Driving the future of relevant, real-time navigation content and services

WindSpring Technical White Paper:

Data Miniaturization Technology (DMT) Miniaturization and Acceleration of Next Generation Mobile and Storage Applications

WindSpring, Inc.
www.windspring.com

1735 North First Street
Suite 102
San Jose, CA 95112

Tel: (408) 452-7400
Fax: (408) 452-7444

Table of Contents

Objectives

WindSpring's Data Miniaturization Technology (DMT™) provides a solution for 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 as follows.

- 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 Mobile, Digital Mapping, High-Speed Storage and Search applications.

These challenges include:

- **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. Mobile networks)

- **Data Storage Constraint**. These applications often require data access and manipulation in storage-constrained environments (e.g. Mobile devices and High-Speed Storage RAM)

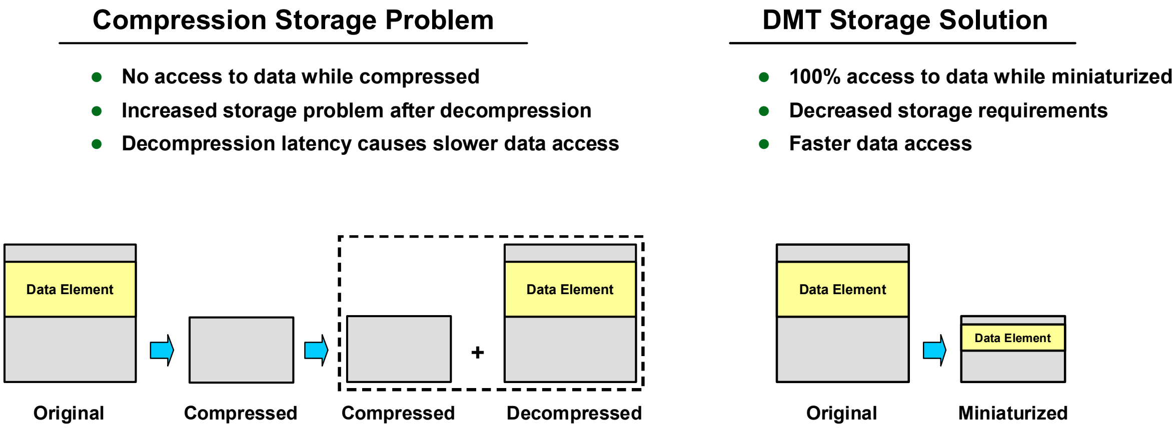
- **Network and Hardware Cost**. Growing network and hardware cost can prohibit the deployment and operation of next generation, intelligent applications

DMT provides a flexible, pervasive 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:

- **Storage**. DMT enables storage-constrained devices and networks to store dramatically increased volumes of data, without the need to increase storage or network capacity.
- **File Transfer**. DMT enables throughput-constrained applications to access and transmit data at significantly higher speeds than otherwise available with conventional storage or mobile solutions.
- **Incremental Updates**. DMT through MDF supports edit in place and update in place for maintaining without replacing.

Figure 1.0 Key Advantages of DMT Over Compression


Compression Storage Problem

- No access to data while compressed
- Increased storage problem after decompression
- Decompression latency causes slower data access

DMT Storage Solution

- 100% access to data while miniaturized
- Decreased storage requirements
- Faster data access

As illustrated in Figure 1.0 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 directly 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.

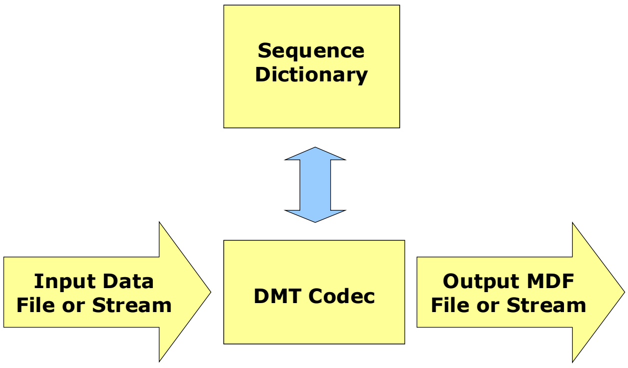
DMT is uniquely suited for space-constrained environments such as cell phones, SmartPhones, PDA's, DVD's and in-car navigation systems. DMT also increases data transfer and data access speeds regardless of the available storage space, constrained or unconstrained.

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, Malaysia 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 heretofore been restricted due to space and speed constraints.

Figure 2.0 DMT Data Flow



As highlighted in Figure 2.0, the key steps in the DMT data miniaturization process are as follows:

1. **Sequence Dictionary Discovery**. The input data set (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 and adapted over time, based on the changing nature of the application data; or
   - Selected or adapted from an existing DMT Sequence Dictionary.

2. **Encoding**. Input data is then transformed by the DMT Codec to create a Mobile 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 and optimized for a wide range of data types typically encountered in Mobile, Wireless, Digital Mapping and High-Speed Storage applications, including text, binary, database and vector map formats.

DMT Dictionary Deployment Options

DMT offers two dictionary deployment options:

- **Internal (File-Specific) Dictionaries**. This is the more common dictionary deployment mode, and the easiest to implement. In this mode, DMT dictionaries are usually developed and deployed with individual MDF files.

  Applications where internal (file-specific) dictionaries are appropriate include both on-board and off-board mobile GPS mapping, internal mobile phone applications, remote computing, enterprise web-based mapping and in-car navigation systems.

- **External (Common) dictionaries**. One of the unique features of DMT is its ability to develop a dictionary that is common to a set of files of similar context (e.g. multiple Word files), 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 external dictionaries are appropriate include mobile gaming, 3D gaming, consumer web-based mapping services and document management systems.

In addition to the flexibility to abstract 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 keys to DMT's unique capability are its innovative and patented Miniaturized Data Format, and Multi-Index Pointer and Exception Handling processes.

These processes include:

- **Quantum Symbol, Quantum Sequence and Terminal Sequence Pointer Identification**. These processes enable:
  - 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.
  - Automatic identification and compilation of the repeated sequences of these symbols, regardless of the data context, as a Quantum Sequence or Terminal Sequence Pointer.

  - The collection of frequency data to enable ongoing Sequence Dictionary learning.

  - High-speed seek, search, edit and render of MDF format data for a single file.

- **Fixed Bit-Length Encoding**: The use of fixed bit-length encoding leads to the most efficient (i.e. smallest) Sequence Dictionary for a given discovered Quantum Symbol and Terminal Sequence Pointer set.

- **Multi-Index Pointer System (MIPS)**. MIPS enables:
  - 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, edit and render individual data sets.

  - The operation and integration of ***multiple Sequence Dictionaries*** across multiple files or streams.

- **Exception Handling**. Exception handling enables:
    - 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 search within MDF files.

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.

- **Quantum Slide (QS)**. The Quantum Slide Codec uses a technique that is suited to files with short to medium length repeating sequences.

QP Codec Operation

QP Codec operation is as follows:

- **QP Sequence Dictionary Creation**. 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 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 (Mbytes). This is the amount of memory reserved for Sequence Dictionary analysis.

    - Block Size (bytes). The original data is encoded in segments equal to the Block Size.

    - Terminal Sequence Dictionary Size. This is the maximum number of Terminal Sequence records allowable in the Sequence Dictionary. This controls the size of the dictionary and the consequent percentage File Size Reduction (FSR) achieved following encoding.

    - Maximum Sequence Length (bytes). This is the longest sequence in bytes that can be discovered during the Sequence Dictionary discovery process.

  In the Learning Dictionary mode, 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. In Fixed Dictionary mode, this data is not tracked.

- **QP Encoding**. The encoding process commences with a Quantum Symbol and scans the Sequence Dictionary for the longest length Terminal Sequence that matches the first Quantum Symbol and symbols that follow it. At this point the Terminal Sequence pointer is written to the MDF output data. 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 rest of the Quantum Sequences. 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 MDF output data. The process continues unit 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 Creation**. 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.

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

  - <u>Window Size (bytes)</u>. The original data is analyzed in segments equal to the Window Size during the Sequence Dictionary creation process.

  - <u>Maximum Sequence Length (bytes)</u>. This is the longest sequence that can be discovered during the Sequence Dictionary discovery process.

  - <u>Mean Rank Adjustment</u>. 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 Rank Adjustment process.

  - <u>Seed Size (bytes)</u>. This is the maximum number analyzed at any time before the Sequence Dictionary is pruned using the Rank Adjustment process.

  - <u>Block Size (bytes)</u>. The original data is encoded in segments equal to the Block Size.

- **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 and symbols that follow it. 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:

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

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

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 (bytes)
Size (MDF) = Size of MDF format data (bytes)
Size (SD) = Size of Sequence Dictionary (bytes)

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 Sequence in the input data is calculated as:

$$Freq\,[\,QS(n)_i\,] = \frac{k\;x \sum [QS(n)_i\,]}{Size\;(ID)}$$

Where:

k = the number of bits in each Quantum Sequence (usually n 8 bits).*
n = the length of the Quantum Sequence in terms of Quantum Symbols.
QS(n) = The Quantum Sequence of length n
Size (ID) = Original Data Size (bits).

In the special case of Quantum Symbols:

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

Where:

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

As an example, most text-based documents would contain 256 Quantum Symbols that are mapped to the 256 ASCII characters.

DMT Quantum Symbol Miniaturization

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

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

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 Terminal Sequence Dictionary, as follows:

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

Where:

LN (SD Entries) = the Natural Logarithm of the number of entries in the Terminal Sequence Dictionary.

LN(2) = the natural logarithm of 2.

As an example, a Terminal 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 (i.e. $MR_{QS\ (1)} < 1$).

DMT Terminal Sequence Miniaturization

If a Terminal Sequence pointing to a Quantum 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:

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

Terminal Sequences representing Quantum 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 Terminal Sequence Dictionary has 32,768 entries or less.

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

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

This may be re-expressed as:

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

Where:

> $Count(n)_{ID}$ *= the number of sequences in the input data that are represented by the Terminal Sequences of length n.*
> $Count(n)_{MDF}$ *= the number of Terminal Sequences of length n in the MDF data.*
> *TS(n) = Terminal Sequence representing Quantum Sequence of length n.*

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

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

In the case of a text document (Quantum Symbols = ASCII characters of 8 bits) with a dominant word length of 5, and a Terminal Sequence Dictionary of 4096 (12 bits) 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 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.

Mean rank for length n is determines as follows:

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

Where:

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

Ri = The 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 and Mean Rank of Quantum Sequences of length one and two is calculated from the input data. 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 for Quantum Sequences of length greater than two.

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:

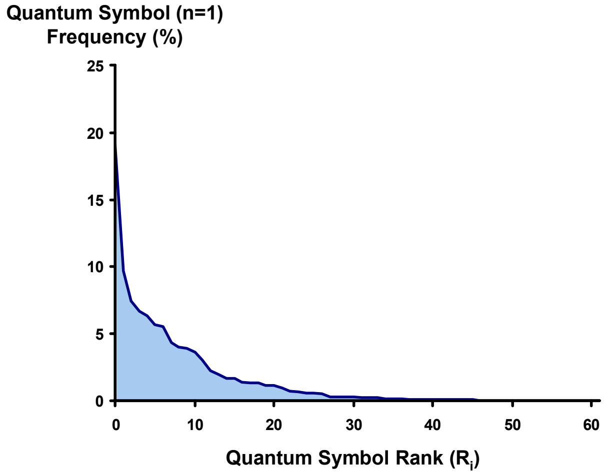
$$\boxed{P(QS_i) = \frac{1}{c} \times R^{-\zeta}}$$

If the rank of a sequence is known from a small sample, then Zipfs Laws suggests the relationship between probability and rank.   For English text, $\varsigma \approx 1$. 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 3.0 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%.

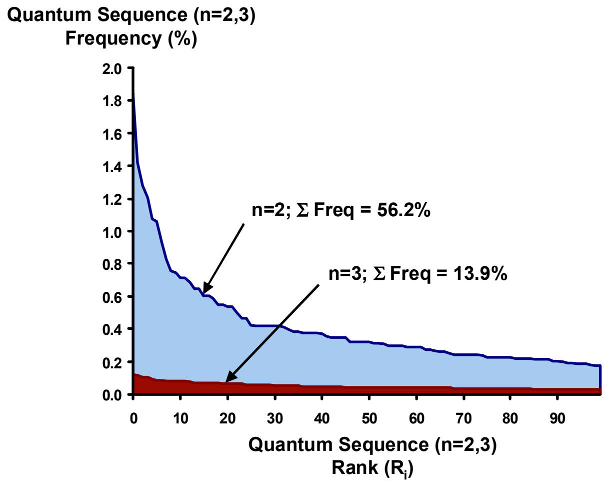
Figure 3.0 Input Data
Quantum Symbol Frequency (n=1)
(4 Mb Text File)



In this example, 62 discrete 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 4.0 (following) describes the frequency profile of Quantum Sequences for length two and three.

Figure 4.0 Input Data
Quantum Sequence Frequency (n = 2, 3)
(4 Mb Text File)



In this same example, 419 discrete Quantum Sequences of length two (n=2, or Quantum Pairs) were discovered in the input data, representing 56% of all Quantum Sequences discovered (of length two or greater). 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%.

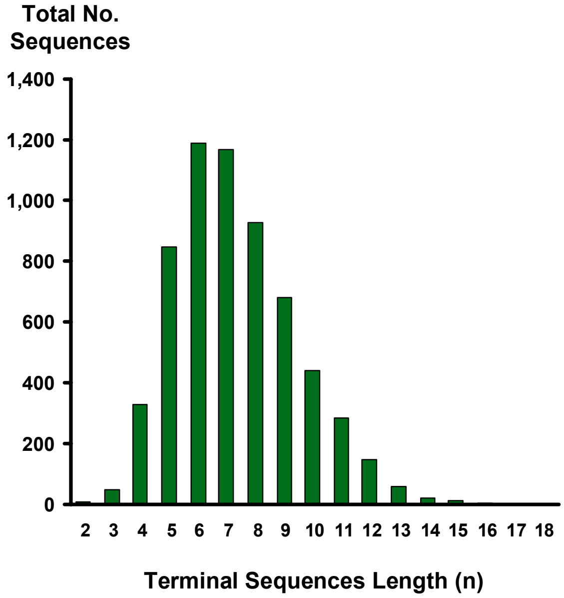
In all, 12,269 sequences of sequence length two to 18 were discovered in the input data.

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, representing Quantum Sequences of lengths 2 to 18, were used to encode the sample text.

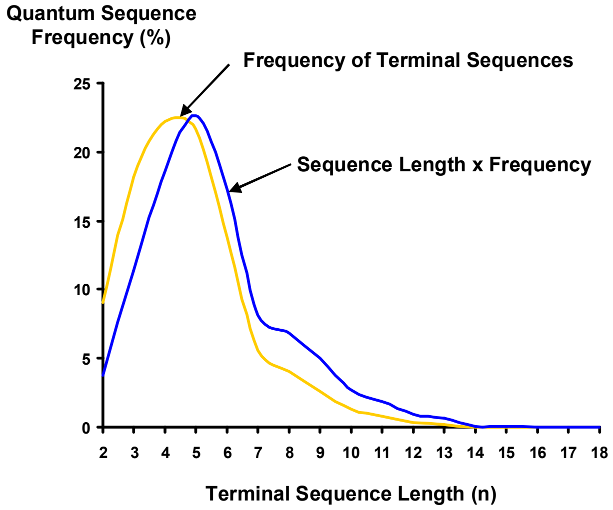
As illustrated in Figure 5.0, 20% of these Terminal Sequences comprised five or less Quantum Symbols, and 53% comprised six to eight Quantum Symbols.

Figure 5.0 Initialized Dictionary
Discrete Terminal Sequences (n = 2 to 18)
(4 Mb Text File)



As illustrated in Figure 6.0, the distribution of input data usage (frequency x length) closely approximates the distribution of Terminal Sequence frequency.

Figure 6.0 Terminal Sequence Frequency (n=2 to 18)
(4 Mb Text File)



Miniaturization Performance

Internal Dictionary Performance by File Type

The key drivers of miniaturization performance using Internal 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. In contrast, 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.

MR performance figures are quoted in Table 1.0 for Internal Sequence Dictionaries.

Table 1.0 FSR and MR Performance
Text-Base and Mapping File Formats

Data Type	**No. Files Optimized**	**Average Input Data Size (Mb)**	**Average File Size Reduction (%)**	**MR**
Text-Based Data				
Text	16	2.2 Mb	62.6%	2.68
HTML	6	9.6 Kb	61.2%	2.58
XML	526	65 Kb	79.0%	4.75
HTM	5	329 Kb	80.9%	5.25
Map Data				
Kiwi-W	12	636 Mb	30.1%	1.43
Telcontar (RMF)	1	190 Mb	49.6%	1.98
ESRI (DBF, SHP, SHX, DAT, ID, IND, MAP, MBX, TAB, WOR, PDF)	301	282 Kb	72.4%	3.62
Geo-TIF	5	5.2 Mb	76.1%	4.18
BMP	27	110 Kb	84.1%	6.29
TeleAtlas (DBF, PRJ, SHP, SHX)	258	231 Kb	84.2%	6.33

As illustrated in Table 1.0 above, DMT typically achieves FSR’s of between 60% and 80% for text-based file formats, and between 30% and 85% reduction for mapping data.

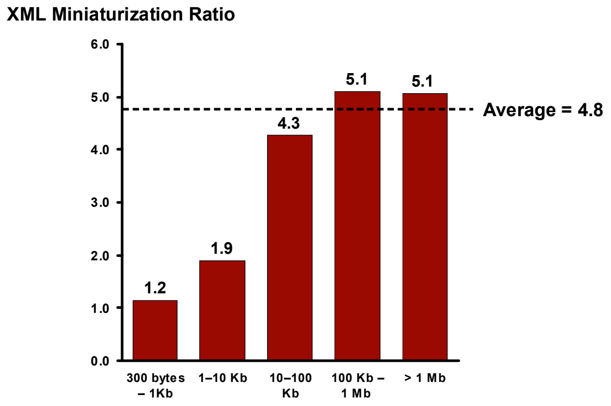
DMT Limitations

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).

Impact of File Size on MR Performance

As highlighted in Figure 7.0, file size has a dramatic effect on MR for XML files.

Figure 7.0 XML MR Performance
Impact of File Size
(Internal Dictionary)



FSR (%)	**13**	**47**	**77**	**80**	**80**
Files Tested	**69**	**260**	**142**	**58**	**7**

External Sequence Dictionary Performance

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

DMT Products

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

WindSpring SDK includes DMT encoder, decoder and analysis modules; a development Graphical User Interface (GUI); sample interface access programs; APIs; Sequence Dictionary optimization toolset; and technical documentation.

WindSpring SDK is available for Windows 2000 and Windows XP, Linux, BREW and Pocket PC. A J2ME version will be available in 4Q07.

Integration Steps

The key steps required for integration are:

- **Analysis.** One-off analysis of the data set using the ***Sequence Dictionary optimization toolset*** to develop a set of standard Sequence Dictionary parameters for use during dictionary creation. Further optimization of these parameters for any specific application modes, FSR requirements, and device storage space constraints.

- **Integration**. Integration may be achieved via file interface (via File Manager), stream interface (via Shell), or DMT APIs.
  - Server-side integration of the DMT Encoder.
  - Client-side integration of the DMT Decoder. Client side integration of the DMT Encoder is also available if edit functionality is required at the client.

- **Deployment**. Once integrated and deployed, the application transmits and uses WindSpring's editable MDF format as the primary application data. Application data may be pre-encoded or encoded on the fly, as data is requested.

Summary

The potential impact of DMT on the Mobile, Digital Mapping, Search and High-Speed Storage industries is significant. By reducing data latency and network cost, DMT can significantly improve subscriber adoption, usage and retention, and materially increase revenue for a large range of applications and services.

WindSpring, Inc.
www.windspring.com

1735 North First Street
Suite 102
San Jose, CA 95112

Tel: (408) 452-7400
Fax: (408) 452-7444