



Data Miniaturization Technology (DMT)

**Miniaturization and Acceleration Technology for
Next Generation Mobile and Storage Applications**

A White Paper

WindSpring, Inc.

www.windspring.com

jarchbold@windspring.com

Table of Contents

Objectives 1
 Introduction..... 1
 DMT Architecture Overview..... 2
 DMT Dictionary Deployment Options 2
 Core DMT Processes 3
 DMT Codec Operation 3
 DMT Theoretical Basis..... 5
 Miniaturization Performance 9
 DMT Products 10
 Integration Steps 10
 Further Questions..... 10

Objectives

This document describes the application of WindSpring’s Data Miniaturization Technology (DMT) for solving the costly problems of data transport latency, storage space constraint and network cost that are typically experienced by vendors and users of Mobile, Wireless, Digital Mapping and High-Speed storage applications.

The objectives of this White Paper are:

- Introduce the reader to DMT
- Demonstrate how DMT solves performance and cost problems in mobile and storage applications
- Describe the theoretical basis and core processes that underpin the unique performance advantages of DMT

Introduction

WindSpring’s DMT is a ground-breaking solution for the data challenges presented by the next

generation of Wireless, Digital Mapping, High-Speed Storage and Search applications.

These challenges include:

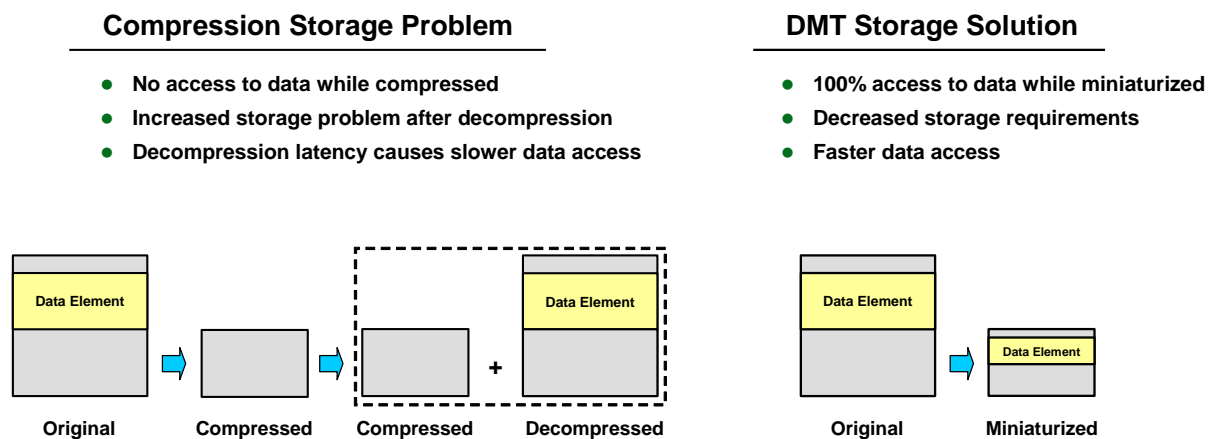
- **Data Storage Constraint.** These applications often require data access and manipulation in storage-constrained environments (e.g. mobile devices and navigation devices)
- **Transmission Latency.** Next generation applications require high-speed and reliable data access and transmission, across networks that often exhibit unpredictable loss and latency (e.g. Wireless networks)
- **Network and Hardware Cost.** Growing network and hardware cost can prohibit the deployment and operation of next generation, intelligent applications

DMT provides a flexible and powerful software solution to these problems, by enabling data to be miniaturized to a fraction of its original size, and then fully manipulated (i.e., seek, search, edit and retrieve) in its loss-less Miniaturized Data Format (MDF).

The key advantages of DMT are:

- DMT enables storage-constrained devices and networks to store increased volumes of data, without the need to increase storage or network capacity.
- DMT also enables throughput-constrained applications to access and transmit data at significantly higher speeds than otherwise available with conventional storage or mobile solutions.

Figure 1. Key Advantages of Miniaturization (DMT) Over Compression



As illustrated in Figure 1. above, compression does not enable access to individual data elements once a file has been compressed. And the decompression process requires storage for both the compressed and decompressed files on the target system.

DMT supersedes compression, by allowing data to be manipulated in the miniaturized state. Using DMT, data never need to be decompressed, any data element can be accessed in the “miniaturized” state, and data storage requirements are significantly reduced.

These factors all contribute to faster seek, search, edit and display of the data in the miniaturized state.

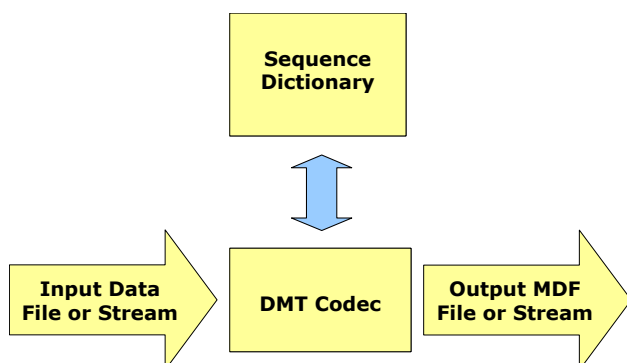
DMT is uniquely suited for space-constrained environments such as cell phones, Smartphones, PDA’s, DVD and in-dash navigation systems. DMT also increases data transfer and data access speeds regardless of the available storage space, constrained or unconstrained.

Patents have been filed in all the major geographies including the EU, Japan and the US, and thus far granted in the United States, Australia and Singapore.

DMT Architecture Overview

DMT has been engineered with the simplicity, scalability and functionality required to accelerate data throughput and lower operating costs for those applications that had been restricted due to space and speed constraints.

Figure 2. DMT Data Flow



As illustrated in Figure 2. above, the key steps in the DMT data miniaturization process are as follows:

1. **Sequence Dictionary Discovery.** The input data file (or stream) is analyzed to create an

appropriate Sequence Dictionary. This dictionary can be:

- Created for the first time, following analysis of the input data;
 - Learned over time, based on the changing nature of the application data;
 - Selected from an existing dictionary; or
 - Adapted from an existing dictionary.
2. **Encoding.** Input data is then transformed by the DMT Codec to create a Miniaturized Data Format (MDF) version of the original file or stream. MDF data is an optimally encoded series of symbols or recurring sequences that represent an exact one-to-one representation of the original data, but is a fraction of the size of the original file or stream.
3. **Seek, Search, Edit and Render.** The Sequence Dictionary operates with the MDF data to enable full manipulation of any data element within the encoded data.

DMT may be configured for a wide range of data types typically encountered in Mobile, Wireless, Digital Mapping and Navigation applications.

DMT Dictionary Deployment Options

DMT offers two primary dictionary deployment options:

- **Internal (File-Specific) Dictionaries.** DMT dictionaries are usually developed and deployed with individual MDF files. Applications where internal (file-specific) dictionaries are appropriate include Mobile XML and HTML synchronization, Telematics and in-car navigation systems.
- **External (Common) dictionaries.** One of the key advantages of DMT is its ability to develop a Common dictionary from a set of files of similar context (e.g. multiple word files or GIF images), and then pre-deploy that dictionary for encoding and manipulation of all files used by the application that have similar format or context. Applications where common (detached) dictionaries are appropriate include Mobile GPS Mapping, Remote computing and remote database applications, Mobile search, and web-based mapping services.

In addition to the flexibility to extract and detach common dictionaries from a set of files, DMT dictionaries may be learned and updated over time. The implementation of learned dictionaries is discussed in more detail in DMT Integration Options.

Core DMT Processes

The key to DMT's unique capabilities reside in the technology's innovative and patent pending Miniaturized Data Format, and Multi-Index Pointer and Exception Handling processes.

These processes include:

- **Quantum Symbol, Quantum Sequence and Terminal Sequence Pointer Identification:**
 - Enables automatic identification of the minimum indivisible units of searchable or editable information as a Quantum Symbol. Examples include ASCII characters, RGB strings and binary vectors.
 - Enables automatic identification and compilation of the repeated sequences of these symbols, regardless of the data context, as a Quantum Sequence or Terminal Sequence Pointer.
 - Facilitates the collection of frequency statistics during this process also enables ongoing Sequence Dictionary learning.
 - Using these processes as an encoding system enables the high-speed seek, search, edit and display of MDF format data for a single file.
- **Fixed Bit-Length Encoding:** The use of multiple fixed bit-length encoding leads to the most efficient (smallest) Sequence Dictionary for a given discovered Quantum Symbol and Terminal Sequence Pointer set.
- **Multi-Index Pointer System (MIPS):**
 - The most efficient miniaturization for a *single file or stream*, while maintaining seek, search and edit capabilities
 - The ability to develop *Common Sequence Dictionaries* in a format that provides for the ability to extend the selected dictionary (or a sub-set of that dictionary) to multiple files or streams

- The most efficient miniaturization for *multiple files or streams* across a particular data context, while maintaining the ability to seek, search and edit individual data sets
- The operation and integration of *multiple Sequence Dictionaries* across multiple files or streams

- **Exception Handling:**

- The encoding of any data in the original file or stream, that does not occur in the Sequence Dictionary
- The application of Common Sequence Dictionaries across the entire data context (multiple files or streams)
- Ongoing Sequence Dictionary learning
- Acceleration of data search within miniaturized files, in the case where Common Sequence Dictionaries have been applied to multiple files or streams

DMT Codec Operation

Two proprietary Codec implementations have been developed, and are available with DMT. Both implementations utilize WindSpring's core DMT processes, and can be executed for encoding, manipulating and rendering MDF data. These Codecs are:

- **Quantum Pair (QP).** The QP Codec uses a data pairing technique that is suited to files with medium to long sequences of repeated data. This codec is suited to small to medium size files.
- **Quantum Slide (QS).** The Quantum Slide Codec uses a technique that is suited to files with short to medium length repeating sequences. It is also capable of analyzing very large files and generating a dictionary that describes the whole of the file.

The two Codecs exhibit different miniaturization and speed performance characteristics with different types of data. These performance characteristics are discussed in the DMT Performance section in this document.

QP Codec Operation

QP Codec operation is as follows:

- **QP Sequence Dictionary Discovery.** The QP Codec uses the MIPS architecture to discover recurring sequences in the input data. The basic unit of encoding is termed a Quantum Symbol. Two adjacent Quantum Symbols are paired to form a Quantum Sequence. Quantum Sequences are then further paired to create Terminal Sequences. Terminal Sequence Pointers are then created as pointers into the Sequence Table, to identify each Terminal Sequences. Finally, the Sequence Dictionary is then built from these pointers.
- **QP Sequence Dictionary Size.** The size of the QP dictionary is controlled by the QP Codec Parameters. These parameters include:
 - QP Dictionary size (MB). This is amount of memory in Mbytes reserved for Sequence Dictionary analysis.
 - Block Size (bytes). The original data is encoded in segments equal to the Block Size in bytes.
 - Maximum Sequence Length (bytes). This is the longest sequence in bytes that can be discovered during the Sequence Dictionary discovery process.

In the case of learning dictionaries, statistical data is kept for the sequences that are included in the dictionary. This information tracks the number of times that the particular sequence has been used. Ongoing selection or rejection of candidate sequences is based on the target dictionary size (which affects miniaturization performance) and the frequency of occurrence of individual sequences.
- **QP Encoding.** The encoding process commences with a Quantum Symbol and scans the forward pointer chain, selecting the highest frequency Quantum Sequences until it encounters a Terminal Sequence. At this point the Terminal Sequence pointer is written to the MDF output. The process continues until all the input data has been encoded. If a match cannot be found in the Sequence Dictionary the data is sent to the output as a Quantum Symbol.

- **QP Decoding.** The decoding process commences with a token of MDF data. The MDF data points to the specific Terminal Sequence entry in the dictionary that was used during encoding, and the entry is then used to retrieve the Quantum Sequences that were used during encoding. Eventually the entry points to a Quantum Symbol and the decoding process is complete. At this point the data series created from the Quantum Symbols is written to the output (rendered MDF data). The process continues until all the requested MDF data has been rendered. If an exception is encountered in the MDF data it is rendered to the output exactly as encountered.

QS Codec Operation

QS Codec operation is as follows:

- **QS Sequence Dictionary Discovery.** The QS Codec Sequence Discovery is essentially the same as for QP, with the following exceptions:
 - The QS Codec is forward looking only i.e. it will not pair a Quantum Sequence with a preceding Quantum Sequence
 - Terminal Sequence Pointers are ordered in the Sequence Dictionary in terms of the length of the symbols they represent
 - During the dictionary finalization process, Terminal Sequences are expanded so that no Quantum Sequences exist in the finalized Sequence Dictionary

Like QP, the size of the QS Sequence Dictionary is controlled by the parameters used for analysis. QS parameters include:

- Windows Size (bytes). The original data is analyzed in segments equal to the Window Size in bytes, during the Sequence Dictionary discovery process.
- Maximum Sequence Length (bytes). This is the longest sequence in bytes that can be discovered during the Sequence Dictionary discovery process.
- Mean Rank Adjustment. This is the number of additional sequences that fall below the mean rank, for a particular sequence length, that will be included in the Sequence Dictionary during the seed process
- Seed Size (bytes). This is the number of bytes analyzed at any time for Sequence Dictionary discovery
- Block Size (bytes). The original data is encoded in segments equal to the Block Size in bytes.
- Dictionary Memory Limit (bytes). This is amount of memory in Mbytes reserved for Sequence Dictionary analysis.
- **QS Encoding**. The encoding process starts with a Quantum Symbol and scans the Sequence Dictionary for the longest length Terminal Sequence that matches the first Quantum Symbol. If a match is found the Terminal Sequence pointer is written to the output. The process continues until all the input data has been encoded. If a match cannot be found in the Sequence Dictionary the data is sent to the output as a Quantum Symbol.
- **QS Decoding**. The decoding process starts with a token of MDF data. This data from the MDF data points to the Terminal Sequence entry in the dictionary that was used to encode. The entry is used to retrieve the *Terminal* Sequence that was used during encoding. At this point the data series created from the Quantum Symbols is written to the output. The process continues until all the requested MDF data has been rendered. If an exception is encountered in the MDF data it is rendered to the output exactly as encountered.

DMT Theoretical Basis

Miniaturization Ratio and File Size Reduction

Miniaturization Ratio (MR) is defined for internal and external Sequence Dictionary cases as follows:

$$MR_{INT} = \frac{\text{Size (ID)}}{\text{Size (MDF) + Size (SD)}}$$

$$MR_{EXT} = \frac{\text{Size (ID)}}{\text{Size (MDF)}}$$

Where:

MR_{INT} = Miniaturization Ratio achieved using an internal (file-specific) Sequence Dictionary

MR_{EXT} = Miniaturization Ratio achieved using an external (common) Sequence Dictionary

Size (ID) = Size of Input Data (bits)

Size (MDF) = Size of MDF format data (bits)

Size (SD) = Size of Sequence Dictionary (bits)

File Size Reduction (FSR) represents the percentage reduction in the size of the original data, achieved with DMT. FSR is defined as:

$$FSR (\%) = \frac{MR - 1}{MR}$$

As an example, 80% FSR equates to a MR of 5:1.

Quantum Symbol and Quantum Sequence Frequency

The Frequency of each Quantum Symbol in the input data is calculated as:

$$Freq [QS(n)_j] = \frac{k \times \sum [QS(n)_j]}{\text{Size (ID)}}$$

Where:

k = the number of bits in each Quantum Symbol (usually set at 8 bits)

n = the length of the Quantum Sequence in terms of Quantum Symbols

$QS(n)$ = Quantum Sequence of length n

Size (OD) = Original Data Size (bits)

In the specific case of Quantum Symbols:

$$\sum_{i=1}^{i=QS(1)_{MAX}} \text{Freq} [QS(1)_i] = 1$$

Where:

$QS(1)_{MAX}$ = The maximum number of discrete Quantum Symbols in the Input Data.

As an example, most text-based documents usually contain 256 Quantum Symbols (ASCII characters).

DMT Quantum Symbol Miniaturization

The **relative** contribution of each Quantum Symbol to Miniaturization Ratio is given as:

$$MR_{QS(1)} = \frac{BL_{ID}}{PL_{MDF}}$$

Where:

BL_{ID} = the bit length for Quantum Symbols identified in the input data.

PL_{MDF} = the pointer bit length used for encoding MDF data

Note: Relative contribution to MR is measured for Quantum Symbols only, and excludes the size of the Sequence Dictionary

The bit length for Quantum Symbols identified in the input data is generally 8 bits for text-based data, and 24 bits for RGB data. DMT has the flexibility to discover and encode Quantum Symbols of any bit length (e.g. as for binary data), as long as this length is constant for any single input data set.

The pointer bit length for MDF encoding (PL_{MDF}) is directly determined by the number of entries in the Sequence Dictionary, as follows:

$$PL_{MDF} = INT \left[\frac{LN (SD \text{ Entries})}{LN (2)} \right] + 1$$

Where:

$LN (SD \text{ Entries})$ = the Natural Logarithm of the number of entries in the Sequence Dictionary

$LN(2)$ = the natural logarithm of 2

As an example, a Sequence Dictionary with 4096 ($4096 = 2^{12}$) entries will require an encoding Pointer Length of 12. For this reason, the relative contribution of Quantum Symbols (when encoded as

Exceptions) to final MDF Miniaturization Ratio is usually negative ($MR_{QS(1)} < 1$).

DMT Terminal Sequence Miniaturization

If a Terminal Sequence of length two (2) is used to encode a sequence within an MDF file (e.g. a two letter word in a word document), then its **relative** contribution to Miniaturization is given by:

$$MR_{QS(2)} = \frac{2 \times BL_{ID}}{PL_{MDF}}$$

Terminal Sequences of length two (2) will provide **positive relative** contributions to miniaturization ($MR_{TS(2)} > 1$) if the MDF pointer length is less than twice the input data bit length. In the case of text documents, this will occur if the Sequence Dictionary has 32,768 entries or less.

A Terminal Sequence of length n will have a **relative** contribution to MR as follows:

$$MR_{TS(n)} = \frac{n \times BL_{ID}}{PL_{MDF}}$$

The **relative** contribution of a Terminal Sequence of length n, to the MDF miniaturization Ratio is given by:

$$MR_{TS(n)} = \frac{\text{Count}(n)_{ID} \times n \times BL_{ID}}{\text{Count}(n)_{MDF} \times PL_{MDF}}$$

Where:

$\text{Count}(n)_{ID}$ = the number of sequences in the input data that are represented by the Terminal Sequences of length n

$\text{Count}(n)_{MDF}$ = the number of Terminal Sequences of length n in the MDF data

The **total** contribution to MR, by all encoded Terminal Sequences, ranging in length from one to $TS(\text{Max})$, is given by:

$$MR = \frac{\sum_{n=1}^{n=TS(\text{Max})} \text{Count}(n)_{ID} \times n \times BL_{ID}}{\sum_{n=1}^{n=TS(\text{Max})} \text{Count}(n)_{MDF} \times PL_{MDF}}$$

In the case of a word document with a dominant word length of 5, and a Sequence Dictionary of 4096 entries (and assuming no Quantum Exceptions), the MR can be estimated as follows:

$$MR = \frac{5}{1} \times \frac{8}{12} = 3.33$$

Sequence Dictionary Pruning and Finalization

All Quantum Symbols are included in the Sequence Dictionary. For text-based data this generally requires the first 256 entries.

DMT uses statistical methods to limit the number of Quantum Sequences in the Sequence Dictionary, based on the Frequency of occurrence of these sequences in the input data.

To limit memory and CPU usage during dictionary discovery and pruning, frequency thresholds for each sequence length (n) are calculated based on very limited samples of frequency data collected for sequence lengths n or below.

For instance, Quantum Sequences of length two are included if their frequency exceeds the frequency threshold for calculated for QS(2). This frequency threshold is initially set as the Probability of the mean rank of Quantum Sequences of length two (2).

Mean rank for length n is determines as follows:

$$Mean R(n) = \sum_{i=1}^{i=QS(n)_{MAX}} Freq[QS(n)_i] \times R_i$$

Where:

Mean R(n) = The mean rank for all observed Quantum Sequences of length n

R_i = The relative rank of each Quantum Sequence of length n, after each Quantum Sequence has been rank ordered in terms of descending frequency.

During the initial stages of Sequence Dictionary discovery, the probability of Quantum Sequences of length greater than two is estimated from the two Quantum Sequences that comprise the new Quantum Sequence.

Once a representative sample of the input data has been obtained, then actual frequencies are used.

Probability for a specific Quantum Sequence of length n may be estimated from its rank relative to other sequences of similar length (R), using Zipfs Law:

$$P(QS_i) = \frac{1}{C} \times R^{-\zeta}$$

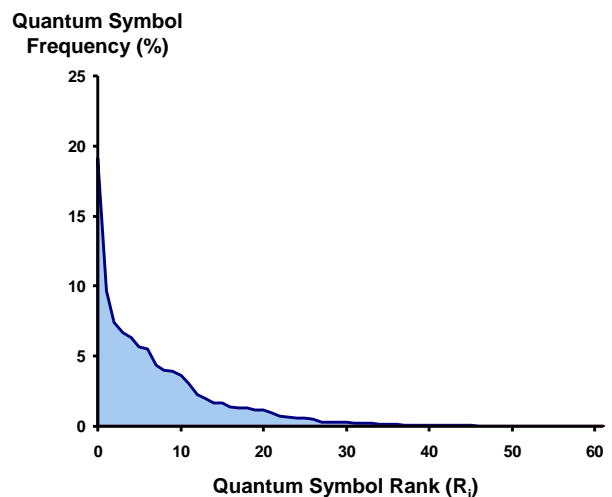
If the Rank of a sequence is known from a small sample calculated then Zipfs Laws suggests the relationship between probability and rank:

For English text, $\zeta \approx 1$. Other studies by Czirok et al illustrate the application of this law to binary datasets.

DMT Frequency Profile – Input Data

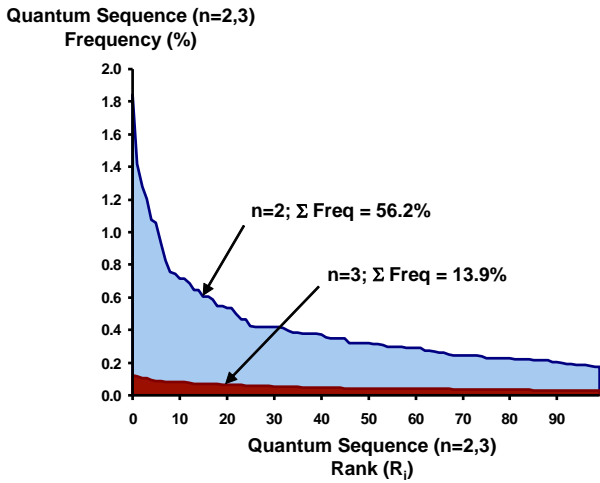
A sample 4MB text file is analyzed here for demonstration purposes. Figure 3. below records the actual frequencies of Quantum Symbols observed in the input sample text data. Note: the sum of all Quantum Symbol frequencies is equal to 100%.

**Figure 3. Input Data
Quantum Symbol Frequency (n=1)
(4MB Text File)**



In this example, 62 distinct Quantum Symbols were discovered. The most frequent five symbols (space, e, t, h and a) comprised 49% by frequency of the input text data. The most frequent 20 symbols comprised 92% of the input data.

**Figure 4. Input Data
Quantum Sequence Frequency (n = 2, 3)**



419 distinct Quantum Sequences of length two (n=2, or Quantum Pairs) were discovered in the input data, representing 56% of all Quantum Sequences (of length two or greater) discovered. The most frequent five Quantum Pairs accounted for 12% of all Quantum Pairs; the most frequent 20 Quantum Pairs comprised 31%.

1,290 Quantum Sequences of length three (n=3) were discovered in the input, representing 14% of all discovered sequences (of length two or greater). The most frequent five sequences comprised 3.9% of all Quantum Sequences of length 3; the most frequent 20 comprised 12.1%.

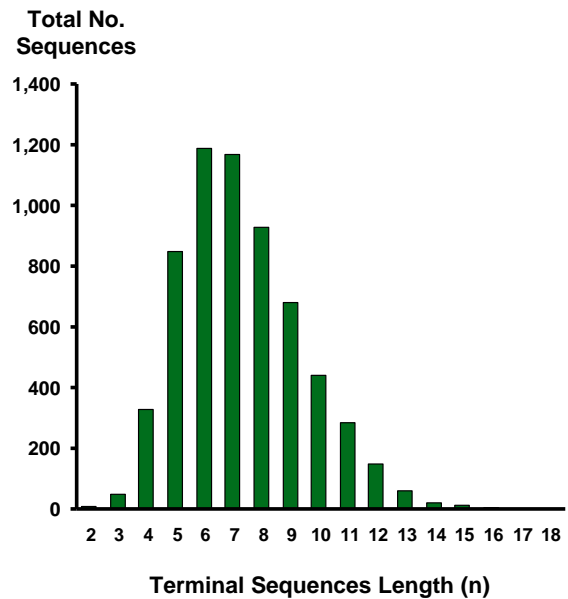
In all, 12,269 sequences of sequence length two to 18 were discovered in the input data.

DMT Frequency Profile – Optimized Dictionary

50% of these discovered sequences were rejected during the Sequence Dictionary optimization and finalization process, based on their frequency and overall contribution to miniaturization. Following Dictionary finalization, 6,177 Terminal Sequences, with sequence length 2 to 18, were used to encode the sample text.

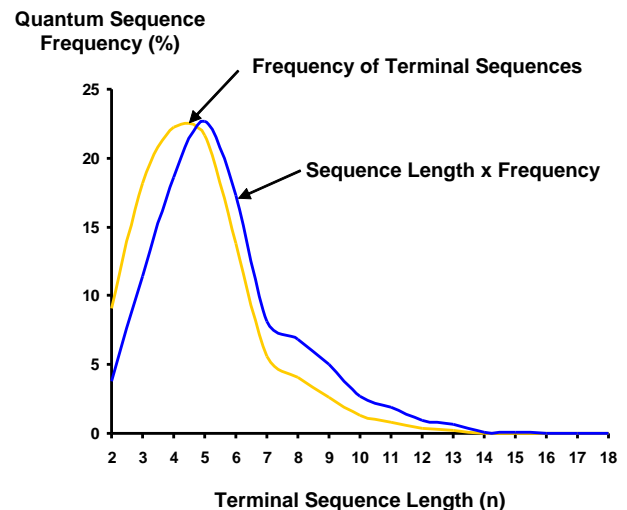
As illustrated in Figure 5, 20% of these Terminal Sequences comprised five or less Quantum Symbols, and 53% comprised six to eight Quantum Symbols.

**Figure 5. Finalized Dictionary
Discrete Terminal Sequences (n = 2 to 18)**



As illustrated in Figure 6, the distribution of Terminal Sequence frequency in the input data closely approximates the distribution of memory usage (frequency x length).

**Figure 6. Terminal Sequence Frequency
(n=2 to 18)**



Miniaturization Performance

Internal Dictionary Performance by File Type

The key drivers of miniaturization performance using Internal (File-Specific) Sequence Dictionaries are:

- **File Type.** Application data that is highly structured (e.g. text, XML, HTML, BMP, Database files) leads to the highest MR and FSR performance using DMT. Similarly, already compressed file formats (e.g. PNG, GIF) exhibit the lowest MR performance.

- **File Size.** For each file format, larger input data sets contain larger proportions of repeating sequences, and therefore exhibit the highest MR performance.

Specific situations where DMT has performance limitations include executable files, compressed format files (e.g. single GIF images), some binary files, and audio/video files (e.g. lossy applications).

MR performance figures are quoted in Tables 1 and 2 below for Internal (File-Specific) Sequence Dictionaries.

**Table 1. FSR and MR Performance
Text-Base File Formats**

Data Type	No. Files Optimized	Average Input Data Size (Mbytes)	Average File Size Reduction (%) (with Dict.)	MR (with Dict.)
Text	16	2.2 Mb	62.6%	2.68
HTM	5	329 Kb	80.9%	5.25
HTML				
XML	526	65 Kb	79%	4.75

As illustrated in Table 1 above, DMT typically achieves File Size Reductions of between 60% and 80% for text based file formats.

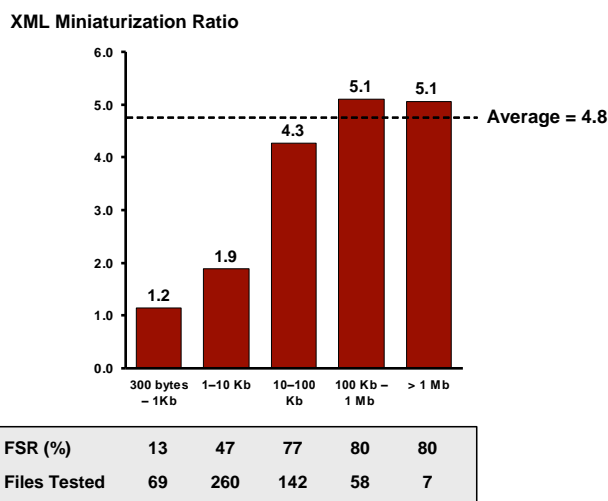
**Table 2. FSR and MR Performance
Mapping Data Formats**

Data Type	No. Files Optimized	Average Input Data Size (Mbytes)	Average File Size Reduction (%) (with Dict.)	MR (with Dict.)
BMP				
BMP (Ant-Alias)				
Geo-TIF				
ESRI				
Telcontar RMF				
TeleAtlas				

Impact of File Size on MR Performance

As highlighted in Figure 7, file size has a dramatic affect on miniaturization performance for XML files.

**Figure 7. XML MR Performance
Impact of File Size**



External Sequence Dictionary Performance

DMT's ability to abstract and deploy a Common dictionary for a large set of common format files leads in some cases to dramatic performance improvements over existing compression techniques. For example, DMT has been demonstrated to reduce GIF file size by 90% using a common (detached) dictionary.

DMT Products

A number of WindSpring products have been developed based on the DMT architecture.

WindSpring's OEM product, the WindSpring Application Codec, has been optimized for Mobile, Wireless and Digital Mapping applications.

The CODEC includes the following components:

- Encoder, decoder and analysis modules
- Development Graphical User Interface (GUI)
- Sample interface access programs and technical documentation
- APIs
- Sequence Dictionary optimization toolset

The Sequence Dictionary optimization toolset enables optimum dictionary parameters to be

discovered for particular application data characteristics.

The CODEC is available for the following server and client operating systems:

The CODEC is available for the following server and client operating systems:

- Windows XP
- Pocket PC 5.0, Pocket PC 2003, (Windows CE 4.2)
- Linux
- JAVA and J2ME
- uITRON (custom installation with NRE)

Integration Steps

Integration of the WindSpring Application Codec may be achieved by file interface (via File Manager), stream interface (via Shell), or via DMT APIs

The key steps required for integration are as follows:

- ◆ Analysis of the application data set using the DMT analysis software, so as to develop an application-specific Sequence Dictionary.
- ◆ Optimization of the Sequence Dictionary for specific application modes, file size reduction requirements, and device storage space constraints.
- ◆ API integration of the existing application with the DMT analysis and encode software
- ◆ Using the developed CODEC Sequence Dictionary, conversion of the existing data set to an editable Miniaturized Index File which is a fraction of the size of the data set.
- ◆ If required, API integration of the mobile device OS with CODEC decode software;
- ◆ Pre-loading of the mobile device, either at the point of manufacture or via the wireless network, with the CODEC map dictionary and CODEC decoder (approximately 6 Kbytes)
- ◆ The service provider then transmits all application data to mobile users as editable Miniaturized Index Files.

Further Questions

The effect of DMT on the Mobile, Digital Mapping, Search and High-Speed Storage industries applications is significant. By reducing data latency and network cost, DMT promises to significantly

improve subscriber adoption, usage and price, and materially increase revenue potential for a large range of applications and services.

If you would like to obtain further information on DMT, the MDF data format, or the application of the WindSpring Application Codec for your existing services or products, or if you would like to evaluate the WindSpring Application Codec, please contact WindSpring, Inc. at

business_development@WindSpring.com.