EASEL: Easy Automatic Segmentation Event Labeler

Isaac Wang[1], Pradyumna Narayana[2], Jesse Smith[1],
Bruce Draper[2], Ross Beveridge[2], Jaime Ruiz[1]

[1]Department of CISE
Gainesville, Florida, USA
{wangi, jd.smith, jaime.ruiz}@ufl.edu

[2]Department of Computer Science
Fort Collins, Colorado, USA
{prady, draper, ross}@cs.colostate.edu

ABSTRACT
Video annotation is a vital part of research examining gestural and multimodal interaction as well as computer vision, machine learning, and interface design. However, annotation is a difficult, time-consuming task that requires high cognitive effort. Existing tools for labeling and annotation still require users to manually label most of the data, limiting the tools' helpfulness. In this paper, we present the Easy Automatic Segmentation Event Labeler (EASEL), a tool supporting gesture analysis. EASEL streamlines the annotation process by introducing assisted annotation, using automatic gesture segmentation and recognition to automatically annotate gestures. To evaluate the efficacy of assisted annotation, we conducted a user study with 24 participants and found that assisted annotation decreased the time needed to annotate videos with no difference in accuracy compared with manual annotation. The results of our study demonstrate the benefit of adding computational intelligence to video and audio annotation tasks.

Author Keywords
Data annotation tools; gesture segmentation; gesture analysis

INTRODUCTION
Gestural research—research focused on the use of gestures as input or channel of communication—has resulted in significant breakthroughs in user interaction (e.g., the Microsoft Kinect, mobile phones, surface gestures). Furthermore, gestural research has broad applications to research focused on multimodal communication, computer vision, machine learning, and interface design. Much of this research relies on video annotation, the process of identifying significant gestures and events in recorded video and then assigning them a shared label. For instance, this approach is used for studying multimodal interaction [10] and is also required for creating ground truth labels in large video datasets, such as the ChaLearn Gesture Dataset 2011 [6]. Often, annotation is done using simple media or text editors [12]. However, this is a difficult and time-consuming task; for example, a 5-minute video could take an hour for a single researcher to properly annotate.

IUI'18, March 7–11, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-4945-1/18/03…$15.00
https://doi.org/10.1145/3172944.3173003

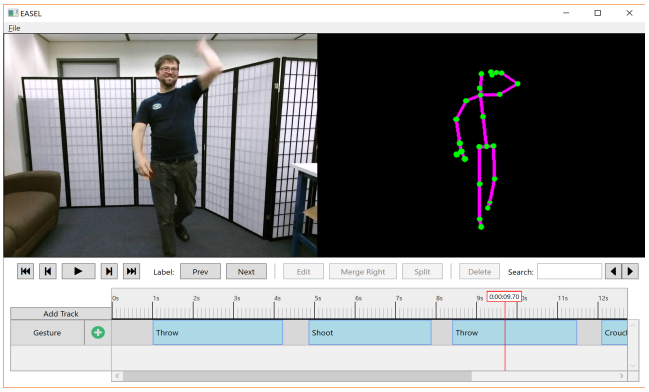

Figure 1. EASEL's user interface, showing the video player (top) and the annotation timeline (bottom).

Given the critical role of video annotation, researchers have developed several specialized tools for performing such analyses. These include programs designed to facilitate the annotation of audio and video for motion-tracking [9], multimodal discourse [8,12,16,18], and semantic description [2,4]. For instance, the ELAN annotation tool [18] uses a hierarchical system of annotations to transcribe and analyze speech components in video and audio files for multimodal studies. Similarly, ANVIL [8] and EXMARaLDA [16] are also used for annotation and transcription but go a step further by including the ability to define custom annotation schemes and built-in analysis functions to parse and visualize the annotated data. However, due to limited and varying feature sets (e.g., inability to handle common video formats, or insufficient support for a specific method of analysis), these annotation systems are often inappropriate for use in research areas for which they were not designed [12].

Additionally, although these tools provide interfaces that help facilitate annotation, they still require users to manually annotate most of the data, limiting their helpfulness. Few tools introduce methods to automate the process. For instance, Aydınlılar and Yazıcı [2] present a tool that helps automatically label semantic content in video. Likewise, the SorTable tool [15] helps automate continuous labeling, but by simplifying the process to better leverage both human and machine capabilities. In the same vein, we saw a potential for including assisted annotation for gesture labeling, as annotating videos can still be a tedious endeavor even when utilizing specially designed tools. Thus, we developed the Easy Automatic Segmentation Event Labeler (EASEL) from the ground up as a flexible but powerful annotation tool,

supporting multiple video formats, requiring little configuration to get started, and featuring assisted annotation to aid in the gesture annotation process.

In this paper, we present EASEL, a tool supporting gesture analysis with Microsoft Kinect recordings that uses automatic gesture segmentation and recognition to streamline the annotation process, a feature we call *assisted annotation*. We describe the design of EASEL and how assisted annotation can benefit users when annotating videos. We also describe the results of a user study evaluating the efficacy of assisted annotation, showing how it can help reduce the time needed to annotate videos by offloading some of the work to the system.

SYSTEM OVERVIEW

EASEL is a tool designed for the annotation of multimodal communication and full-body motion gesture data. The main emphasis of EASEL is the inclusion of assisted annotation, or the use of machine-generated annotations to reduce the user's workload. The goal is for the system to intelligently assist the user by recognizing and labeling gestures automatically, thus changing the task from annotation to verification. The goal is not only to speed up annotation, but also reduce the amount of effort needed to complete the task.

EASEL's UI has two main components: the video player and the annotation timeline (Figure 1). These were designed to function similar to video editing software and other annotation tools [8,16,18], offering a level of familiarity for users. The video player was designed with the goal of allowing simultaneous playback of multiple videos. This allows for playback of videos from different angles or multiple parties, and can also display a video side-by-side with the corresponding Microsoft Kinect skeleton. Additionally, a requirement of the video player was supporting a variety of modern video formats; most existing annotation tools rely on the Java Media Framework [19], which lacks modern codec support (e.g., MPEG-4). EASEL uses the Microsoft Media Foundation framework [20] and supports modern video formats such as MPEG-4 and H.264.

The annotation timeline was designed to visually represent annotations using *musical score*, or *piano roll,* notation, which is particularly suited for depicting multimodal, multi-level, and multi-party communication [16] because it supports the combination of annotation and visualization [13]. The timeline displays annotations on individual tracks, allowing multiple attributes (e.g., gesture, intent, speech) to be labeled for the same video. The start/end times of each annotation can be dragged, and the annotation's label can be edited by double-clicking on the annotation and typing in a label. The timeline was also designed with the goal of not requiring any extensive setup before starting annotation; instead, EASEL allows users to define new tracks and labels as needed, adding a degree of simplicity and flexibility.

Annotating Videos with EASEL

We describe how a user would annotate a Kinect recording with EASEL. Although EASEL can open a variety of video formats for general use, EASEL currently supports XEF recordings from Kinect Studio [21] for assisted annotation, but could conceivably handle similar formats as well. First, the Kinect XEF file is extracted into RGB video and skeleton data (describing body joint positions for each frame in the video). The video is directly loaded by EASEL for annotation and shows up in the video player. The skeleton data is loaded by the assisted annotation module, which segments and labels gestures in the skeleton data stream. This is shown in the annotation timeline, prepopulating the timeline with annotations (Figure 2). The first time the user loads a video (before any gestures are defined), EASEL will still auto-segment the data into annotations marking when it thinks a gesture occurred, labeling each annotation as "Unknown." The user then goes through the annotations, manually refining the start/end times or correcting the label if needed. These gesture annotations are saved to a database, and, the next time a video is loaded, EASEL will try to auto-label the data by recognizing the previously labeled gestures, effectively getting better as the user annotates more gestures.

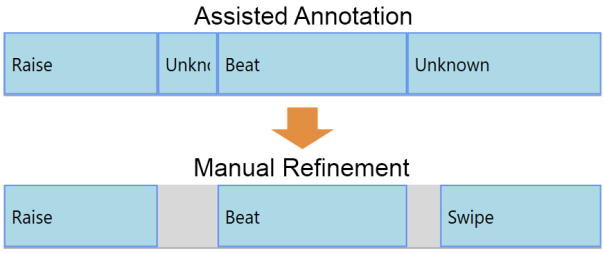

Figure 2. EASEL supports assisted annotation by segmenting and recognizing gestures to prepopulate the timeline. The user then manually refines annotations as needed.

IMPLEMENTATION

EASEL is written primarily in C#, with portions written in C++ to interface with Microsoft's native Media Foundation libraries. Other portions of EASEL, such as our implementation of Dynamic Time Warping (DTW) for gesture recognition, are also written in C++ for increased performance. EASEL uses Windows Presentation Foundation (WPF) as its GUI framework and supports systems running the .NET 4.5 runtime (Windows 7 or above). This operating system requirement was selected to take advantage of the Media Foundation library, which provides native video and audio playback for a large number of file formats and codecs and is the most up-to-date media playback framework currently supported by Microsoft.

Assisted Annotation

To streamline the annotation process, EASEL implements assisted annotation, reducing the need to manually identify gestures. Our approach consists of two parts: gesture segmentation, which first segments the video into potential gestures, and gesture recognition, which tries to label each potential gesture by recognizing previously labeled gestures.

Gesture Segmentation

Segmentation is one of the most time-consuming tasks when annotating video, audio, or motion capture data, and thus is an excellent candidate for improvement by automatic

methods. EASEL can automatically segment a video into discrete gestures by estimating curvatures in high dimensional space using the ACE-PC technique by Arn et al. [1]. This method was chosen because of its ability to handle "open-world" gesture segmentation, i.e. identifying motions as potential gestures without any prior training. This method operates on a stream of multidimensional motion data over time and is not specific to the Kinect, making it possible to work with other types of input (e.g., accelerometer data for mobile interaction). After the motion data is loaded and segmented, these potential gestures appear as candidate annotations in the timeline.

Gesture Recognition

The output from gesture segmentation is further enhanced by attempting to automatically determine a label for each candidate annotation created. This is done by extracting the Kinect skeleton data for a given annotation and running a gesture matching or recognition technique on it. The annotation is then given a label based on a detected match.

As a simple baseline technique, we implemented a version of dynamic time warping (DTW) [14] for gesture recognition. As a template-based approach, DTW is particularly suited for EASEL, as it does not require training and can automatically recognize new gestures as they are added to the gesture database. For our implementation, the motion from the skeleton data is normalized by subtracting the spine base from every joint. Once a few instances of gestures have been labeled and stored in the database, EASEL will try to match new annotations against these gestures. A threshold is used to ensure that suggested labels are close enough matches (calibrated by finding an optimal threshold against the G3D [3] and EGGNOG [17] datasets, as they represent the typical use case for EASEL). If the similarity score of the closest matched gesture is below the threshold, then the annotation is pre-filled with the matched label. Otherwise, the annotation is classified as "Unknown," leaving the user to label it manually.

EVALUATION

We conducted a within-subject user study to evaluate the efficacy of assisted annotation in gesture annotation systems. We developed two versions of the system: *Assisted*, which included assisted annotation, and *Manual*, which did not. Our goal was to determine if:

(G1) Using *Assisted* results in decreased time to annotate videos than *Manual*.
(G2) The accuracy of annotations is higher when using *Assisted* as opposed to *Manual*.
(G3) *Assisted* decreases the task load involved with video annotation.

Participants

We recruited 24 participants (16 female, 8 male) from a local university through word-of-mouth and inter-departmental emails. Participants were between the ages of 19 and 29 (mean = 22.5, SD = 3.2). Seven participants had prior experience with similar video annotation software. Each participant was compensated with $20 in Amazon gift cards. Our study was approved by our Institutional Review Board.

Procedure

Participants were asked to complete two one-hour sessions on separate days (no more than three days apart). At the start of each session, participants were given a brief tutorial to the annotation task and given a reference sheet describing the gestures used in the videos. For each task, participants were asked to watch a video of a person performing gestures and use EASEL to annotate when and what gestures were performed, with an emphasis on ensuring that the entire gesture was annotated. Each session consisted of 10 trials: participants annotated a set of 10 videos by a single actor in the first session, and a set of 10 videos with a different actor in the second session. Participants used both versions of the system (*Assisted* and *Manual*), one for each session. The order of exposure to the two systems was counterbalanced.

We measured the time it took for participants to annotate each video. Annotations were also saved for later comparison against ground truth. At the end of each session, we asked participants to complete a short questionnaire to describe their experience with the system. The questionnaire included the questions from the NASA-TLX survey [7] for measuring task load, asking participants to rate their perceived workload across six continuous scales.

Videos Used in the Experiment

We created a small dataset of people performing gestures. The gestures in the dataset were based on the gaming gesture sets used in the MSRC-12 dataset by Fothergill et al. [5]. Each video contained a sequence of 10 gestures randomly selected from a gesture set containing six distinct gestures. There were two different sets of gestures: *Iconic* (crouch, shoot, throw, change, kick, goggles) and *Metaphoric* (raise, swipe, wind, bow, protest, beat). We recorded videos of four different actors, each performing gestures in 10 videos: five videos with *Iconic* gestures and five with *Metaphoric* gestures, resulting in 40 videos, each 30 seconds on average. The order of videos in each set was randomized, and the video sets used in the study sessions were counterbalanced.

To compare the accuracy of participants' annotations, we had three expert annotators independently create a set of annotations for all videos. Experts correctly labeled all gestures, giving us three sets of start/end times for each gesture. The start times and end times for each gesture were averaged to create a ground truth set of annotations.

Results

To compensate for learning effects, we treated the first half of the session as practice, using the latter five trials for analysis. The results from these trials were averaged to get the mean task time and annotation accuracy for each session.

Time Performance

We analyzed the time it took for participants to complete each task. Figure 3 shows boxplots comparing the distribution of task times for the two system conditions. The average task time was 166.9 seconds (SD = 50.5) when using

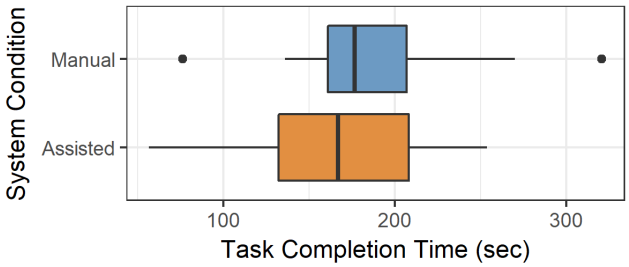

Figure 3. Boxplots comparing the average task completion time between the two system conditions. The *Assisted* system was faster ($p < 0.05$) than the *Manual* system.

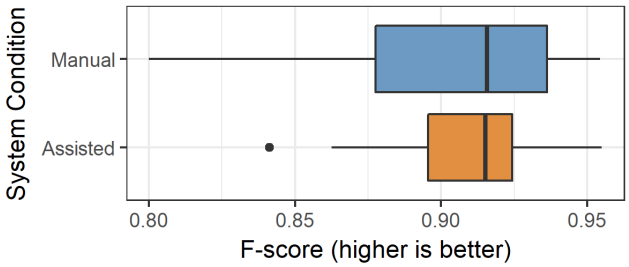

Figure 4. Boxplots comparing annotation accuracy (F-score) between the two system conditions. There was no significant difference in accuracy.

the *Assisted* system, compared with 188.2 seconds (SD = 50.7) for the *Manual* system. A within-subjects repeated measures ANOVA revealed a significant effect of system on task time ($F_{1,22} = 4.84$, $p < 0.05$), and no ordering effect of the system was observed ($p > 0.05$). Additionally, no interaction effects were observed ($p > 0.05$). We saw that, on average, participants using the *Assisted* version of EASEL completed tasks 11.3% faster than those using the *Manual* version. This indicates that assisted annotation is helping users accomplish tasks rather than being a hindrance.

Annotation Accuracy

We computed a frame-wise F-score [11] for each annotation, calculating precision and recall based on the number of frames matched against ground truth. F-scores were then averaged for each trial. The average F-score was 0.91 (SD = 0.03) for trials using the *Assisted* system and 0.90 (SD = 0.04) for trials using the *Manual* system (Figure 4). ANOVA revealed no significant difference in F-score between the two system conditions. The average F-scores for both conditions were similar to that of the expert annotators, who had an average F-score of 0.92. Thus, we found no difference in annotation accuracy, even when automatically labeled.

Task Load

We analyzed the results from the NASA-TLX survey to see if there was any reduction in task load when using the *Assisted* version of EASEL. We computed a raw TLX score averaging the values (between 0-100) from the six scales presented in the survey. The overall TLX score (lower is better) was 22.8 (SD = 16.5) for the *Assisted* condition and 27.9 (SD = 15.8) for the *Manual* condition. ANOVA revealed no significant difference in TLX scores between the two system conditions. When comparing scores individually across the six scales, we did not find any significant differences, although average scores for the *Assisted* condition were lower than *Manual* for each scale.

Discussion

Our evaluation study aimed to determine if including assisted annotation in gesture annotation tools improves task performance and decreases task load. From our results, we saw that task times in the *Assisted* condition were 11.3% faster on average (fulfilling goal G1), which quickly adds up when dealing with a large number of videos. One possible concern with the increase in speed is a decrease in quality; however, we found no significant difference in annotation accuracy between the two system conditions. Although we did not fulfill G2, we found that the accuracy of annotations for both conditions was comparable to that of experts. Thus, the inclusion of assisted annotation helped reduce annotation time while still retaining accuracy.

To determine how we could improve performance in future iterations, we took a closer look at the data to identify if there were any areas of EASEL that could be refined. Thus, we looked at how often users made changes to the machine-generated annotations. We found that gesture segmentation is not yet perfect, requiring adjustments to the start/end times for 72% of generated annotations. In contrast, only 32% of labels needed to be edited. Thus, improving automatic segmentation is a focus of future work, as it would decrease the need to manually edit annotations and further improve temporal performance.

CONCLUSION AND FUTURE WORK

In this paper, we presented EASEL, a video annotation tool that uses gesture segmentation and recognition to automatically label gestures in Kinect recordings. We showed how assisted annotation can improve temporal performance for gesture annotation with no difference in accuracy. The ability to have the system share the workload through assisted annotation has the potential to make video annotation tools more effective and intelligent. Our results not only affirm the benefits of assisted annotation for gesture annotation in EASEL, but also open up new possibilities for adapting the feature to work for other research domains. Future work will include incorporating automatic speech and affect recognition as well as expanding segmentation and recognition to easily support data formats other than Kinect recordings. The use of other segmentation and recognition techniques and ways to continually improve assisted annotation with live user feedback will also be explored.

ACKNOWLEDGEMENTS

This work was partially funded by the U.S. Defense Advanced Research Projects Agency and the U.S. Army Research Office under contract #W911NF-15-1-0459.

REFERENCES

1. Robert Arn, Pradyumna Narayana, Teegan Emerson, Bruce Draper, Michael Kirby, and Chris Peterson. Motion Segmentation via Generalized Curvatures. Under review in *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

2. Merve Aydınlılar and Adnan Yazıcı. 2013. Semi-Automatic Semantic Video Annotation Tool. In *Computer and Information Sciences III*. Springer, London, 303–310. https://doi.org/10.1007/978-1-4471-4594-3_31
3. Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. 2012. G3D: A gaming action dataset and real time action recognition evaluation framework. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 7–12. https://doi.org/10.1109/CVPRW.2012.6239175
4. Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. 2011. A Survey of Semantic Image and Video Annotation Tools. In *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*. Springer, Berlin, Heidelberg, 196–239. https://doi.org/10.1007/978-3-642-20795-2_8
5. Simon Fothergill, Helena Mentis, Pushmeet Kohli, and Sebastian Nowozin. 2012. Instructing People for Training Gestural Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 1737–1746. https://doi.org/10.1145/2207676.2208303
6. Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Hugo Jair Escalante. 2014. The ChaLearn gesture dataset (CGD 2011). *Machine Vision and Applications* 25, 8: 1929–1951. https://doi.org/10.1007/s00138-014-0596-3
7. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, Peter A. Hancock and Najmedin Meshkati (eds.). North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9
8. Michael Kipp. 2001. Anvil - A Generic Annotation Tool for Multimodal Dialogue. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, 1367–1370.
9. Michael Nebeling, David Ott, and Moira C. Norrie. 2015. Kinect Analysis: A System for Recording, Analysing and Sharing Multimodal Interaction Elicitation Studies. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (EICS '15), 142–151. https://doi.org/10.1145/2774225.2774846
10. Francis Quek, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E. McCullough, and Rashid Ansari. 2002. Multimodal Human Discourse: Gesture and Speech. *ACM Trans. Comput.-Hum. Interact.* 9, 3: 171–193. https://doi.org/10.1145/568513.568514
11. C. J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA.
12. Katharina Rohlfing, Daniel Loehr, Susan Duncan, Amanda Brown, Amy Franklin, Irene Kimbara, J-T Milde, Fey Parrill, Travis Rose, Thomas Schmidt, and others. 2006. Comparison of multimodal annotation tools: Workshop report. *Gesprächsforschung* 7.
13. R. Travis Rose, Francis Quek, and Yang Shi. 2004. MacVisSTA: a system for multimodal analysis. In *Proceedings of the 6th international conference on Multimodal interfaces*, 259–264.
14. Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1: 43–49. https://doi.org/10.1109/TASSP.1978.1163055
15. Advait Sarkar, Cecily Morrison, Jonas F. Dorn, Rishi Bedi, Saskia Steinheimer, Jacques Boisvert, Jessica Burggraaff, Marcus D'Souza, Peter Kontschieder, Samuel Rota Bulò, Lorcan Walsh, Christian P. Kamm, Yordan Zaykov, Abigail Sellen, and Siân Lindley. 2016. Setwise Comparison: Consistent, Scalable, Continuum Labels for Computer Vision. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 261–271. https://doi.org/10.1145/2858036.2858199
16. Thomas Schmidt and Kai Wörner. 2009. EXMARaLDA—Creating, analyzing and sharing spoken language corpora for pragmatics research. *Pragmatics-Quarterly Publication of the International Pragmatics Association* 19, 4: 565.
17. Isaac Wang, Mohtadi Ben Fraj, Pradyumna Narayana, Dhruva Patil, Gururaj Mulay, Rahul Bangar, J. Ross Beveridge, Bruce A. Draper, and Jaime Ruiz. 2017. EGGNOG: A Continuous, Multi-modal Data Set of Naturally Occurring Gestures with Ground Truth Labels. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, 414–421. https://doi.org/10.1109/FG.2017.145
18. Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. Elan: a professional framework for multimodality research. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 1556–1559. Retrieved from http://tla.mpi.nl/tools/tla-tools/elan/
19. JMF 2.1.1 - Supported Formats. Retrieved October 3, 2017 from http://www.oracle.com/technetwork/java/javase/formats-138492.html
20. Supported Media Formats in Media Foundation (Windows). Retrieved January 10, 2017 from https://msdn.microsoft.com/en-us/library/windows/desktop/dd757927
21. Kinect Studio. Retrieved October 3, 2017 from https://msdn.microsoft.com/en-us/library/dn785306.aspx