) regex'+slinebreak+
    'operators will also match right after and right before a newline'+slinebreak+
    'in the Subject string. This effectively treats 1 string with'+slinebreak+
    'multiple lines as multiple strings.'+slinebreak+
    'Equivalent to Perl''s /m modifier.'+slinebreak+
    'Note that Multi-Line and Single-Line can be used together.';
   DBCheckBox3.Hint :=   //Single-line
    'Normally, dot (.) matches anything but a newline (\n).'+slinebreak+
    'With Single-Line, dot(.) will match anything, including newlines.'+slinebreak+
    'This allows a multiline string to be regarded as a single entity.'+slinebreak+
    'Equivalent to Perl''s /s modifier.'+slinebreak+
    'Note that Multi-Line and Single-Line can be used together.';
   DBCheckBox4.Hint := //Extended
    'Allow regex to contain extra whitespace, newlines and Perl-style'+slinebreak+
    'comments, all of which will be filtered out.'+slinebreak+
    'This is sometimes called "free-spacing mode."';
   DBCheckBox5.Hint := //ungreedy
     'Repeat operators (?,*,+,{num,num}) are greedy by default, i.e.'+slinebreak+
     'they try to match as many characters as possible.'+slinebreak+
     'Set Ungreedy to use ungreedy repeat operators by default, i.e.'+ slinebreak+
     'they try to match as few characters as possible.';
   DBCheckBox6.Hint :=  //Anchored
    'Allows the regex to match only at the start of the subject or'+slinebreak+
    'right after the previous match.';
 end;

begin
   InitializeValues;
   Width := Screen.Width;
   StatusBar1.Panels[0].Text := 'Ready';
   SetRegExOptionHints;
   MyMsg := RegisterWindowMessage(
```

```
                              '{38D1BAC7-A11E-4CDE-9306-8A0E9F2FBB30}');
   OldWindowProc := Pointer(SetWindowLong(Self.Handle, GWL_WNDPROC,
                             LongInt(@NewWindowProc)));
   RegExpQuery.Open;
end;

procedure TRegular_Expression_Form.FormDestroy(Sender: TObject);
begin
   SetWindowLong(Self.Handle, GWL_WNDPROC, LongInt(OldWindowProc));
end;

procedure TRegular_Expression_Form.FormKeyDown(Sender: TObject;
                                               var Key: Word;
                                               Shift: TShiftState);
begin
   if ((CloseButtonToggledOff) and
       (Key = VK_F4) and (ssAlt in Shift)) then
      Key := 0;
end;

function TRegular_Expression_Form.GetMD5Value(S:ANSIString):
ANSIString;
Var
   StACrypt, StCrypt: ANSIString;
   Digest: array [0..15] of Byte;
   I: Integer;
begin
   StACrypt := S;
   DCP_md51.Init;
   DCP_md51.UpdateStr(StACrypt);
   DCP_md51.Final(Digest);
   StCrypt := '';
   for I := 0 to 15 do
      StCrypt:=StCrypt+ANSIString(IntToHex(Digest[I], 2));
   Result := StCrypt;
end;

function TRegular_Expression_Form.GetRegexID: Integer;
begin
   Result := RegexpQueryidregex_Pattern.AsInteger;
end;

function TRegular_Expression_Form.GetRegexName: String;
begin
   Result := RegexpQueryPattern_name.AsString;
end;

procedure TRegular_Expression_Form.PerlRegEx1Match(Sender: TObject);
var
   OutString:  UTF8String;
begin
   if NewYear then
```

```
begin
   Treeview1.Items.BeginUpdate;
   YrMatch := Treeview1.Items.AddChild(Root, YrQueryYear.AsString);
   YrMatch.MakeVisible;
   NewYear := False;
   Treeview1.Items.EndUpdate;
end;
if NewCase then
begin
   Treeview1.Items.BeginUpdate;
   CaseMatch := Treeview1.Items.AddChild(YrMatch,
                                  CaseQueryCase_Number.AsString);
   CaseMatch.MakeVisible;
   NewCase := False;
   Treeview1.Items.EndUpdate;
end;
if NewDoc then
begin
   Treeview1.Items.BeginUpdate;
   DocMatch := Treeview1.Items.AddChild(CaseMatch,
               DocQuerydocument_name.AsString);
   DocMatch.MakeVisible;
   NewDoc := False;
   Treeview1.Items.EndUpdate;
end;
if NewRegex then
begin
   Treeview1.Items.BeginUpdate;
   RegMatch := Treeview1.Items.AddChild(DocMatch,
                                   RegExpQueryPattern_Name.AsString);
   RegMatch.MakeVisible;
   NewRegex := False;
   Treeview1.Items.EndUpdate;
end;
if RegExpQuerysubexpression.AsString <> '' then
   OutString := PerlRegEx1.SubExpressions[
                                  RegExpQuerysubexpression.AsInteger]
else
   OutString := PerlRegEx1.MatchedExpression;
UpdateMatchedTextTable(RegExpQueryidRegEx_Pattern.AsInteger,
                       YrQueryidYear.AsInteger,
                       CaseQueryidCase.AsInteger,
                       DocQueryidcase_document.AsInteger,
                       UTF8toString(OutString));
Treeview1.Items.BeginUpdate;
OutStrMatch := Treeview1.Items.AddChild(RegMatch,
                                        UTF8toString(OutString));
OutStrMatch.MakeVisible;
Treeview1.Items.EndUpdate;
Application.ProcessMessages;
if (Not (Stop)) then
   PerlRegEx1.MatchAgain;
```

```
end;

procedure TRegular_Expression_Form.SetPerlRegExOptions;
begin
   PerlRegEx1.options := [];
   if RegExpQuerycaseless.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preCaseLess];
   if RegExpQuerysingle_line.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preSingleLine];
   if RegExpQuerymulti_line.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preMultiLine];
   if RegExpQueryanchored.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preAnchored];
   if RegExpQueryextended.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preExtended];
   if RegExpQueryungreedy.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preUngreedy];
end;

procedure TRegular_Expression_Form.SpeedButton1Click(Sender: TObject);
var
   ChildForm: TFilterForm;
begin
   ChildForm := TFilterForm.Create(Application);
   try
      ChildForm.ShowModal;
   finally
      ChildForm.Free;
   end;
end;

procedure TRegular_Expression_Form.Toggle_Close_Button(Value: Boolean);
var
   hSysMenu:  THandle;
begin
   hSysMenu := GetSystemMenu(Handle, false);
   if Value then
   begin
      EnableMenuItem(hSysMenu, SC_CLOSE, MF_BYCOMMAND or MF_GRAYED);
      CloseButtonToggledOff := True;
   end
   else
   begin
      EnableMenuItem(hSysMenu, SC_CLOSE, MF_BYCOMMAND or MF_ENABLED);
      CloseButtonToggledOff := False;
   end;
end;

procedure TRegular_Expression_Form.UpdateMatchedTextTable(REID, YrID,
                                                          CsID, DcID:
                                                          Integer;
                                                          OutString:
```

```
                                                          String);
begin
   MatchedTextTable.Insert;
   MatchedTextTableidregex_pattern.Value := REID;
   MatchedTextTableidyear.Value          := YrID;
   MatchedTextTableidcase.Value          := CsID;
   MatchedTextTableidcase_document.Value := DcID;
   MatchedTextTableMatched_Text.AsString := OutString;
   MatchedTextTable.Post;
end;

procedure TRegular_Expression_Form.UpdateSearchedDocument(REID, YrID,
                                                          CsID, DcID:
                                                          Integer);
begin
   SearchedDocTable.Insert;
   SearchedDocTableidregex_pattern.Value := REID;
   SearchedDocTableidyear.Value          := YrID;
   SearchedDocTableidcase.Value          := CsID;
   SearchedDocTableidcase_document.Value := DcID;
   SearchedDocTable.Post;
end;

begin
   ShowWindow(Application.Handle, SW_HIDE);
end.
```

RegExMain.dfm

```
object Regular_Expression_Form: TRegular_Expression_Form
  Left = 0
  Top = 0
  ActiveControl = DBEdit1
  Caption = 'Regular Expression Parser'
  ClientHeight = 612
  ClientWidth = 839
  Color = clBtnFace
  Constraints.MinHeight = 500
  Constraints.MinWidth = 855
  DoubleBuffered = True
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  Position = poDesigned
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  OnDestroy = FormDestroy
```

```
  OnKeyDown = FormKeyDown
  DesignSize = (
    839
    612)
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 109
    Top = 93
    Width = 131
    Height = 13
    Caption = 'Regular Expression Pattern'
  end
  object Label2: TLabel
    Left = 280
    Top = 13
    Width = 70
    Height = 13
    Caption = 'Pattern Name:'
    FocusControl = DBEdit1
  end
  object Label3: TLabel
    Left = 32
    Top = 37
    Width = 105
    Height = 13
    Caption = 'Description of Pattern'
    FocusControl = DBMemo1
  end
  object Label4: TLabel
    Left = 32
    Top = 225
    Width = 79
    Height = 13
    Caption = 'Pattern Matches'
  end
  object Label5: TLabel
    Left = 396
    Top = 217
    Width = 62
    Height = 13
    Anchors = [akTop, akRight]
    Caption = 'Export Name'
    FocusControl = DBEdit2
  end
  object Label6: TLabel
    Left = 606
    Top = 217
    Width = 74
    Height = 13
    Anchors = [akTop, akRight]
    Caption = 'Sub-Expression'
```

```
    FocusControl = DBEdit3
    ExplicitLeft = 1089
  end
  object SpeedButton1: TSpeedButton
    Left = 797
    Top = 8
    Width = 23
    Height = 22
    Hint = 'Assign Pattern Filters'
    Anchors = [akTop, akRight]
    Glyph.Data = {
      36030000424D360300000000000036000000280000001000000010000000100
      18000000000000030000120B0000120B00000000000000000000FF00FFFF00FF
      FF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00
      FFFF00FF485058FF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF
      00FFFF00FFFF00FFFF00FFFF00FFFF00FF485058215A96485058FF00FFFF00FF
      C7BBB6C7BBB6C7BBB6C7BBB6C7BBB6C7BBB6C7BBB6C7BBB6C7BBB64850
      581B5B9F40ABFF1C8FFFFF00FFFF00FFEEBB9EFFFFFFFFFFFFFFFFFFFFFFFD3
      B5A7BD9186BD9186BD9186BD91861B5B9F40ABFF1C8FFFFF00FFEEBB9E9A9A9A
      EEBB9E3D463D28382E284134284536BD91868BBA888BBA888BBA888BBA88BD91
      861C8FFFFF00FFFF00FFEEBB9EFFFFFFEEBB9E18513A07462E085235BD91868B
      BA888BBA888BBA888BBA888BBA888BBA88BD9186FF00FFFF00FFEEBB9E3D463D
      F0C4A722916B0C82590D7A56BD91868BBA888BBA888BBA88FEE7A9F7E1A3E5D1
      97BD9186FF00FFFF00FFF0C4A718513AF0C4A7AAAC8395986E928A67BD9186EB
      D79CFBE6A6FAE9B8FBE5A7F0DDA4D9C791BD9186FF00FFFF00FFF0C4A722916B
      F0C4A7F4A97DF6A67EF3834EBD9186CBAC87E7D499F2DDA0F3E2B1E9D8A7D0B8
      93BD9186FF00FFFF00FFF0C4A7AAAC83F0C4A7EE8F5CEE8453EB7843EB6732BD
      9186CFAC8BD5BB90D3BA90CCB299BD9186C7BBB6FF00FFFF00FFF0C4A7F4A97D
      F0C4A7ED703AEA5E27E95C25E75622E96435BD9186BD9186BD9186BD9186E387
      62C7BBB6FF00FFFF00FFF0C4A7EE8F5CF3CCBAEFBAA1EEA68BEB7F59EB8663EE
      A086ED9A7EE5734FE37350EEA68FEBC6B5C7BBB6FF00FFFF00FFF0C4A7ED703A
      F3CCBAFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
      FFC7BBB6FF00FFFF00FFF3CCBAEFBAA1F3CCBAF3CCBAF3CCBAF0C4A7F0C4A7F0
      C4A7F0C4A7F0C4A7F0C4A7EEBB9EEEBB9EEEBB9EFF00FFFF00FFF3CCBAFFFFFF
      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9A9A9AFF00
      FFFF00FFFF00FFFF00FFF3CCBAF3CCBAF3CCBAF3CCBAF0C4A7F0C4A7F0C4A7F0
      C4A7F0C4A7EEBB9EEEBB9EEEBB9EFF00FFFF00FFFF00FFFF00FF}
    ParentShowHint = False
    ShowHint = True
    OnClick = SpeedButton1Click
    ExplicitLeft = 1272
  end
  object DBRichView1: TDBRichView
    Left = 8
    Top = 593
    Width = 321
    Height = 40
    DataField = 'document_text'
    DataSource = DocDS
    TabOrder = 19
    Visible = False
    DoInPaletteMode = rvpaCreateCopies
```

```
    RTFReadProperties.TextStyleMode = rvrsAddIfNeeded
    RTFReadProperties.ParaStyleMode = rvrsAddIfNeeded
    RVFOptions = [rvfoSavePicturesBody, rvfoSaveControlsBody,
rvfoSaveBinary, rvfoSaveTextStyles, rvfoSaveParaStyles,
rvfoSaveDocProperties, rvfoLoadDocProperties]
    Style = RVStyle1
  end
  object BitBtn3: TBitBtn
    Left = 595
    Top = 562
    Width = 105
    Height = 25
    Anchors = [akRight, akBottom]
    Caption = 'Stop'
    DoubleBuffered = True
    Enabled = False
    Glyph.Data = {
      26050000424D260500000000000036040000280000000E0000000F0000000100
      080000000000F000000000000000000000000001000000000000000000000000
      BF0000BF000000BFBF00BF000000BF00BF00BFBF0000C0C0C000C0DCC000F0CA
      A600000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      000000000000000000000000000000000000F0FBFF00A4A0A000808080000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00070707070707
      0707070707070707000007070707070000000007070707070000070707000000
```

```
      000000000007070700000707000000F9F9F9F900000007070000070700000F9F9
      F9F9F9F9000007077C00070000F9F9F9F9F9F9F9F900000070110070000F9F9F9
      F9F9F9F9F9000007E28D070000F9F9F9F9F9F9F9F9000007F400070000FFF9F9
      F9F9F9F9F900000704000700000FFFFF9F9F9F9F9F900000070300070700000FFFF
      F9F9F9F900000707910207070000000FFFFF9F900000000707 4E08070707000000
      0000000000070707010007070707070000000007070707070FFFF070707070707
      07070707070707070000}
    ParentDoubleBuffered = False
    TabOrder = 15
    Visible = False
    OnClick = BitBtn3Click
  end
  object DBNavigator1: TDBNavigator
    Left = 16
    Top = 8
    Width = 240
    Height = 25
    DataSource = RegExpDS
    Hints.Strings = (
      'First record'
      'Prior record'
      'Next record'
      'Last record'
      'Insert record'
      'Delete record'
      'Edit record'
      'Save edit'
      'Cancel edit'
      'Refresh data')
    ParentShowHint = False
    ShowHint = True
    TabOrder = 17
  end
  object DBEdit1: TDBEdit
    Left = 356
    Top = 10
    Width = 426
    Height = 21
    Anchors = [akLeft, akTop, akRight]
    DataField = 'pattern_name'
    DataSource = RegExpDS
    TabOrder = 0
  end
  object DBMemo1: TDBMemo
    Left = 32
    Top = 52
    Width = 788
    Height = 38
    Anchors = [akLeft, akTop, akRight]
    DataField = 'pattern_description'
    DataSource = RegExpDS
    ScrollBars = ssVertical
```

```
  TabOrder = 1
end
object DBMemo2: TDBMemo
  Left = 109
  Top = 108
  Width = 711
  Height = 97
  Anchors = [akLeft, akTop, akRight]
  DataField = 'pattern'
  DataSource = RegExpDS
  ScrollBars = ssVertical
  TabOrder = 2
end
object TreeView1: TTreeView
  Left = 32
  Top = 240
  Width = 788
  Height = 306
  Anchors = [akLeft, akTop, akRight, akBottom]
  DoubleBuffered = True
  Indent = 19
  ParentDoubleBuffered = False
  TabOrder = 18
  TabStop = False
end
object StatusBar1: TStatusBar
  Left = 0
  Top = 593
  Width = 839
  Height = 19
  Panels = <
    item
      Width = 600
    end
    item
      Width = 50
    end>
end
object DBEdit2: TDBEdit
  Left = 464
  Top = 213
  Width = 134
  Height = 21
  Anchors = [akTop, akRight]
  DataField = 'export_name'
  DataSource = RegExpDS
  TabOrder = 11
end
object DBEdit3: TDBEdit
  Left = 686
  Top = 213
  Width = 134
```

```
  Height = 21
  Anchors = [akTop, akRight]
  DataField = 'subexpression'
  DataSource = RegExpDS
  TabOrder = 12
end
object BitBtn1: TBitBtn
  Left = 595
  Top = 562
  Width = 105
  Height = 25
  Anchors = [akRight, akBottom]
  Caption = 'Start Search'
  Default = True
  DoubleBuffered = True
  Glyph.Data = {
    36030000424D360300000000000036000000280000001000000010000000100
    18000000000000030000120B0000120B00000000000000000000FF00FF314B62
    AC7D7EFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00
    FFFF00FFFF00FFFF00FF5084B20F6FE1325F8CB87E7AFF00FFFF00FFFF00FFFF
    00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FF32A0FE37A1FF
    106FE2325F8BB67D79FF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00
    FFFF00FFFF00FFFF00FFFF00FF37A4FE379FFF0E6DDE355F89BB7F79FF00FFFF
    00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FF
    37A4FE359EFF0F6FDE35608BA67B7FFF00FFFF00FFFF00FFFF00FFFF00FFFF00
    FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FF38A5FE329DFF156DCE444F5BFF
    00FF9C6B65AF887BAF887EAA8075FF00FFFF00FFFF00FFFF00FFFF00FFFF00FF
    FF00FFFF00FF3BABFFA1CAE7AD8679A98373E0CFB1FFFFDAFFFFDDFCF8CFCCB2
    9FA1746BFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFC0917DFC
    E9ACFFFFCCFFFFCFFFFFD0FFFFDEFFFFFAE3D3D1996965FF00FFFF00FFFF00FF
    FF00FFFF00FFFF00FFFF00FFB08978FAD192FEF4C2FFFFD0FFFFDAFFFFF6FFFF
    FCFFFFFCB69384FF00FFFF00FFFF00FFFF00FFFF00FFFF00FFB08978FEDA97ED
    B478FBEEBBFFFFD3FFFFDCFFFFF4FFFFF4FFFFE2E9DDBCA67B73FF00FFFF00FF
    FF00FFFF00FFFF00FFB18A78FFDE99E9A167F4D199FEFCCCFFFFD5FFFFDAFFFF
    DCFFFFD7EFE6C5A97E75FF00FFFF00FFFF00FFFF00FFFF00FFAA7F73FAE0A4F0
    B778EEBA7BF6DDA6FEFBCCFFFFD3FFFFD1FFFFD7D9C5A7A3756CFF00FFFF00FF
    FF00FFFF00FFFF00FFFF00FFCEB293FFFEDDF4D1A5EEBA7BF2C78FF8E1ABFCF0
    BAFCFACAA3776FFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFA1746BE1
    D4D3FFFEEEF7CC8CF0B473F7C788FCE3A5C2A088A5776CFF00FFFF00FFFF00FF
    FF00FFFF00FFFF00FFFF00FFFF00FF986865BA9587EAD7A4EAD59EE0C097A577
    6CA5776CFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF00FFFF
    00FFFF00FFA77E70A98073A4786EFF00FFFF00FFFF00FFFF00FF}
  ParentDoubleBuffered = False
  TabOrder = 13
  OnClick = BitBtn1Click
end
object BitBtn2: TBitBtn
  Left = 715
  Top = 562
  Width = 105
  Height = 25
  Anchors = [akRight, akBottom]
```

```
    Cancel = True
    Caption = 'Close'
    DoubleBuffered = True
    Glyph.Data = {
      4E010000424D4E010000000000007600000028000000120000001200000001000
      040000000000D8000000000000000000000000100000010000000000000000000
      8000008000000080800080000000800080008080000080808000C0C0C0000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00888888888888
      8888880000008888800888888888880000008880FF00888888880000008880
      F00FF00888888800000880F0FF00FF00888880000080F0F78FF00FF0888800
      0000880F70078FF00888880000080F70FB0078FF08888000008800FBFBF007
      088888000000880FBFBFBFB00888880000080FBFBFBFBFBF088880000008800
      BFBFBFBF08888800000888800FBFBF0880888000000888888800B80880008800
      0000888888880088000080000008888888888888088000000888888888888
      880888000000888888888888888888000000}
    ParentDoubleBuffered = False
    TabOrder = 14
    OnClick = BitBtn2Click
  end
  object DBCheckBox1: TDBCheckBox
    Left = 32
    Top = 108
    Width = 71
    Height = 17
    Caption = 'Caseless'
    DataField = 'caseless'
    DataSource = RegExpDS
    ParentShowHint = False
    ShowHint = True
    TabOrder = 3
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object DBCheckBox2: TDBCheckBox
    Left = 32
    Top = 124
    Width = 71
    Height = 17
    Caption = 'Multi-Line'
    DataField = 'multi_line'
    DataSource = RegExpDS
    ParentShowHint = False
    ShowHint = True
    TabOrder = 4
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object DBCheckBox3: TDBCheckBox
    Left = 32
    Top = 140
    Width = 71
    Height = 17
```

```
    Caption = 'Single-Line'
    DataField = 'single_line'
    DataSource = RegExpDS
    ParentShowHint = False
    ShowHint = True
    TabOrder = 5
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object DBCheckBox4: TDBCheckBox
    Left = 32
    Top = 156
    Width = 71
    Height = 17
    Caption = 'Extended'
    DataField = 'extended'
    DataSource = RegExpDS
    ParentShowHint = False
    ShowHint = True
    TabOrder = 6
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object DBCheckBox5: TDBCheckBox
    Left = 32
    Top = 172
    Width = 71
    Height = 17
    Caption = 'Ungreedy'
    DataField = 'ungreedy'
    DataSource = RegExpDS
    ParentShowHint = False
    ShowHint = True
    TabOrder = 7
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object DBCheckBox6: TDBCheckBox
    Left = 32
    Top = 188
    Width = 71
    Height = 17
    Caption = 'Anchored'
    DataField = 'anchored'
    DataSource = RegExpDS
    ParentShowHint = False
    ShowHint = True
    TabOrder = 8
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object DBCheckBox7: TDBCheckBox
```

```
    Left = 172
    Top = 216
    Width = 105
    Height = 17
    Anchors = [akTop, akRight]
    Caption = 'Include In Export'
    DataField = 'include_in_export'
    DataSource = RegExpDS
    TabOrder = 9
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object BitBtn4: TBitBtn
    Left = 471
    Top = 562
    Width = 105
    Height = 25
    Anchors = [akRight, akBottom]
    Caption = 'Export To XML'
    DoubleBuffered = True
    Glyph.Data = {
      36030000424D360300000000000036000000280000001000000010000000100
      18000000000000030000000000000000000000000000000000C0C0C0C0C0C0
      C0C0C0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0
      A0B0B0B0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFBFBFBFBFBFBF7F7F7F7F
      7F7F7F7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7F7F7F7F7F7FBFBFBFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F
      7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFFFFFFFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFFFFFFFBFBFBFBFBF7F
      7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFFFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFBFBFBFBFBFBF7F7F7F7F7F7F7F
      7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFFFFFFFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFFFFFFFBFBFBFBFBF7F
      7F7F7F7F7F7F7F7F7F7F7F7FFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7F7F7FFFFFFFFFFFFFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F
      7F7F7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
      DFDFDFFFFFFFFFFFFFFFFFFF000000000000000007F7F7FFFFFFFFFFFFFFFF
      FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0CFCFCFDFDFDFDFDFDFDFDFDFDFDFDFDF
      DFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFC0C0C0C0C0C0C0C0C0}
    ParentDoubleBuffered = False
    TabOrder = 16
    OnClick = BitBtn4Click
  end
  object DBCheckBox8: TDBCheckBox
```

```
    Left = 290
    Top = 216
    Width = 97
    Height = 17
    Anchors = [akTop, akRight]
    Caption = 'Include  In Scan'
    DataField = 'include_in_scan'
    DataSource = RegExpDS
    TabOrder = 10
    ValueChecked = '1'
    ValueUnchecked = '0'
  end
  object UpdateDocCheckBox: TCheckBox
    Left = 32
    Top = 565
    Width = 185
    Height = 17
    Anchors = [akLeft, akBottom]
    Caption = 'Update Date Document Scanned?'
    Checked = True
    State = cbChecked
    TabOrder = 21
  end
  object ZConnection1: TZConnection
    Protocol = 'mysql-5'
    HostName = '127.0.0.1'
    Port = 3306
    Database = 'thesis'
    User = 'thesis'
    Password = 'pass123'
    TransactIsolationLevel = tiReadCommitted
    Connected = True
    Top = 328
  end
  object RegExpQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'select'
      '   idregex_pattern,'
      '   include_in_export,'
      '   include_in_scan,'
      '   pattern,'
      '   subexpression,'
      '   pattern_name,'
      '   export_name,'
      '   pattern_description,'
      '   caseless,'
      '   single_line,'
      '   anchored,'
      '   multi_line,'
      '   extended,'
      '   ungreedy'
```

```
    'from'
    '   regex_pattern'
    'order by'
    '   pattern_name')
  Params = <>
  Left = 296
  Top = 8
  object RegExpQueryidregex_pattern: TIntegerField
    FieldName = 'idregex_pattern'
    Required = True
  end
  object RegExpQueryinclude_in_export: TSmallintField
    FieldName = 'include_in_export'
    Required = True
  end
  object RegExpQuerypattern: TMemoField
    FieldName = 'pattern'
    Required = True
    BlobType = ftMemo
  end
  object RegExpQuerysubexpression: TIntegerField
    FieldName = 'subexpression'
  end
  object RegExpQuerypattern_name: TWideStringField
    FieldName = 'pattern_name'
    Required = True
    Size = 45
  end
  object RegExpQueryexport_name: TWideStringField
    FieldName = 'export_name'
    Size = 10
  end
  object RegExpQuerypattern_description: TMemoField
    FieldName = 'pattern_description'
    Required = True
    BlobType = ftMemo
  end
  object RegExpQuerycaseless: TSmallintField
    FieldName = 'caseless'
    Required = True
  end
  object RegExpQuerysingle_line: TSmallintField
    FieldName = 'single_line'
    Required = True
  end
  object RegExpQueryanchored: TSmallintField
    FieldName = 'anchored'
    Required = True
  end
  object RegExpQuerymulti_line: TSmallintField
    FieldName = 'multi_line'
    Required = True
```

```
    end
    object RegExpQueryextended: TSmallintField
      FieldName = 'extended'
      Required = True
    end
    object RegExpQueryungreedy: TSmallintField
      FieldName = 'ungreedy'
      ReadOnly = True
    end
    object RegExpQueryinclude_in_scan: TSmallintField
      FieldName = 'include_in_scan'
      Required = True
    end
  end
  object RegExpDS: TDataSource
    DataSet = RegExpQuery
    Left = 256
    Top = 8
  end
  object RVStyle1: TRVStyle
    TextStyles = <
      item
        StyleName = 'Normal text'
        FontName = 'Arial'
        Unicode = True
      end
      item
        StyleName = 'Heading'
        FontName = 'Arial'
        Style = [fsBold]
        Color = clBlue
        Unicode = True
      end
      item
        StyleName = 'Subheading'
        FontName = 'Arial'
        Style = [fsBold]
        Color = clNavy
        Unicode = True
      end
      item
        StyleName = 'Keywords'
        FontName = 'Arial'
        Style = [fsItalic]
        Color = clMaroon
        Unicode = True
      end
      item
        StyleName = 'Jump 1'
        FontName = 'Arial'
        Style = [fsUnderline]
        Color = clGreen
```

```
      Jump = True
      Unicode = True
    end
    item
      StyleName = 'Jump 2'
      FontName = 'Arial'
      Style = [fsUnderline]
      Color = clGreen
      Jump = True
      Unicode = True
    end>
  ParaStyles = <
    item
      StyleName = 'Paragraph Style'
      Tabs = <>
    end
    item
      StyleName = 'Centered'
      Alignment = rvaCenter
      Tabs = <>
    end>
  ListStyles = <>
  InvalidPicture.Data = {
    07544269746D617036100000424D361000000000000036000000280000002000
    0000200000000100200000000000001000000000000000000000000000000000
    0000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
```

```
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0
C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C000C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF000000FF000000FF00FFFF
FF00FFFFFF000000FF000000FF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF000000FF000000
FF000000FF000000FF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF000000
FF000000FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF000000FF000000
FF000000FF000000FF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF000000FF000000FF00FFFF
FF00FFFFFF000000FF000000FF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
```

```
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF0080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00C0C0C00000000000FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF008080800080808000808080008080800080808000808080008080
    800080808000808080008080800080808000808080008080800080808000FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    800080808000FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFF
    FF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF00FFFFFF008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000808080008080800080808000808080008080800080808000808080008080
    8000}
StyleTemplates = <>
Top = 472
```

```
  end
  object PerlRegEx1: TPerlRegEx
    Options = [preCaseLess]
    OnMatch = PerlRegEx1Match
    Top = 376
  end
  object YrDS: TDataSource
    DataSet = YrQuery
    Left = 32
    Top = 512
  end
  object YrQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'Select *'
      'from'
      '   year'
      'order by'
      '   year desc')
    Params = <>
    Left = 72
    Top = 512
    object YrQueryidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object YrQueryyear: TWideStringField
      FieldName = 'year'
      Required = True
      Size = 4
    end
  end
  object CaseDS: TDataSource
    DataSet = CaseQuery
    Left = 120
    Top = 464
  end
  object CaseQuery: TZQuery
    Connection = ZConnection1
    SQL.Strings = (
      'select *'
      'from '
      '   case_file'
      'where'
      '   idyear = :yr'
      'order by'
      '   case_number desc')
    Params = <
      item
        DataType = ftUnknown
        Name = 'yr'
        ParamType = ptUnknown
```

```
    end>
  Left = 160
  Top = 472
  ParamData = <
    item
      DataType = ftUnknown
      Name = 'yr'
      ParamType = ptUnknown
    end>
  object CaseQueryidcase: TIntegerField
    FieldName = 'idcase'
    Required = True
  end
  object CaseQueryidyear: TIntegerField
    FieldName = 'idyear'
    Required = True
  end
  object CaseQuerycase_number: TWideStringField
    FieldName = 'case_number'
    Required = True
    Size = 10
  end
  object CaseQuerycase_start_date: TDateField
    FieldName = 'case_start_date'
    Required = True
  end
  object CaseQuerycase_close_date: TDateField
    FieldName = 'case_close_date'
  end
end
object DocDS: TDataSource
  DataSet = DocQuery
  Left = 208
  Top = 512
end
object DocQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'select * from case_document'
    'where idyear = :yr and idcase = :cs'
    'order by document_name')
  Params = <
    item
      DataType = ftUnknown
      Name = 'yr'
      ParamType = ptUnknown
    end
    item
      DataType = ftUnknown
      Name = 'cs'
      ParamType = ptUnknown
    end>
```

```
Left = 264
Top = 512
ParamData = <
  item
    DataType = ftUnknown
    Name = 'yr'
    ParamType = ptUnknown
  end
  item
    DataType = ftUnknown
    Name = 'cs'
    ParamType = ptUnknown
  end>
object DocQueryidcase_document: TIntegerField
  FieldName = 'idcase_document'
  Required = True
end
object DocQueryidyear: TIntegerField
  FieldName = 'idyear'
  Required = True
end
object DocQueryidcase: TIntegerField
  FieldName = 'idcase'
  Required = True
end
object DocQueryiddocument_type: TIntegerField
  FieldName = 'iddocument_type'
  Required = True
end
object DocQuerydocument_name: TWideStringField
  FieldName = 'document_name'
  Required = True
  Size = 45
end
object DocQuerydocument_text: TBlobField
  FieldName = 'document_text'
end
object DocQuerydatetime_last_modified: TDateTimeField
  FieldName = 'datetime_last_modified'
  Required = True
end
object DocQuerydatetime_last_scan: TDateTimeField
  FieldName = 'datetime_last_scan'
end
object DocQuerydatetime_created: TDateTimeField
  FieldName = 'datetime_created'
  Required = True
end
object DocQuerydatetime_last_accessed: TDateTimeField
  FieldName = 'datetime_last_accessed'
  Required = True
end
```

```
  end
  object SearchedDocTable: TZTable
    Connection = ZConnection1
    TableName = 'searched_document'
    Left = 344
    Top = 472
    object SearchedDocTableidregex_pattern: TIntegerField
      FieldName = 'idregex_pattern'
      Required = True
    end
    object SearchedDocTableidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object SearchedDocTableidcase: TIntegerField
      FieldName = 'idcase'
      Required = True
    end
    object SearchedDocTableidcase_document: TIntegerField
      FieldName = 'idcase_document'
      Required = True
    end
  end
  object SearchedDocDS: TDataSource
    DataSet = SearchedDocTable
    Left = 288
    Top = 456
  end
  object MatchedTextDS: TDataSource
    DataSet = MatchedTextTable
    Left = 440
    Top = 504
  end
  object MatchedTextTable: TZTable
    Connection = ZConnection1
    TableName = 'matched_text'
    Left = 488
    Top = 488
    object MatchedTextTableidmatched_text: TIntegerField
      FieldName = 'idmatched_text'
      Required = True
    end
    object MatchedTextTableidregex_pattern: TIntegerField
      FieldName = 'idregex_pattern'
      Required = True
    end
    object MatchedTextTableidyear: TIntegerField
      FieldName = 'idyear'
      Required = True
    end
    object MatchedTextTableidcase: TIntegerField
      FieldName = 'idcase'
```

```
    Required = True
  end
  object MatchedTextTableidcase_document: TIntegerField
    FieldName = 'idcase_document'
    Required = True
  end
  object MatchedTextTablematched_text: TMemoField
    FieldName = 'matched_text'
    Required = True
    BlobType = ftMemo
  end
end
object DeleteQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'delete from searched_document')
  Params = <>
  Left = 576
  Top = 504
end
object DCP_md51: TDCP_md5
  Id = 16
  Algorithm = 'MD5'
  HashSize = 128
  Top = 424
end
object UniqueQuery: TZQuery
  Connection = ZConnection1
  Params = <>
  Left = 656
  Top = 512
end
object PostProcessQuery: TZQuery
  Connection = ZConnection1
  SQL.Strings = (
    'select * from post_process')
  Params = <>
  Left = 720
  Top = 448
  object PostProcessQueryidpost_process: TIntegerField
    FieldName = 'idpost_process'
    Required = True
  end
  object PostProcessQueryidregex_pattern: TIntegerField
    FieldName = 'idregex_pattern'
    Required = True
  end
  object PostProcessQuerysql_command: TMemoField
    FieldName = 'sql_command'
    Required = True
    BlobType = ftMemo
  end
```

```
  end
  object PostProcessExecuteQuery: TZQuery
    Connection = ZConnection1
    Params = <>
    Left = 784
    Top = 472
  end
end
```

Filter.pas

```
unit Filter;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, DB, Dialogs, ZAbstractRODataset, ZAbstractDataset,
ZDataset, StdCtrls, Buttons, DBCtrls, ExtCtrls;

type
  TFilterForm = class(TForm)
    DBNavigator1:                   TDBNavigator;
    DBMemo1:                        TDBMemo;
    BitBtn2:                        TBitBtn;

    FilterDS:                       TDataSource;
    FilterQuery:                    TZQuery;
    FilterQueryidpost_process:      TIntegerField;
    FilterQueryidregex_pattern:     TIntegerField;
    FilterQuerysql_command:         TMemoField;

    procedure FormClose            (Sender: TObject;
                                    var Action: TCloseAction);
    procedure FormCreate           (Sender: TObject);
    procedure BitBtn2Click         (Sender: TObject);
    procedure FilterQueryBeforePost(DataSet: TDataSet);
    procedure FormCloseQuery       (Sender: TObject;
                                    var CanClose: Boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  protected
    procedure CreateParams         (var Params: TCreateParams);
override;
  end;

var
  FilterForm: TFilterForm;

implementation
```

```
Uses RegExMain;

{$R *.dfm}

procedure TFilterForm.BitBtn2Click(Sender: TObject);
begin
   Close;
end;

procedure TFilterForm.CreateParams(var Params: TCreateParams);
begin
   inherited;
   Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
   Params.WndParent := GetDesktopWindow;
end;

procedure TFilterForm.FilterQueryBeforePost(DataSet: TDataSet);
begin
   FilterQueryidregex_pattern.AsInteger :=
                                    Regular_Expression_Form.GetRegexID;
end;

procedure TFilterForm.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
   FilterQuery.Close;
end;

procedure TFilterForm.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
   if FilterQuery.State in [dsInsert, dsEdit] then
   begin
      MessageDlg('Please Complete Edits Before Closing', mtWarning,
                 [mbOK], 0);
      CanClose := False;
   end
   else
      CanClose := True;
end;

procedure TFilterForm.FormCreate(Sender: TObject);
begin
   Caption := Caption + Regular_Expression_Form.GetRegexName;
   FilterQuery.ParamByName('regexpattern').AsInteger :=
                                    Regular_Expression_Form.GetRegexID;
   FilterQuery.Open;
end;

end.
```

Filter.dfm

```
object FilterForm: TFilterForm
  Left = 0
  Top = 0
  BorderIcons = [biSystemMenu]
  Caption = 'Regular Expression Filter - '
  ClientHeight = 330
  ClientWidth = 589
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  DesignSize = (
    589
    330)
  PixelsPerInch = 96
  TextHeight = 13
  object DBNavigator1: TDBNavigator
    Left = 16
    Top = 8
    Width = 240
    Height = 25
    DataSource = FilterDS
    Hints.Strings = (
      'First record'
      'Prior record'
      'Next record'
      'Last record'
      'Insert record'
      'Delete record'
      'Edit record'
      'Save edit'
      'Cancel edit'
      'Refresh data')
    ParentShowHint = False
    ShowHint = True
    TabOrder = 0
  end
  object DBMemo1: TDBMemo
    Left = 16
    Top = 48
    Width = 564
    Height = 232
```

```
    Anchors = [akLeft, akTop, akRight, akBottom]
    BevelKind = bkTile
    DataField = 'sql_command'
    DataSource = FilterDS
    ScrollBars = ssVertical
    TabOrder = 1
  end
  object BitBtn2: TBitBtn
    Left = 476
    Top = 297
    Width = 105
    Height = 25
    Anchors = [akRight, akBottom]
    Caption = 'Close'
    DoubleBuffered = True
    Glyph.Data = {
      4E010000424D4E010000000000007600000028000000120000001200000001000
      040000000000D8000000000000000000000000100000010000000000000000000
      8000008000000080800080000000800080008080000080808000C0C0C0000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00888888888888
      8888880000008888800888888888880000008880FF008888888880000008880
      F00FF008888888000000880F0FF00FF008888800000080F0F78FF00FF0888800
      0000880F70078FF008888800000080F70FB0078FF088880000008800FBFBF007
      088888000000880FBFBFBFB008888800000080FBFBFBFBFBF088880000008800
      BFBFBFBF088888000000888800FBFBF0880888000000088888800B80880008800
      0000888888880088000008000000888888888888808880000008888888888888
      8808880000008888888888888888888000000}
    ParentDoubleBuffered = False
    TabOrder = 2
    OnClick = BitBtn2Click
  end
  object FilterDS: TDataSource
    DataSet = FilterQuery
    Left = 464
    Top = 8
  end
  object FilterQuery: TZQuery
    Connection = Regular_Expression_Form.ZConnection1
    BeforePost = FilterQueryBeforePost
    SQL.Strings = (
      'select * '
      'from post_process'
      'where idregex_pattern = :regexpattern')
    Params = <
      item
        DataType = ftUnknown
        Name = 'regexpattern'
        ParamType = ptUnknown
      end>
    Left = 528
    Top = 8
    ParamData = <
```

```
      item
        DataType = ftUnknown
        Name = 'regexpattern'
        ParamType = ptUnknown
      end>
    object FilterQueryidpost_process: TIntegerField
      FieldName = 'idpost_process'
      Required = True
    end
    object FilterQueryidregex_pattern: TIntegerField
      FieldName = 'idregex_pattern'
      Required = True
    end
    object FilterQuerysql_command: TMemoField
      FieldName = 'sql_command'
      Required = True
      BlobType = ftMemo
    end
  end
end
```

XML.pas

```
unit XML;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, DB, ZAbstractRODataset,
ZAbstractDataset, ZAbstractTable, ZDataset, Buttons, SHFolder, XMLDoc,
XMLIntf, OleCtrls, SHDocVw_EWB, EwbCore, EmbeddedWB, ExtCtrls;

type
  TXMLForm = class(TForm)
    SaveDialog1:                        TSaveDialog;

    CaseTable:                          TZTable;
    CaseTableidcase:                    TIntegerField;
    CaseTableidyear:                    TIntegerField;
    CaseTablecase_number:               TWideStringField;
    CaseTablecase_start_date:           TDateField;
    CaseTablecase_close_date:           TDateField;

    RegExTable:                         TZTable;
    RegExTableidregex_pattern:          TIntegerField;
    RegExTableinclude_in_export:        TSmallintField;
    RegExTablepattern:                  TMemoField;
    RegExTablesubexpression:            TIntegerField;
    RegExTablepattern_name:             TWideStringField;
    RegExTableexport_name:              TWideStringField;
    RegExTablepattern_description:      TMemoField;
```

```
    RegExTablecaseless:                TSmallintField;
    RegExTablesingle_line:             TSmallintField;
    RegExTableanchored:                TSmallintField;
    RegExTablemulti_line:              TSmallintField;
    RegExTableextended:                TSmallintField;
    RegExTableungreedy:                TSmallintField;

    IndicatorQuery:                    TZQuery;
    IndicatorQueryunique_matched_text: TMemoField;
    EmbeddedWB1:                       TEmbeddedWB;
    Panel1:                            TPanel;
    BitBtn1:                           TBitBtn;
    BitBtn2:                           TBitBtn;

    procedure BitBtn2Click            (Sender: TObject);
    procedure BitBtn1Click            (Sender: TObject);
    procedure FormCreate              (Sender: TObject);
    procedure BitBtn2MouseEnter       (Sender: TObject);
    procedure BitBtn2MouseLeave       (Sender: TObject);
  private
     Closed:                           Boolean;
     function GetSpecialFolderPath    (Folder: Integer): String;
    { Private declarations }
  public
    { Public declarations }
  end;

var
  XMLForm: TXMLForm;

implementation

{$R *.dfm}

Uses RegExMain;

procedure TXMLForm.FormCreate(Sender: TObject);
begin
   SaveDialog1.InitialDir := GetSpecialFolderPath(CSIDL_PERSONAL)+
                             '\Thesis_Docs\';
   Closed := False;
end;

procedure TXMLForm.BitBtn1Click(Sender: TObject);
var
   xmlDoc:         TXMLDocument;
   rootParsedNode: IXMLNode;
   rootcaseNode:   IXMLNode;
   indicatorNode:  IXMLNode;
   valuesNode:     IXMLNode;
   valueNode:      IXMLNode;
   stdateNode:     IXMLNode;
```

```
   enddateNode:    IXMLNode;
   typeNode:       IXMLNode;
   HoldQuery:      String;
begin
   screen.Cursor := crHourGlass;
   try
      BitBtn1.Enabled := False;
      HoldQuery := IndicatorQuery.SQL.Text;
      xmlDoc := TXMLDocument.Create(nil);
      try
         xmlDoc.Active := True;
         xmlDoc.Version := '1.0';
         xmlDoc.Encoding := 'utf-8';
         rootParsedNode := xmlDoc.AddChild('parsedcases');
         caseTable.Open;
         while (Not (CaseTable.EOF)) and (Not (Closed)) do
         begin
            Application.ProcessMessages;
            rootcaseNode := rootParsedNode.AddChild('case');
            rootcaseNode.Attributes['caseNumber'] :=
                                       CaseTablecase_number.AsString;
            stdateNode := rootcaseNode.AddChild('startdate');
            stdateNode.Text := CaseTablecase_start_date.AsString;
            enddateNode := rootcaseNode.AddChild('enddate');
            enddateNode.Text := CaseTablecase_close_date.AsString;
            indicatorNode := rootcaseNode.AddChild('indicators');
            RegExTable.Open;
            while (Not (RegExTable.EOF)) and (Not (Closed)) do
            begin
               Application.ProcessMessages;
               typeNode := indicatorNode.AddChild('type');
               typeNode.Attributes['patternType'] :=
                                    RegExTableExport_Name.AsString;
               valuesNode := typeNode.AddChild('values');
               IndicatorQuery.SQL.Text :=
                            StringReplace(IndicatorQuery.SQL.Text,
                            ':year',caseTableidyear.AsString,
                           [rfReplaceAll]);
               IndicatorQuery.SQL.Text :=
                                StringReplace(IndicatorQuery.SQL.Text,
                                ':case',caseTableidcase.AsString,
                               [rfReplaceAll]);
               IndicatorQuery.SQL.Text :=
                          StringReplace(IndicatorQuery.SQL.Text,
                         ':pattern',RegExTableidRegex_Pattern.AsString,
                         [rfReplaceAll]);
               IndicatorQuery.Open;
               while (Not (IndicatorQuery.EOF)) and (Not (Closed)) do
               begin
                  Application.ProcessMessages;
                  valueNode := valuesNode.AddChild('value');
                  valueNode.Text :=
```

```
                              indicatorQueryUnique_Matched_Text.AsString;
                     IndicatorQuery.Next;
                  end;
                  IndicatorQuery.Close;
                  IndicatorQuery.SQL.Text := HoldQuery;
                  RegExTable.Next;
               end;
               RegExTable.Close;
               caseTable.Next;
            end;
            caseTable.Close;
            if Not Closed then
               if SaveDialog1.Execute then
               begin
                  xmlDoc.SaveToFile(SaveDialog1.FileName);
                  EmbeddedWB1.Navigate(SaveDialog1.FileName);
               end;
            xmlDoc.Active := False;
         finally
            xmlDoc := nil;
         end;
         BitBtn1.Enabled := True;
      finally
         Screen.Cursor := crDefault;
      end;
end;

procedure TXMLForm.BitBtn2Click(Sender: TObject);
begin
   Closed := True;
   screen.Cursor := crDefault;
   Close;
end;

procedure TXMLForm.BitBtn2MouseEnter(Sender: TObject);
begin
   BitBtn2.Tag := screen.cursor;
   screen.Cursor := crDefault;
end;

procedure TXMLForm.BitBtn2MouseLeave(Sender: TObject);
begin
   Screen.Cursor := BitBtn2.Tag;
   BitBtn2.Tag := 0;
end;

function TXMLForm.GetSpecialFolderPath(Folder: Integer): String;
const
   SHGFP_TYPE_CURRENT = 0;
var
   Path: Array[0..MAX_PATH] of char;
begin
```

```
   if SUCCEEDED(SHGetFolderPath(
                0,folder,0,SHGFP_TYPE_CURRENT,@path[0])) then
      Result := path
   else
      Result := '';
end;

end.
```

XML.dfm

```
object XMLForm: TXMLForm
  Left = 0
  Top = 0
  BorderIcons = [biSystemMenu]
  Caption = 'XML Export'
  ClientHeight = 524
  ClientWidth = 810
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  Position = poDesigned
  OnCreate = FormCreate
  DesignSize = (
    810
    524)
  PixelsPerInch = 96
  TextHeight = 13
  object EmbeddedWB1: TEmbeddedWB
    Left = 0
    Top = 0
    Width = 809
    Height = 467
    Anchors = [akLeft, akTop, akRight, akBottom]
    TabOrder = 0
    Silent = False
    DisableCtrlShortcuts = 'N'
    UserInterfaceOptions = [EnablesFormsAutoComplete, EnableThemes]
    About = ' EmbeddedWB http://bsalsa.com/'
    PrintOptions.HTMLHeader.Strings = (
      '<HTML></HTML>')
    PrintOptions.Orientation = poPortrait
    ExplicitWidth = 769
    ExplicitHeight = 257
    ControlData = {
      4C0000009D530000443000000000000000000000000000000000000000000000
      000000004C0000000000000000000000001000000E0D057007335CF11AE690800
```

```
      2B2E12620800000000000004C000000011402000000000C000000000000046
      8000000000000000000000000000000000000000000000000000000000000
      0000000000000000100000000000000000000000000000000000}
  end
  object Panel1: TPanel
    Left = 0
    Top = 467
    Width = 810
    Height = 57
    Align = alBottom
    BevelInner = bvRaised
    BevelKind = bkTile
    BevelWidth = 2
    TabOrder = 1
    DesignSize = (
      806
      53)
    object BitBtn1: TBitBtn
      Left = 549
      Top = 13
      Width = 105
      Height = 25
      Anchors = [akRight, akBottom]
      Caption = 'Create XML'
      DoubleBuffered = True
      Glyph.Data = {
        36030000424D360300000000000036000000280000001000000010000000100
        180000000000000300000000000000000000000000000000000C0C0C0C0C0C0
        C0C0C0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0
        A0B0B0B0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFBFBFBFBFBFBF7F7F7F7F
        7F7F7F7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7F7F7F7F7F7FBFBFBFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFBFBFBFBFBFBF7F7F7F7F
        7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFFFFFFFFFFFFFBFBFBFBFBF7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFFFFFFBFBFBFBFBF7F
        7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFBFBFBFBFBFBF7F7F7F7F7F7F
        7F7FBFBFBFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFFFFFFBFBFBFBFBF7F
        7F7F7F7F7F7F7F7F7F7F7FFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7F7F7F7F7FFFFFFFFFFFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFBFBFBFBFBFBF7F7F7F7F
        7F7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFBFBFBFBFBFBF7F7F7F7F7F7F7F7F7FBFBFBFFFFFFFFFFFFFFFFFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0C0C0DFDFDFFFFFFFFFFFFFFFFFFFFFFF
        FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0
        DFDFDFFFFFFFFFFFFFFFFFF000000000000000007F7F7FFFFFFFFFFFFFFF
        FFA0A0A0C0C0C0C0C0C0C0C0C0C0C0C0CFCFCFDFDFDFDFDFDFDFDFDFDFDFDFDF
        DFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFDFC0C0C0C0C0C0C0C0C0}
      ParentDoubleBuffered = False
      TabOrder = 0
```

```
    OnClick = BitBtn1Click
  end
  object BitBtn2: TBitBtn
    Left = 677
    Top = 13
    Width = 105
    Height = 25
    Anchors = [akRight, akBottom]
    Cancel = True
    Caption = 'Close'
    DoubleBuffered = True
    Glyph.Data = {
      4E010000424D4E010000000000007600000028000000120000012000000010
      0400000000000D800000000000000000000000010000000100000000000000000000
      8000008000000080800080000000800080008080000080808000C0C0C0000000
      FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00888888888888
      8888880000008888800888888888880000008880FF008888888880000008880
      F00FF008888888000000880F0FF00FF008888800000080F0F78FF00FF0888800
      0000880F70078FF008888800000080F70FB0078FF088880000008800FBFBF007
      088888000000880FBFBFBFB008888800000080FBFBFBFBFBF088880000008800
      BFBFBFBF088888000000888800FBFBF0880888000000888888800B80880008800
      0000888888880088000008000000088888888888880888000000888888888888
      8808880000008888888888888888888000000}
    ParentDoubleBuffered = False
    TabOrder = 1
    OnClick = BitBtn2Click
    OnMouseEnter = BitBtn2MouseEnter
    OnMouseLeave = BitBtn2MouseLeave
  end
end
object CaseTable: TZTable
  Connection = Regular_Expression_Form.ZConnection1
  TableName = 'case_file'
  Left = 16
  Top = 264
  object CaseTableidcase: TIntegerField
    FieldName = 'idcase'
    Required = True
  end
  object CaseTableidyear: TIntegerField
    FieldName = 'idyear'
    Required = True
  end
  object CaseTablecase_number: TWideStringField
    FieldName = 'case_number'
    Required = True
    Size = 10
  end
  object CaseTablecase_start_date: TDateField
    FieldName = 'case_start_date'
    Required = True
  end
  object CaseTablecase_close_date: TDateField
```

```
      FieldName = 'case_close_date'
    end
  end
  object RegExTable: TZTable
    Connection = Regular_Expression_Form.ZConnection1
    Filter = 'include_in_export = 1'
    Filtered = True
    TableName = 'regex_pattern'
    Left = 56
    Top = 264
    object RegExTableidregex_pattern: TIntegerField
      FieldName = 'idregex_pattern'
      Required = True
    end
    object RegExTableinclude_in_export: TSmallintField
      FieldName = 'include_in_export'
      Required = True
    end
    object RegExTablepattern: TMemoField
      FieldName = 'pattern'
      Required = True
      BlobType = ftMemo
    end
    object RegExTablesubexpression: TIntegerField
      FieldName = 'subexpression'
    end
    object RegExTablepattern_name: TWideStringField
      FieldName = 'pattern_name'
      Required = True
      Size = 45
    end
    object RegExTableexport_name: TWideStringField
      FieldName = 'export_name'
      Size = 10
    end
    object RegExTablepattern_description: TMemoField
      FieldName = 'pattern_description'
      Required = True
      BlobType = ftMemo
    end
    object RegExTablecaseless: TSmallintField
      FieldName = 'caseless'
      Required = True
    end
    object RegExTablesingle_line: TSmallintField
      FieldName = 'single_line'
      Required = True
    end
    object RegExTableanchored: TSmallintField
      FieldName = 'anchored'
      Required = True
    end
```

```
    object RegExTablemulti_line: TSmallintField
      FieldName = 'multi_line'
      Required = True
    end
    object RegExTableextended: TSmallintField
      FieldName = 'extended'
      Required = True
    end
    object RegExTableungreedy: TSmallintField
      FieldName = 'ungreedy'
      Required = True
    end
  end
  object IndicatorQuery: TZQuery
    Connection = Regular_Expression_Form.ZConnection1
    SQL.Strings = (
      'select unique_matched_text'
      'from unique_matched_text'
      'where'
      '   idyear = :year and'
      '   idcase = :case and'
      '   idregex_pattern = :pattern'
      'order by'
      '   unique_matched_text')
    Params = <
      item
        DataType = ftUnknown
        Name = 'year'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'case'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'pattern'
        ParamType = ptUnknown
      end>
    Left = 96
    Top = 264
    ParamData = <
      item
        DataType = ftUnknown
        Name = 'year'
        ParamType = ptUnknown
      end
      item
        DataType = ftUnknown
        Name = 'case'
        ParamType = ptUnknown
```

```
      end
      item
        DataType = ftUnknown
        Name = 'pattern'
        ParamType = ptUnknown
      end>
    object IndicatorQueryunique_matched_text: TMemoField
      FieldName = 'unique_matched_text'
      Required = True
      BlobType = ftMemo
    end
  end
  object SaveDialog1: TSaveDialog
    DefaultExt = '*.xml'
    Filter = 'eXtenisble Markup Language|*.XML'
    Options = [ofOverwritePrompt, ofHideReadOnly,
               ofShareAware, ofEnableSizing]
    Left = 144
    Top = 264
  end
end
```

The Regular Expression Parser application was developed using Delphi Professional 2010, however; to recompile this application, additional components are needed. To connect to the MySQL database, the ZEOS database components version 7.0 were used. These components can be found here: http://zeos.firmos.at/. To calculate the MD5 Hash values necessary to save matched text to the unique_matched_text table, the dcpCrypt components were used, which can be found here: http://www.cityinthesky.co.uk/opensource/dcpcrypt. To conduct the regular expression searches, the PCRE Regex component was used, and can be found here: http://www.regular-expressions.info/delphi.html. Finally, while wordprocessing capabilities were not part of this application, the TRichView components Version 12.0 were still utilized specifically to extract the pure text from the documents and can be found here: http://www.trichview.com/.

Appendix E – Regular Expression Service Application Source Code

ThesisService.pas

```
unit ThesisService;

interface

uses Windows, Messages, SysUtils, Classes, Graphics, Controls, SvcMgr,
Dialogs, Registry, SvcProcess;

type
  TThesisRegExService = class(TService)
    procedure ServiceAfterInstall(Sender: TService);
    procedure ServiceExecute(Sender: TService);
    procedure ServiceStart(Sender: TService; var Started: Boolean);
    procedure ServiceStop(Sender: TService; var Stopped: Boolean);
  private
    { Private declarations }
  public
    function GetServiceController: TServiceController; override;
    { Public declarations }
  end;

var
  ThesisRegExService: TThesisRegExService;

implementation

{$R *.DFM}

procedure ServiceController(CtrlCode: DWord); stdcall;
begin
  ThesisRegExService.Controller(CtrlCode);
end;

function TThesisRegExService.GetServiceController: TServiceController;
begin
  Result := ServiceController;
end;

procedure TThesisRegExService.ServiceAfterInstall(Sender: TService);
var
   Reg:  TRegistry;
begin
   Reg := TRegistry.Create(KEY_READ or KEY_WRITE);
   try
      reg.RootKey := HKEY_LOCAL_MACHINE;
      if Reg.OpenKey(
              '\SYSTEM\CurrentControlSet\Services\' + Name, False) then
      begin
```

```
        Reg.WriteString('Description',
                    'Partial fulfillment of Regis University Masters '+
                    'Thesis.  This service parses the documents in the '+
                    'Thesis DB for pattern matches using Reg Expressions');
         Reg.CloseKey;
      end;
   finally
      FreeAndNil(Reg);
   end;
end;

procedure TThesisRegExService.ServiceExecute(Sender: TService);
const
   secBetweenRuns = 10;
var
   currCount:  Integer;
begin
   currCount := 0;
   while Not Terminated do
   begin
      Inc(currCount);
      if CurrCount >= SecBetweenRuns then
      begin
         currCount := 0;
         try
            if MainProcessForm.CheckForDocumentsToScan then
            begin
               try
                  MainProcessForm.ProcessDocuments;
               except

               end;
            end;
         except

         end;
      end;
      if Not Terminated then
         sleep(1000);
      ServiceThread.ProcessRequests(False);
   end;
end;

procedure TThesisRegExService.ServiceStart(Sender: TService;
                                           var Started: Boolean);
begin
   Started := True;
end;

procedure TThesisRegExService.ServiceStop(Sender: TService;
                                          var Stopped: Boolean);
begin
```

```
   Stopped := True;
end;

initialization
   if (ParamCount >= 1) and (CompareText(
                             '/uninstall', ParamStr(1))= 0) then
   begin
      WinExec('cmd.exe /c net stop ThesisRegExService', SW_HIDE);
      Sleep(300);
   end;

finalization
   if (ParamCount >=1) and (CompareText(
                            '/install', ParamStr(1))=0) then
      WinExec('cmd.exe /c net start ThesisRegExService', SW_HIDE);

end.
```

SvcProcess.pas

```
unit SvcProcess;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, DB,
Controls, Forms, Dialogs, ZConnection, ZAbstractRODataset, ZDataset,
ZAbstractDataset, RVScroll, RichView, RVEdit, DBRV, RVStyle,
rvGetTextW, PerlRegEx, ZAbstractTable, DCPmd5, DCPcrypt2;

type
  TMainProcessForm = class(TForm)
    ZConnection1: TZConnection;
    DocListQuery: TZQuery;
    DocCountQuery: TZQuery;
    DocCountQueryDocCount: TLargeintField;
    DocumentTextQuery: TZQuery;
    DeleteMatches: TZQuery;
    PerlRegEx1: TPerlRegEx;
    RVStyle1: TRVStyle;
    DataSource1: TDataSource;
    RegExpQuery: TZQuery;
    RegExpQueryidregex_pattern: TIntegerField;
    RegExpQueryinclude_in_export: TSmallintField;
    RegExpQuerypattern: TMemoField;
    RegExpQuerysubexpression: TIntegerField;
    RegExpQuerypattern_name: TWideStringField;
    RegExpQueryexport_name: TWideStringField;
    RegExpQuerypattern_description: TMemoField;
    RegExpQuerycaseless: TSmallintField;
    RegExpQuerysingle_line: TSmallintField;
    RegExpQueryanchored: TSmallintField;
```

```
    RegExpQuerymulti_line: TSmallintField;
    RegExpQueryextended: TSmallintField;
    RegExpQueryungreedy: TSmallintField;
    DBRichView1: TDBRichView;
    DataSource2: TDataSource;
    SearchedDocTable: TZTable;
    SearchedDocTableidregex_pattern: TIntegerField;
    SearchedDocTableidyear: TIntegerField;
    SearchedDocTableidcase: TIntegerField;
    SearchedDocTableidcase_document: TIntegerField;
    CaseListQuery: TZQuery;
    DocListQueryidcase: TIntegerField;
    DocListQueryidyear: TIntegerField;
    DocListQueryidcase_document: TIntegerField;
    CaseListQueryidcase: TIntegerField;
    DocumentTextQueryidcase_document: TIntegerField;
    DocumentTextQueryidcase: TIntegerField;
    DocumentTextQueryidyear: TIntegerField;
    DocumentTextQuerydocument_text: TBlobField;
    DocumentTextQuerydatetime_last_scan: TDateTimeField;
    MatchedTextTable: TZTable;
    MatchedTextTableidmatched_text: TIntegerField;
    MatchedTextTableidregex_pattern: TIntegerField;
    MatchedTextTableidyear: TIntegerField;
    MatchedTextTableidcase: TIntegerField;
    MatchedTextTableidcase_document: TIntegerField;
    MatchedTextTablematched_text: TMemoField;
    PostProcessQuery: TZQuery;
    PostProcessQueryidpost_process: TIntegerField;
    PostProcessQueryidregex_pattern: TIntegerField;
    PostProcessQuerysql_command: TMemoField;
    PostProcessExecuteQuery: TZQuery;
    UniqueQuery: TZQuery;
    DCP_md51: TDCP_md5;
    DeleteFromSearchedDocQuery: TZQuery;
    RegExpQueryinclude_in_scan: TSmallintField;
    procedure PerlRegEx1Match(Sender: TObject);
  private
    function GetMD5Value(S:ANSIString): ANSIString;

    procedure CondenseToUnique;
    procedure FilterRecords;
    procedure UpdateMatchedTextTable(REID,YrID,CsID,DcID:Integer;
                                     OutString: String);
    procedure UpdateSearchedDocument(REID, YrID, CsID, DcID: Integer);
    procedure SetPerlRegExOptions;
    { Private declarations }
  public
    function CheckForDocumentsToScan: Boolean;
    procedure ProcessDocuments;
    { Public declarations }
  end;
```

```
var
  MainProcessForm: TMainProcessForm;

implementation

{$R *.dfm}

function TMainProcessForm.CheckForDocumentsToScan: Boolean;
begin
   ZConnection1.Connected := True;
   try
      DocCountQuery.Open;
      try
         if DocCountQueryDocCount.AsInteger > 0 then
            Result := True
         else
            Result := False;
      finally
         DocCountQuery.Close;
      end;
   finally
      ZConnection1.Connected := False;
   end;
end;

procedure TMainProcessForm.CondenseToUnique;
begin
   MatchedTextTable.Refresh;
   MatchedTextTable.First;
   while Not (MatchedTextTable.EOF) do
   begin
      Application.ProcessMessages;
      UniqueQuery.SQL.Clear;
      UniqueQuery.SQL.Text :=
        'insert into unique_matched_text '+
        '(idregex_pattern,idcase,idyear,md5_value,unique_matched_text) ' +
        'values ('+MatchedTextTableidregex_pattern.AsString +','+
                  MatchedTextTableidcase.AsString +','+
                  MatchedTextTableidYear.AsString +",'"+
                  String(GetMD5Value(ANSIString(UpperCase(
                  MatchedTextTableMatched_Text.AsString))))+"',"+
                  QuotedStr(MatchedTextTableMatched_Text.AsString)+')';
      try
         UniqueQuery.ExecSQL;
      except
        //Expecting key violation for duplicate matches within the same case
        //thus the key violation error reported by the database is silenced.
      end;
      MatchedTextTable.Next;
   end;
   MatchedTextTable.Close;
```

```
end;

procedure TMainProcessForm.FilterRecords;
begin
   PostProcessQuery.Open;
   while Not PostProcessQuery.EOF do
   begin
      PostProcessExecuteQuery.SQL.Clear;
      PostProcessExecuteQuery.SQL.Text :=
                                  PostProcessQuerySQL_Command.AsString;
      PostProcessExecuteQuery.ExecSQL;
      PostProcessQuery.Next;
   end;
   PostProcessQuery.Close;
end;

function TMainProcessForm.GetMD5Value(S: ANSIString): ANSIString;
Var
   StACrypt, StCrypt: ANSIString;
   Digest: array [0..15] of Byte;
   I: Integer;
begin
   StACrypt := S;
   DCP_md51.Init;
   DCP_md51.UpdateStr(StACrypt);
   DCP_md51.Final(Digest);
   StCrypt := '';
   for I := 0 to 15 do
      StCrypt:=StCrypt+ANSIString(IntToHex(Digest[I], 2));
   Result := StCrypt;
end;

procedure TMainProcessForm.PerlRegEx1Match(Sender: TObject);
var
   OutString:  UTF8String;
begin
   if RegExpQuerysubexpression.AsString <> '' then
      OutString := PerlRegEx1.SubExpressions[
                                    RegExpQuerysubexpression.AsInteger]
   else
      OutString := PerlRegEx1.MatchedExpression;
   UpdateMatchedTextTable(RegExpQueryidRegEx_Pattern.AsInteger,
                          DocListQueryidYear.AsInteger,
                          DocListQueryidcase.AsInteger,
                          DocListQueryidcase_document.AsInteger,
                          UTF8toString(OutString));
   PerlRegEx1.MatchAgain;
end;

procedure TMainProcessForm.ProcessDocuments;
var
   SearchStream:  UTF8String;
```

```
begin
   ZConnection1.Connected := True;
   DeleteFromSearchedDocQuery.ExecSQL;
   MatchedTextTable.Open;
   SearchedDocTable.Open;
   try
      CaseListQuery.Open;
      try
         while Not CaseListQuery.EOF do
         begin
            DeleteMatches.SQL.Text :=
                'delete from unique_matched_text where idcase = ' +
                CaseListQueryidcase.AsString;
            DeleteMatches.ExecSQL;
            DocListQuery.SQL.Text :=
            'select idcase, idyear, idcase_document from case_document '+
            'where idcase = ' + CaseListQueryidcase.AsString;
            DocListQuery.Open;
            try
               while Not DocListQuery.EOF do
               begin
                DocumentTextQuery.SQL.Text :=
                'select datetime_last_scan,idcase_document,idcase,idyear, '+
                'document_text from case_document where idcase_document = '+
                  DocListQueryidcase_document.AsString;
                  DocumentTextQuery.Open;
                  try
                     if DocumentTextQuerydocument_Text.AsString
                        <> '' then
                     begin
                        RegExpQuery.Open;
                        try
                           while Not RegExpQuery.EOF do
                           begin
                              if RegExpQueryinclude_in_scan.AsInteger
                                 = 1 then
                              begin
                                 PerlRegEx1.RegEx := UTF8Encode(
                                          RegExpQueryPattern.AsString);
                                 SearchStream := UTF8Encode(
                                             GetAllText(DBRichview1));
                                 PerlRegEx1.Subject := SearchStream;
                                  UpdateSearchedDocument(
                                  RegExpQueryidRegEx_Pattern.AsInteger,
                                  DocListQueryidyear.AsInteger,
                                  DocListQueryidcase.AsInteger,
                                  DocListQueryidcase_document.AsInteger);
                                    SetPerlRegExOptions;
                                    PerlRegEx1.Match;
                              end;
                              RegExpQuery.Next;
                           end;
```

```
                        finally
                           RegExpQuery.Close;
                        end;
                        DocumentTextQuery.Edit;
                        DocumentTextQueryDateTime_last_Scan.AsDateTime
                          := Now;
                        DocumentTextQuery.Post;
                     end;
                  finally
                     DocumentTextQuery.Close;
                  end;
                  DocListQuery.Next;
               end;
            finally
               DocListQuery.Close;
            end;
            CaseListQuery.Next;
         end;
      finally
         CaseListQuery.Close;
      end;
      FilterRecords;
      CondenseToUnique;
      MatchedTextTable.Close;
      SearchedDocTable.Close;
   finally
      DeleteFromSearchedDocQuery.ExecSQL;
      ZConnection1.Connected := False;
   end;
end;

procedure TMainProcessForm.SetPerlRegExOptions;
begin
   PerlRegEx1.options := [];
   if RegExpQuerycaseless.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preCaseLess];
   if RegExpQuerysingle_line.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preSingleLine];
   if RegExpQuerymulti_line.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preMultiLine];
   if RegExpQueryanchored.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preAnchored];
   if RegExpQueryextended.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preExtended];
   if RegExpQueryungreedy.AsInteger = 1 then
      PerlRegEx1.options := PerlRegEx1.options + [preUngreedy];
end;

procedure
TMainProcessForm.UpdateMatchedTextTable(REID,YrID,CsID,DcID:Integer;
                                        OutString: String);
begin
```

```
   MatchedTextTable.Insert;
   MatchedTextTableidregex_pattern.Value := REID;
   MatchedTextTableidyear.Value          := YrID;
   MatchedTextTableidcase.Value          := CsID;
   MatchedTextTableidcase_document.Value := DcID;
   MatchedTextTableMatched_Text.AsString := OutString;
   MatchedTextTable.Post;
end;

procedure TMainProcessForm.UpdateSearchedDocument(REID, YrID,
                                                  CsID, DcID: Integer);
begin
   SearchedDocTable.Insert;
   SearchedDocTableidregex_pattern.Value := REID;
   SearchedDocTableidyear.Value          := YrID;
   SearchedDocTableidcase.Value          := CsID;
   SearchedDocTableidcase_document.Value := DcID;
   SearchedDocTable.Post;
end;

end.
```

The Regular Expression Service application was developed using Delphi Professional 2010, however; to recompile this application, additional components are needed. To connect to the MySQL database, the ZEOS database components version 7.0 were used. These components can be found here: http://zeos.firmos.at/. To calculate the MD5 Hash values necessary to save matched text to the unique_matched_text table, the dcpCrypt components were used, which can be found here: http://www.cityinthesky.co.uk/opensource/dcpcrypt. To conduct the regular expression searches, the PCRE Regex component was used, and can be found here: http://www.regular-expressions.info/delphi.html. Finally, while wordprocessing capabilities were not part of this application, the TRichView components Version 12.0 were still utilized specifically to extract the pure text from the documents and can be found here: http://www.trichview.com/.