NATIONAL
ENDOWMENT
FOR THE
HUMANITIES

Sample Application

Program: Digital Humanities Start-Up Grants

Note: The attached sample application was awarded a grant during a previous competition. Note that resumes, letters of support, coversheets, and other pieces of the application that contain personal contact information have been removed.

1.0) Statement of CDL significance and impact

Chinese writing has been in continual use for more than 3,000 years, and Chinese-related writing systems are in use today by perhaps one-fifth of the world's population. Chinese and Chinese-derived (CJKV = Chinese, Japanese, Korean, Vietnamese) scripts preserve knowledge from many times and many places, but these scripts are so complex that they present immense challenges to digitization efforts. Because CJKV characters have traditionally not been encoded at a distinctive-feature level (i.e., the *character* is not the fundamental element of the writing system, but rather, the *stroke*), CJKV character-sets are open-ended, and writers take it for granted that for some purposes, writing with a computer is necessarily imperfect. The present proposal seeks to promote a practical and fundamental solution to this long-standing problem, to provide online tools for the use of international standards bodies, and ultimately to put these powerful tools in the hands of all who wish to use them. This work leads the way toward more precise and useful digitization of CJKV humanities collections, and promotes the international cooperation necessary to make the knowledge in these vast libraries available worldwide.

The *Character Description Language* (CDL) specification defines an XML application for describing any CJKV script entity (Bishop & Cook, 2003: <http://www.wenlin.com/cdl/>). Given the size and complexity of the current (Unicode 5.0) CJKV character-set (a.k.a. *UniHan*) which encodes some 72,000 characters, the task of maintaining and augmenting the international encoding is growing more difficult all the time. The CDL specification submitted to US and international standards bodies (Unicode and ISO/IEC 10646/WG2/IRG) inspired a great deal of work geared toward adoption of the CDL methods. At present, the only complete implementation of CDL is the C programming language implementation by Wenlin Institute (the educational software company owned by co-author Bishop, inventor of CDL). *Wenlin* is well known in Chinese second-language education as the preëminent software application for learning and studying Chinese (ancient and modern). Wenlin Institute is the maintainer of the Univ. of Hawaii's electronic *ABC Chinese-English Comprehensive Dictionary* data (DeFrancis et al.). The CDL database itself has been developed over the last decade in conjunction with co-author Cook (post-doctoral research fellow in the UC Berkeley Linguistics Dept., International Computer Science Institute, and co-editor of the *Unicode Standard 5.0*). The authors of this proposal have collaborated in the creation of some 56,000 CDL descriptions, and approximately 16,000 descriptions remain to be created, to cover the current encoded character-set. The present proposal aims at opening up the CDL system more widely, so that members of international standards bodies can employ CDL in their work to encode new characters and to secure and refine the mappings of existing characters. In the proposed funding period the proposers will create a prototype of an online centralized repository (MySQL database) of the existing CDL XML data, and complete a prototype version of the Wenlin client application for interacting with the server-side database. This client will be provided to the members of the international standards bodies contributing to *UniHan* development, and they will receive training in its use and access to the online CDL data. Long-term goals of the CDL project include creation of a public CDL interface to the Unicode Consortium's public *UniHan* database, and opening up the CDL source code itself to collaborative development, in an open-source framework compatible with Wenlin Institute's educational software business model. The proposed project will help to raise the profile of CDL among institutions and businesses with a vested interest in promoting long-term computing stability, and it is anticipated that present and future Unicode Consortium members will provide support for future open-source development.

2.0) Table of contents

[List all parts of the application and corresponding page numbers.]

The Character Description Language (CDL) Digital Humanities Start-up

3.0) List of CDL Project participants

[On a separate page, list in alphabetical order, surnames first, all project participants and collaborators and their institutional affiliations, if any. The names on this list should match the names mentioned in the staff section of the project's narrative description. The list is used to ensure that prospective panelists and reviewers have no conflict of interest with the project that they will be evaluating. This list should include advisory board members, if any.]

<table>
<tr><th colspan="2">Project Leaders</th></tr>
<tr><td>Bishop, Thomas E.</td><td>Wenlin Institute, Inc. (CEO and lead programmer); ABC Dictionary Project, University of Hawaii</td></tr>
<tr><td>Cook, Richard S.</td><td>Post-doctoral research fellow, Dept. of Linguistics, University of California, Berkeley; Artificial Intelligence Group, International Computer Science Institute; Unicode Consortium</td></tr>
<tr><th colspan="2">Advisory Board Members</th></tr>
<tr><td>Anderson, Deborah</td><td>Researcher, University of California, Berkeley Dept. of Linguistics; Script Encoding Initiative</td></tr>
<tr><td>Jenkins, John</td><td>Technical Director, Unicode Consortium, Apple Computer, Inc.</td></tr>
<tr><td>Lu Chin</td><td>Hong Kong Polytechnic University; Rapporteur, ISO/IEC JTC1/SC2/WG2/IRG</td></tr>
<tr><td>Lunde, Ken</td><td>Adobe Systems, Inc.</td></tr>
<tr><td>McGowan, Richard</td><td>Vice President, Unicode Consortium; Script Encoding Initiative</td></tr>
<tr><td>Whistler, Kenneth</td><td>Technical Director, Unicode Consortium, Sybase, Inc.</td></tr>
</table>

4.0) CDL Project Narrative

[Applicants should provide an intellectual justification for the project and a work plan. Narrative descriptions are limited to fifteen double-spaced pages. All pages should have one-inch margins and the font size should be no smaller than eleven point. Use appendices to provide supplementary material such as detailed work plans and résumés for project participants. The narrative should address the long-term goals for the project as well as the start-up activities that the Digital Humanities Start-Up Grant would support. Applicants should keep in mind the criteria (listed below) used to evaluate proposals. Provide a detailed project description that addresses the following topics:]

4.1) Enhancing the humanities through the use of CDL technologies

[Provide a clear and concise explanation of the start-up activities and the ultimate project results noting their value to scholars, students, and general audiences in the humanities. Describe the scope of the project activities, the relationship of the project to other published and ongoing work in the field, and major issues to be addressed. Applicants should provide a rationale for the compatibility of their methodological approach with the intellectual goals of the project and the expectations of its users. NEH views the use of open source software as a key component in the broad distribution of exemplary digital scholarship in the humanities. If either the start-up project or the long-term project is not predicated on generally accessible open source software, explain why and also explain how the Endowment's dissemination goals will still be satisfied by the project.]

The CDL project is directed in particular at resolving long-standing problems with the digitization of the Chinese, Japanese, Korean, and Vietnamese scripts (CJKV, or often simply shortened to CJK). The proposed start-up project will promote the collaborative use of the innovative CDL font technology, for the building of international computing standards essential to the stable function of all modern software, and for the accurate digitization and preservation of CJK documents and libraries. The CDL project benefits all computer users with an interest in CJK texts, or with an interest in dealing with CJK partners, since CDL has core applications for information input, storage, and retrieval. CDL improves data-management practices in the development of international standards and in the usage of end-users. CDL ensures the integrity of those standards, enriches the possibilities for end-user content creation, and therefore brings new richness to online digital humanities resources.

The CDL specification (Bishop & Cook, 2003) defines a standard (and standards-based) method for describing script entities (letters, characters, punctuation marks, symbols, etc.). The CDL method was developed specifically for CJK, but is not simply applicable to CJK. CDL has utility in the management of data for any character set for which there is some underlying reuse of basic components. The CDL XML application for CJK is based on a simple set of traditional stroke types, positioned in a standard grid space. Combinations of strokes define higher-level units, which may be reused in other CDL descriptions. The set of stroke types is termed "the ABC's of CJK", in that these strokes are the basic elements of CJK writing, rather like the alphabet is in English writing.

The problem solved by CDL is best explained by analogy with English. Imagine that instead of typing the letters of the alphabet on your computer keyboard to write English words, you were instead required to have a keyboard with one key for each *word* in the English language. This would be a big English keyboard indeed, for several reasons. There are many more words in English than there are lettres in the alphabet, and the notion of "English word" is not well-defined: there are many different varieties of English, and each variety may have unique words and unique spelling rules. As English lexicographers must grapple with the fuzzy definitions of "English" and "word" in the compilation of their dictionaries, they often limit themselves to a specific genre or specific genres of English written in a specific time and place with specific orthographic rules. The situation in CJK encoding is rather similar to this, in that the CJK *character* (comprised of strokes) is rather like the English *word* (comprised of letters). The CJK *character* is ill-defined for precisely the same reasons that the English *word* is ill-defined. If English writers were encumbered by the restrictions currently afflicting CJK writers, the situation would be obviously intolerable.

For example, some English writers would not be able to spell their surnames or personal names, and they would be unable to use Shakespearean spellings in production of editions of Shakespeare, unless those words or spellings happened to be in a dictionary, and that dictionary also happened to have been used as a source for a computer encoding (assigning a unique number to each word type). Spellings from the works of Shakespeare, being fairly well known, would surely be on everyone's computer; but quirky personal or place names, or words in use by lesser-known writers, would surely be lacking. For example, we would be unable to mention in the present Unicode 5.0 proposal document the fact that Shakespeare reportedly spelled his name variously in his own day as "Shakspere, Shaksper, Shaxper, and Shake-speare", without resorting to some non-standard text representation practice. The digitization of such documents would be hindered by the lack of prior lexicographic and encoding work, and this presents an unacceptable and unnecessary bottleneck. Given that a goal of Unicode is to provide an international CJK character encoding, transcending temporal and political boundaries, Unicode does not have the luxury of narrow scope, and in fact the problems faced in Unicode's CJK encoding are greatly compounded, requiring in effect innovative and comprehensive lexicographic work to be the gating constraint on the digitization of texts. Clearly, unless an adequate system such as CDL for computerized encoding and structured combination of the minimally distinctive features of the script is developed and promoted for wide adoption, the short-term progress and long-term success of CJK digitization projects must remain limited.

By design (and by *necessity*), Unicode unifies (i.e. lumps together) many variants of CJK characters that differ in small ways, such as the presence or absence of a dot, or whether stroke segments are joined or separated. For some purposes such variations may be very important. For example, different national standards sometimes assign different official stroke counts to the "same" character, and the stroke counts determine the organization of dictionaries, etc. For another example, historical Chinese texts can often be dated on the basis of a single stroke being intentionally omitted in taboo avoidance of the name of the reigning emperor. Unicode has recognized the need for identifying variants that share a code point, and for this purpose has established the "variant selector" mechanism. The need remains, however, to assign a precise meaning to each combination of code point and variant selector (see Hiura & Muller 2006: *UTS #37*, < http://www.unicode.org/reports/tr37/>). CDL is ideal for this purpose. Each code point can have multiple CDL descriptions, one description for each variant, uniquely identified by a selector.

Of course, there will always be relatively rare or obscure variants that have not (yet) been assigned standard variant selectors, as well as characters that have not (yet) been assigned Unicode code points. CDL can be used to represent such characters directly without the use of code points or variant selectors, to feed candidates into the encoding and variant-mapping processes. For example, a CDL description can be included directly in the text, which might employ a higher-level markup such as XHTML. If the displaying software is CDL-enabled, it can display the character simply by interpreting the CDL, without the need for a customized font, without the need for a *Private-Use Area* code point, and without the need for an embedded graphical image. When end-users are able to create CDL descriptions of characters, encoded or not, and can embed these in online documents, web spiders crawling the internet can collect them automatically, and use associated metadata to feed these descriptions into encoding and variant-mapping processes.

Another exciting (and still highly theoretical) application of CDL is to the problem of CJK *optical character recognition* (OCR). Current OCR technologies, when applied to CJK text, yield largely binary results: either a printed character is identified with an encoded character, or it is not. Where matching fails or is imperfect, either a completely wrong character is selected, or else OCR fails completely. Using CDL, the results of partial or failed OCR matches become meaningful. Just as the inability to recognize a single letter in OCR of English text might not result in failure to recognize the word, and just as the absence of an English word from a spell-checker's dictionary needn't signal complete OCR failure, so too OCR of unencoded or damaged CJK characters can succeed where current CJK OCR fails, by including CDL in the OCR output.

The applications of CDL technology for scripts beyond the CJK world are just as important. The CDL team has discussed this with researchers working on other scripts, contributing to the on-going development of Unicode to handle digital text representation of the scripts of the world. For example, there has been growing interest in developing CDL schemes for Cuneiform scripts, Tangut (西夏 Xīxià), Egyptian Hieroglyphs, Mayan, and other complex scripts with componential basis and special component and character positioning requirements. Such scripts are similarly limited by the open-endedness of their character sets due to historical and local variation, and due to the ill-defined nature of the bounds of the higher-level units of writing. There are, nevertheless, basic script elements identifiable in each of these scripts, that can be employed in conjunction with CDL to remove the encoding bottleneck, and to empower the paleographers, lexicographers and linguists who must ultimately seek to resolve these problems.

CDL is a core infrastructure technology, providing a rock-solid framework for data structure, data storage and data interchange, and CDL should be adopted internationally in work to digitize and preserve humanities collections and paper and archeological archives. Because CDL is *pure Unicode* and *pure XML*, and because its applications for CJK are clearly based on *traditional orthographic standards* active across CJK scripts (as evident in the finite set of stroke types appearing in the representative glyphs in Unicode code charts), *the CDL standard is completely standards-based*, and poised to have a significant impact in the international community. This contrasts with existing "analogue" font technologies (TrueType, PostScript, OpenType), which make use of machine-focused technologies, the elements of which, although effective at producing the character shapes on various output devices, sever the connection to the actual human orthographic practices underlying those shapes. A conventional TrueType font may represent the same characters as a given CDL font, and that TrueType font may support a certain limited reuse of components. But the TrueType font does not preserve information about the many levels of componential structure of the characters, nor does it know anything about traditional stroke types or stroke order. Critical indexing information is built right into the heart of the CDL font, whereas in analogue fonts bulky external indices are simply tacked on later by an input method creator in a process separate from the typography. Thus, CDL technology bridges a significant gap, putting real intelligence into fonts, and giving CJK digitization projects the freedom which roman-based orthographies take for granted. The returns on small initial investments in this technology will be large indeed, and shared globally, as significant improvements are made to electronic data stability, electronic data content, and electronic data access.

4.2) History and duration of the CDL project

[Provide a concise history of the project, including information about preliminary research or planning, previous related work, previous financial support, publications produced, and resources or research facilities available. It is anticipated that work on projects initiated during the term of a Digital Humanities Start-Up Grant will continue after the period of the grant. The applicant should describe plans for that work and probable sources of support for subsequent phases of the project.]

The project proposers have been breaking old and new ground in Chinese computing for the last two decades, and have been collaborating since the early 1990's, to resolve many problems critical to Chinese information processing. Wenlin Institute, Inc. has been a leader in promoting the use of Unicode in CJK computing, and Wenlin was among the first developers to implement support for the vast Unicode 3.1 character-set (which added more than 40,000 new CJK characters to the already large repertory). The CDL system arose out of gradual refinements to Wenlin Institute's software for learning Chinese (in particular, out of software written to teach students the proper stroke order of handwritten characters), and the applications of this technology for resolving long-standing computing problems gradually became apparent.

Wenlin Institute, Inc. has been a California corporation since 1996, and has a well-established reputation in the educational software industry. Wenlin Institute, Inc. is itself not a "start-up" (which often connotes a new business in search of a working business model), since its revenues derive from sales of mature software. However, its CDL technology branch (with special emphasis on building international and institutional cooperation to solve a thorny shared computing problem in an open-source environment) certainly does fit this notion of "start-up". Wenlin's CDL technology is a powerful tool in search of a ecological niche in which it can flourish, and it would be wise for the international encoding community which so badly needs this system, to share some of the development burden. In fact, appreciation of the wider applications of the CDL system arose in collaboration, with contributions from end-users, and this is the best way for CDL to continue.

Beginning in 1998 the Wenlin Unicode text editor and the unpublished and undocumented CDL font editing system were used by Dr. Cook in preparation of large lexical databases for the NSF and NEH-funded STEDT Project at UC Berkeley. This database work resulted in publication of the first-ever digitization of important ancient Chinese lexical sources (Cook 2003), and resulted also in contributions of extremely large mapping tables (with nearly 100,000 records) to the Unicode Consortium's public *UniHan* database. The *UniHan* database (<http://www.unicode.org/Public/UNIDATA/Unihan.html>) at present contains more than one million records, and is maintained by Dr. Cook in collaboration with John Jenkins of Apple Computer, Inc. Existing *UniHan* data has undergone and is undergoing extensive proofing and continual validation using the CDL system. Wenlin Institute has donated its software for the use of Unicode editors and staff, and has also donated CDL data for use in publication of the *Unicode Standard*, including both Traditional-to-Simpified mapping tables, and also data employed in the sorting of the Radical/Stroke charts appearing in the new *Unicode Standard 5.0* book. Wenlin Institute has contributed to international standards work directly, sending Mr. Bishop to ISO/IEC 10646 meetings at its own expense, and has received support from Kyoto University (*Text Encoding Initiative,* Dr. Christian Wittern) to promote CDL technology in Japan.

The elements of the CDL system were first publicly specified in two core documents (Bishop & Cook, 2003; these documents are also included in the Appendix, §6.0):

- **"A Specification for CDL: Character Description Language"** .
  <http://www.wenlin.com/cdl/cdl_spec_2003_10_31.pdf >
- **"Character Description Language (CDL): The Set of Basic CJK Unified Stroke Types"**
  <http://www.wenlin.com/cdl/cdl_strokes_2004_05_23.pdf>

That initial documentation grew in stages to become an entire website promoting the use of CDL technology (<http://www.wenlin.com/cdl/>), and tracking CDL-related developments in the international encoding community. That website includes links to a number of encoding proposals and other documents submitted to Unicode and ISO/IEC 10646, introducing the power of CDL technology to the members of these standards organizations. CDL technology has resulted in significant refinements to *UniHan* mapping tables, in the correction of numerous errors in the international encoding standard, and in the addition of a new block of CJK Stroke characters (forthcoming, in Unicode 5.1). This latter work resulted from IRG work inspired by our initial specification of CDL, and the new block of stroke characters is intended to make the use of CDL more intuitive for end-users.

The present proposal seeks to take CDL work one step further, to open up the CDL system more widely as a suite of tools not only for the international encoding community, but also to make these tools available to collaborators and end-users worldwide. It is our firm belief that once the power of this technology is appreciated, it must naturally come to be used widely, and that wide adoption of the technology is the first obstacle to be overcome in seeking sustainable open-source development. This technology will attract new members to the Unicode Consortium, members who will see, as current members have begun to see, that application of CDL to CJK problem-solving should be a high priority, and is best undertaken in a collaborative open-source framework. NEH seed funding will help to get this project online, and help to make it accessible to the wider audience that will sustain it in the long run.

4.3) CDL Project Staff

[Identify the project director and collaborators who would work on the project during the proposed grant period, and describe their responsibilities and qualifications. Provide résumés for the principal collaborators (maximum of two pages each) in an appendix. Project directors must devote a significant portion of their time to their projects. All persons directly involved in the conduct of the proposed project--whether or not their salaries are paid from grant funds--should be listed, their anticipated commitments of time should be indicated, and the reasons for and nature of their collaboration explained. If the project has an advisory board, provide a statement of its function and a list of board members.]

<table>
<tr><th colspan="2">Project Leaders (see resumés in §6.2)</th></tr>
<tr><td>Bishop, Thomas E.</td><td>Wenlin Institute, Inc. (CEO and lead programmer); ABC Dictionary Project, University of Hawaii</td></tr>
<tr><td colspan="2">Mr. Bishop is the inventor of CDL, and the principal software architect of Wenlin Institute's C programming language implementation of CDL. A staunch proponent of Unicode and educational software development, his role is key in the proposed project. Mr. Bishop will implement further refinements in the C source code, in order to produce the client application. He will also work to document the system, and introduce it to end-users. Total time commitment will be 33%.</td></tr>
<tr><td>Cook, Richard S.</td><td>Post-doctoral research fellow, Dept. of Linguistics, University of California, Berkeley; Artificial Intelligence Group, International Computer Science Institute; Unicode Consortium</td></tr>
<tr><td colspan="2">Dr. Cook is the main user and principal evangelist of CDL in the international encoding community. He is one of the editors of the <i>Unicode Standard</i>, and a maintainer of the <i>UniHan</i> component of the <i>Unicode Character Database</i>. In the proposed CDL project, he will develop the server-side software components. He will also assist in preparing the system documentation, and introduce it to end-users. Total time commitment will be 33%.</td></tr>
<tr><td colspan="2"></td></tr>
<tr><th colspan="2">Advisory Board Members</th></tr>
<tr><td>Anderson, Deborah</td><td>Researcher, University of California, Berkeley Dept. of Linguistics; Script Encoding Initiative</td></tr>
<tr><td colspan="2">Dr. Anderson has contributed logistical support in organizing IRG meetings, and it is anticipated that she will continue in this capacity in the future.</td></tr>
<tr><td>Jenkins, John</td><td>Technical Director, Unicode Consortium, Apple Computer, Inc.</td></tr>
<tr><td colspan="2">Mr. Jenkins is co-editor of <i>UniHan</i> with Dr. Cook, and a user of CDL.</td></tr>
<tr><td>Lu Chin</td><td>Hong Kong Polytechnic University; Rapporteur, ISO/IEC JTC1/SC2/WG2/IRG</td></tr>
<tr><td colspan="2">Dr. Lu is the leader of the IRG, and has been key in introducing CDL to IRG member delegates.</td></tr>
<tr><td>Lunde, Ken</td><td>Adobe Systems, Inc.</td></tr>
<tr><td colspan="2">Dr. Lunde is a strong proponent of CDL, and has organized many meetings on CDL at Adobe , Inc.; his contribution is essential in promoting the use of CDL in the international community.</td></tr>
<tr><td>McGowan, Richard</td><td>Vice President, Unicode Consortium; Script Encoding Initiative</td></tr>
<tr><td colspan="2">Mr. McGowan provides technical and other assistance, via the Unicode Consortium.</td></tr>
<tr><td>Whistler, Kenneth</td><td>Technical Director, Unicode Consortium, Sybase, Inc.</td></tr>
<tr><td colspan="2">Dr. Whistler is the inventor of the name "CDL", though he disagrees on what CDL stands for. Whistler will continue to assist us in the international encoding arena.</td></tr>
</table>

4.4) CDL Methods

[Explain the project's methods. • Describe in detail the tasks to be undertaken and the computer technology to be employed, indicating what technical and staff resources will be required, as well as the staff's experience with the technology and its application to the humanities. • Describe plans for evaluating the results of the start-up activities. This evaluation should be simultaneously summative with regard to the Digital Humanities Start-Up Grant and formative with regard to the long-term project goals.]

The CDL XML application as detailed in the *CDL Specification* (Bishop & Cook 2003; see the links given above in §4.2) is already fully implemented in C programming language in Wenlin Institute's *Wenlin* software package, and has been thoroughly tested in the creation of more than 56,000 descriptions. This software combines a full-featured Unicode text editor with an editing interface for the Wenlin CDL font database. Wenlin's CDL font technology is used internally in the editor as one means of character display (conventional fonts may also be used). CDL descriptions define script elements which may be used recursively in the definition of other script elements. The CDL font database is simultaneously a stroked font able to display characters at any resolution, and also an index into the CDL descriptions. Thus, the same information that displays the characters on the computer screen provides access to the characters for input (hand-writing recognition or stroke-based input method) or indexing (by stroke type and component structure). CDL descriptions can serve as the basis for regular expression searches, and keys of various types can be generated from this data, in order to perform validations prior to augmentation of the set (to prevent duplication, and to regularize descriptions). The descriptions in the CDL font database are rendered by means of the Wenlin interpreter, which is built into the editor. The CDL font format also supports embedded SVG, for display in applications lacking Wenlin's interpreter.

For the purpose of the present project, the Wenlin C implementation has been augmented with open-source code (libcurl) for communication with the server-side MySQL database via an intermediary CGI application running on the server. The Wenlin application running on the client will transmit CDL descriptions to the server, where they will be stored with versioning and other information, in the MySQL database. The Wenlin client will communicate user access control information to the server, and all changes to the centralized server database will be associated with a specific user. The CDL client at present allows practically infinite roll-back, via a variant mechanism which supports up to four billion CDL descriptions per Unicode code point. Similar roll-back functionality will also be supported in the server database. Very important in the long-term, CDL's variant mechanism is key for addressing the important issue of distinctive CJK character variation (in conjunction with variant selectors, registries, etc.).

The prototype client to be prepared in the current proposal will be a proof of concept, and we will not try to implement at once all of the features that will eventually be a part of the system. Instead, the client applications will have rather limited capabilities, aimed primarily at completing descriptions of the remaining 16,000 encoded characters. A future step will involve creating descriptions of the tens of thousands of characters which are currently in the pipeline for future Unicode encoding. This latter process can be boot-strapped on the basis of componential descriptions which are currently being constructed by other IRG contributors. A similar boot-strapping process was already employed in the creation of the existing CDL descriptions.

The CDL font editor is rather easy to use, but at present the only documentation is that available on the CDL website. In order to introduce collaborators to the use of CDL, tutorials will be conducted at IRG meetings during the funding period. The CDL team has already presented CDL at several IRG meetings, and both project leaders have extensive teaching experience, which should provide for a gentle introduction to this powerful technology.

4.5) Final CDL Product and Dissemination

[Describe the plans to disseminate the project results through various media (printed articles or books, presentations at meetings, electronic media or some combination). Applicants should also discuss how the project's ultimate product is likely to be disseminated and what provisions will be made for the long-term maintenance of such a product.]

The deliverables resulting from this CDL project will be functional software prototypes, documentation, and increased awareness of CDL in the international encoding community. Software prototypes of both the client and server will be produced, and made available to IRG delegates. The CDL documentation on the current CDL website will be expanded (leading eventually to a book on CDL, though the book might not be completed in the funding period). The elements of the CDL system and documentation will be presented at IRG conferences, to introduce details of system usage. The public CDL data repository will also have a simple browser-based query interface, for regular expression searches, and this will also be available on the CDL website at the conclusion of the funding period.

4.6) CDL Work Plan

[Describe the specific tasks that will be accomplished during the grant period and identify the staff members involved. The start-up activities described in the proposal should be completed by the end of the grant period.]

The Work Plan for the CDL start-up has sub-projects relating to accomplishing three main tasks:

1.) **Modifications to the Wenlin source code:**
Mr. Bishop will undertake modifications to Wenlin Institute's existing codebase, to permit the Wenlin client application (which runs on Windows and Macintosh computers) to communicate with the server-side MySQL database.

2.) **Server-side database and CGI application programming:**
Dr. Cook will create the server-side database environment (MySQL database and CGI programs) for communication with the Wenlin clients, and for the browser-based query system. These components will be hosted on one of the many servers on the domains at the CDL team's disposal (*wenlin.com*, *linguistics.berkeley.edu*, or *unicode.org*).

3.) **Documentation and promotion:**
The current CDL website (<http://www.wenlin.com/cdl/>) will be augmented with additional documentation, such as is necessary for end-users to understand proper usage of the CDL client. This website will also be updated with links to the browser-based query system, and with status pages giving status on the CDL MySQL database.

Work on these three components of the project will be undertaken concurrently, over the 18-month funding period, with a 33% time commitment by each of the CDL project leaders. We anticipate that prototypes of the two main software components, and also of the browser-based interface for regular-expression searching, can be completed within the first twelve months of funding, and that during the end of that period and in the remaining six-month funding period the system can be refined and made available to IRG members for initial testing.

The many special functions of the Wenlin client application (Unicode text editor, font editor, dictionary interface) will ideally be available in the future in a browser-based version, circumventing the need to install the stand-alone Wenlin client on individual machines. Creation of a browser-based version of Wenlin is however a more distant goal, dependent upon exploration of the necessary technologies.

5.0) CDL Project budget

[Budget narrative (optional) If needed, include a brief supplement to the narrative explaining projected expenses or other items in the financial information provided on NEH's budget form. The budget narrative may be single-spaced.]

The CDL project budget sheets are attached on the following pages, seeking funding for the 18-month period beginning in April, 2007.

OMB No. 3136-0134
Expires 6/30/09

NATIONAL ENDOWMENT FOR THE HUMANITIES

THREE-YEAR BUDGET FORM

Project Director: Richard S. Cook

Applicant Organization: Individual submission

Requested Grant Period From (mo/yr): 4/2007 Thru (mo/yr): 10/2008

If this is a revised budget, indicate the NEH application/grant number: ______

The three-column budget has been developed for the convenience of those applicants who wish to identify the project costs that will be charged to NEH funds and those that will be cost shared. **FOR NEH PURPOSES, THE ONLY COLUMN THAT NEEDS TO BE COMPLETED IS COLUMN C.** *The method of cost computation should clearly indicate how the total charge for each budget item was determined. If more space is needed for any budget category, please follow the budget format on a separate sheet of paper. Click HERE to see the detailed instructions.*

SECTION A – Year #1

Budget detail for the period FROM (mo/yr): 4/2007 THRU (mo/yr): 4/2008

When the proposed grant period is eighteen months or longer, project expenses for each twelve-month period are to be listed separately and totaled in the last column of the budget. For projects that will run less than eighteen months, only the last column of the budget should be completed.

1. Salaries and Wages

Provide the names and titles of the principal project personnel. For support staff, include the title of each position and indicate in brackets the number of persons who will be employed in that capacity. For persons employed on an academic year basis, list separately any salary charge for work done outside the academic year.

Name/Title of Position	No.	Method of Cost Computation (see sample)	NEH Funds (a)	Cost Sharing (b)	Total (c)
Dr. Richard S. Cook, PI, Programmer	[1]	$833.33/mo x 12 (stipend)	$ 10000	$ 0	$ 10000
Mr. Thomas E. Bishop, Lead Programmer	[1]	$833.33/mo x 12 (stipend)	$ 10000	$ 0	$ 10000
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
		SUBTOTAL	$ 20000	$ 0	$ 20000

2. Fringe Benefits

If more than one rate is used, list each rate and salary base.

Rate		Salary Base	(a)	(b)	(c)
	% of	$	$ 0	$ 0	$ 0
	% of	$	$	$	$
	% of	$	$	$	$
		SUBTOTAL	$ 0	$ 0	$ 0

3. Consultant Fees

Include payments for professional and technical consultants and honoraria.

Name or type of consultant	No. of days on project	Daily rate of compensation	NEH Funds (a)	Cost Sharing (b)	Total (c)
			$	$	$
			$	$	$
			$	$	$
			$	$	$
		SUBTOTAL	$ 0	$ 0	$ 0

4. Travel

For each trip, indicate the number of persons traveling, the total days they will be in travel status, and the total subsistence and transportation costs for that trip. When a project will involve the travel of a number of people to a conference, institute, etc., these costs may be summarized on one line by indicating the point of origin as "various." All foreign travel must be listed separately.

From/To	#	*	Subsistence Costs +	Transportation Costs =	(a)	(b)	(c)
US/Asia	[1]	[7]	$ 400	$ 1100	$ 2000	$ 0	$ 2000
SFO/IAD	[1]	[1]	$ 200	$ 300	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
				SUBTOTAL	$ 2000	$ 0	$ 2000

# - Number of persons   * - Total travel days

5. Supplies and Materials

Include consumable supplies, materials to be used in the project and items of expendable equipment (i.e., equipment items costing less than $5,000 and with an estimated useful life of less than a year).

Item	Basis/Method of Cost Computation	(a)	(b)	(c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

6. Services

Include the cost of duplication and printing, long distance telephone calls, equipment rental, postage, and other services related to project objectives that are not included under other budget categories or in the indirect cost pool. For subcontracts, provide an itemization of subcontract costs as an attachment.

Item	Basis/Method of Cost Computation	NEH Funds (a)	Cost Sharing (b)	Total (c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

7. Other Costs

Include participant stipends and room and board, equipment purchases, and other items not previously listed. Please note that "miscellaneous" and "contingency" are not acceptable budget categories. Refer to the budget instructions for the restriction on the purchase of permanent equipment.

Item	Basis/Method of Cost Computation	NEH Funds (a)	Cost Sharing (b)	Total (c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

8. Total Direct Costs (add subtotals of items 1 to 7) **$** 22000 **$** 0 **$** 22000

9. Indirect Costs

This budget item applies only to institutional applicants. If indirect costs are to be charged to this project, **CHECK THE APPROPRIATE BOX BELOW** and provide the information requested. Refer to the budget instructions for explanations of these options.

□ Current indirect cost rate(s) has/have been negotiated with federal agency. (Complete items A and B.)

□ Indirect cost proposal has been submitted to a federal agency, but not yet negotiated. (Indicate the name of the agency in Item A and show proposed rate(s) and base(s) and the amount(s) of indirect costs in item B.)

□ Indirect cost proposal will be sent to NEH if application is funded. (Provide in Item B an estimate of the rate that will be used and indicate the base against which it will be charged and the amount of indirect costs.)

□ Applicant chooses to use a rate not to exceed 10% of direct costs, less distorting items, up to a maximum charge of $5,000 per year. (Under Item B, enter the proposed rate, the base against which the rate will be charged, and the computation of indirect costs or $5,000 per year, whichever value is less.)

□ For Public Program projects only: Applicant is a sponsorship (umbrella) organization and chooses to charge an administrative fee of 5% of total direct costs. (Complete Item B.)

Item A. Name of federal agency: ____________________

Date of agreement: ____________________

Item B.

Rate(s)	Base(s)	NEH Funds (a)	Cost Sharing (b)	Total (c)
______% of	$ ____________	$ ______	$ ______	$ ______
______% of	$ ____________	$ ______	$ ______	$ ______
______% of	$ ____________	$ ______	$ ______	$ ______
	TOTAL INDIRECT COSTS	$ 0	$ 0	$ 0

10. Total Project Costs (Direct and Indirect) for budget period.	$ 22000	$ 0	$ 22000

NATIONAL ENDOWMENT FOR THE HUMANITIES

THREE-YEAR BUDGET FORM

Project Director: Richard S. Cook

Applicant Organization: Individual submission

Requested Grant Period From (mo/yr): 4/2007 Thru (mo/yr): 10/2008

If this is a revised budget, indicate the NEH application/grant number: ____________

The three-column budget has been developed for the convenience of those applicants who wish to identify the project costs that will be charged to NEH funds and those that will be cost shared. **FOR NEH PURPOSES, THE ONLY COLUMN THAT NEEDS TO BE COMPLETED IS COLUMN C.** *The method of cost computation should clearly indicate how the total charge for each budget item was determined. If more space is needed for any budget category, please follow the budget format on a separate sheet of paper.*

SECTION A – Year #2 (if needed)

Budget detail for the period FROM (mo/yr): 05/2008 THRU (mo/yr): 10/2008

When the proposed grant period is eighteen months or longer, project expenses for each twelve-month period are to be listed separately and totaled in the last column of the summary budget. For projects that will run less than eighteen months, only the last column of the summary budget should be completed.

1. Salaries and Wages

Provide the names and titles of the principal project personnel. For support staff, include the title of each position and indicate in brackets the number of persons who will be employed in that capacity. For persons employed on an academic year basis, list separately any salary charge for work done outside the academic year.

Name/Title of Position	No.	Method of Cost Computation (see sample)	NEH Funds (a)	Cost Sharing (b)	Total (c)
Dr. Richard S. Cook, PI, Programmer	[1]	$666.66/mo x 6 (stipend)	$ 4000	$ 0	$ 4000
Mr. Thomas E. Bishop, Lead Programmer	[1]	$666.66/mo x 6 (stipend)	$ 4000	$ 0	$ 4000
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
		SUBTOTAL	$ 8000	$ 0	$ 8000

2. Fringe Benefits

If more than one rate is used, list each rate and salary base.

Rate	Salary Base	(a)	(b)	(c)
% of	$	$	$	$
% of	$	$	$	$
% of	$	$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

3. Consultant Fees

Include payments for professional and technical consultants and honoraria.

Name or type of consultant	No. of days on project	Daily rate of compensation	NEH Funds (a)	Cost Sharing (b)	Total (c)
			$	$	$
			$	$	$
			$	$	$
			$	$	$
		SUBTOTAL	$ 0	$ 0	$ 0

4. Travel

For each trip, indicate the number of persons traveling, the total days they will be in travel status, and the total subsistence and transportation costs for that trip. When a project will involve the travel of a number of people to a conference, institute, etc., these costs may be summarized on one line by indicating the point of origin as "various." All foreign travel must be listed separately.

From/To	#	*	Subsistence Costs +	Transportation Costs =	(a)	(b)	(c)
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
				SUBTOTAL	$ 0	$ 0	$ 0

# - Number of persons   * - Total travel days

5. Supplies and Materials

Include consumable supplies, materials to be used in the project and items of expendable equipment (i.e., equipment items costing less than $5,000 and with an estimated useful life of less than a year).

Item	Basis/Method of Cost Computation	(a)	(b)	(c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

6. Services

Include the cost of duplication and printing, long distance telephone calls, equipment rental, postage, and other services related to project objectives that are not included under other budget categories or in the indirect cost pool. For subcontracts, provide an itemization of subcontract costs as an attachment.

Item	Basis/Method of Cost Computation	NEH Funds (a)	Cost Sharing (b)	Total (c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

7. Other Costs

Include participant stipends and room and board, equipment purchases, and other items not previously listed. Please note that "miscellaneous" and "contingency" are not acceptable budget categories. Refer to the budget instructions for the restriction on the purchase of permanent equipment.

Item	Basis/Method of Cost Computation	NEH Funds (a)	Cost Sharing (b)	Total (c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

8. Total Direct Costs (add subtotals of items 1 to 7) **$** 8000 **$** 0 **$** 8000

9. Indirect Costs

This budget item applies only to institutional applicants. If indirect costs are to be charged to this project, **CHECK THE APPROPRIATE BOX BELOW** and provide the information requested. Refer to the budget instructions for explanations of these options.

□ Current indirect cost rate(s) has/have been negotiated with federal agency. (Complete items A and B.)

□ Indirect cost proposal has been submitted to a federal agency, but not yet negotiated. (Indicate the name of the agency in Item A and show proposed rate(s) and base(s) and the amount(s) of indirect costs in item B.)

□ Indirect cost proposal will be sent to NEH if application is funded. (Provide in Item B an estimate of the rate that will be used and indicate the base against which it will be charged and the amount of indirect costs.)

□ Applicant chooses to use a rate not to exceed 10% of direct costs, less distorting items, up to a maximum charge of $5,000 per year. (Under Item B, enter the proposed rate, the base against which the rate will be charged, and the computation of indirect costs or $5,000 per year, whichever value is less.)

□ For Public Program projects only: Applicant is a sponsorship (umbrella) organization and chooses to charge an administrative fee of 5% of total direct costs. (Complete Item B.)

Item A. Name of federal agency: ____________________

Date of agreement: ____________________

Item B.

Rate(s)	Base(s)	NEH Funds (a)	Cost Sharing (b)	Total (c)
______% of	$ ______	$ ______	$ ______	$ ______
______% of	$ ______	$ ______	$ ______	$ ______
______% of	$ ______	$ ______	$ ______	$ ______
	TOTAL INDIRECT COSTS	$ 0	$ 0	$ 0

	NEH Funds (a)	Cost Sharing (b)	Total (c)
10. Total Project Costs (Direct and Indirect) for budget period.	$ 8000	$ 0	$ 8000

NATIONAL ENDOWMENT FOR THE HUMANITIES

THREE-YEAR BUDGET FORM

Project Director: Richard S. Cook

Applicant Organization: Individual submission

Requested Grant Period From (mo/yr): 4/2007 Thru (mo/yr): 10/2008

If this is a revised budget, indicate the NEH application/grant number: __________

The three-column budget has been developed for the convenience of those applicants who wish to identify the project costs that will be charged to NEH funds and those that will be cost shared. **FOR NEH PURPOSES, THE ONLY COLUMN THAT NEEDS TO BE COMPLETED IS COLUMN C.** *The method of cost computation should clearly indicate how the total charge for each budget item was determined. If more space is needed for any budget category, please follow the budget format on a separate sheet of paper.*

SECTION A – Year #3 (if needed)

Budget detail for the period FROM (mo/yr): __________ THRU (mo/yr): __________

When the proposed grant period is eighteen months or longer, project expenses for each twelve-month period are to be listed separately and totaled in the last column of the summary budget. For projects that will run less than eighteen months, only the last column of the summary budget should be completed.

1. Salaries and Wages

Provide the names and titles of the principal project personnel. For support staff, include the title of each position and indicate in brackets the number of persons who will be employed in that capacity. For persons employed on an academic year basis, list separately any salary charge for work done outside the academic year.

Name/Title of Position	No.	Method of Cost Computation (see sample)	NEH Funds (a)	Cost Sharing (b)	Total (c)
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
	[ ]		$	$	$
		SUBTOTAL	$ 0	$ 0	$ 0

2. Fringe Benefits

If more than one rate is used, list each rate and salary base.

Rate	Salary Base	(a)	(b)	(c)
% of	$	$	$	$
% of	$	$	$	$
% of	$	$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

3. Consultant Fees

Include payments for professional and technical consultants and honoraria.

Name or type of consultant	No. of days on project	Daily rate of compensation	NEH Funds (a)	Cost Sharing (b)	Total (c)
			$	$	$
			$	$	$
			$	$	$
			$	$	$
		SUBTOTAL	$ 0	$ 0	$ 0

4. Travel

For each trip, indicate the number of persons traveling, the total days they will be in travel status, and the total subsistence and transportation costs for that trip. When a project will involve the travel of a number of people to a conference, institute, etc., these costs may be summarized on one line by indicating the point of origin as "various." All foreign travel must be listed separately.

From/To	#	*	Subsistence Costs +	Transportation Costs =	(a)	(b)	(c)
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
	[ ]	[ ]	$	$	$	$	$
				SUBTOTAL	$ 0	$ 0	$ 0

# - Number of persons   * - Total travel days

5. Supplies and Materials

Include consumable supplies, materials to be used in the project and items of expendable equipment (i.e., equipment items costing less than $5,000 and with an estimated useful life of less than a year).

Item	Basis/Method of Cost Computation	(a)	(b)	(c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

6. Services

Include the cost of duplication and printing, long distance telephone calls, equipment rental, postage, and other services related to project objectives that are not included under other budget categories or in the indirect cost pool. For subcontracts, provide an itemization of subcontract costs as an attachment.

Item	Basis/Method of Cost Computation	NEH Funds (a)	Cost Sharing (b)	Total (c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

7. Other Costs

Include participant stipends and room and board, equipment purchases, and other items not previously listed. Please note that "miscellaneous" and "contingency" are not acceptable budget categories. Refer to the budget instructions for the restriction on the purchase of permanent equipment.

Item	Basis/Method of Cost Computation	NEH Funds (a)	Cost Sharing (b)	Total (c)
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
		$	$	$
	SUBTOTAL	$ 0	$ 0	$ 0

8. Total Direct Costs (add subtotals of items 1 to 7) **$** 0 **$** 0 **$** 0

9. Indirect Costs

This budget item applies only to institutional applicants. If indirect costs are to be charged to this project, **CHECK THE APPROPRIATE BOX BELOW** and provide the information requested. Refer to the budget instructions for explanations of these options.

□ Current indirect cost rate(s) has/have been negotiated with federal agency. (Complete items A and B.)

□ Indirect cost proposal has been submitted to a federal agency, but not yet negotiated. (Indicate the name of the agency in Item A and show proposed rate(s) and base(s) and the amount(s) of indirect costs in item B.)

□ Indirect cost proposal will be sent to NEH if application is funded. (Provide in Item B an estimate of the rate that will be used and indicate the base against which it will be charged and the amount of indirect costs.)

□ Applicant chooses to use a rate not to exceed 10% of direct costs, less distorting items, up to a maximum charge of $5,000 per year. (Under Item B, enter the proposed rate, the base against which the rate will be charged, and the computation of indirect costs or $5,000 per year, whichever value is less.)

□ For Public Program projects only: Applicant is a sponsorship (umbrella) organization and chooses to charge an administrative fee of 5% of total direct costs. (Complete Item B.)

Item A. Name of federal agency: ____________________

Date of agreement: ____________________

Item B.

Rate(s)	Base(s)	NEH Funds (a)	Cost Sharing (b)	Total (c)
________% of	$ ____________	$ ________	$ ________	$ ________
________% of	$ ____________	$ ________	$ ________	$ ________
________% of	$ ____________	$ ________	$ ________	$ ________
	TOTAL INDIRECT COSTS	$ 0	$ 0	$ 0

10. Total Project Costs

$ 0 $ 0 $ 0

(Direct and Indirect) for budget period.

SECTION B

SUMMARY BUDGET

Transfer from Section A the total costs (column C) for each category of project expense. When the proposed grant period is eighteen months or longer, project expenses for each twelve-month period are to be listed separately and totaled in the last column of the summary budget. For projects that will run less than eighteen months, only the last column of the summary budget should be completed.

Budget categories	First year from: 4/2007 thru: 4/2008	Second year from: 05/2008 thru: 10/2008	Third year from: thru:		TOTAL COSTS FOR ENTIRE GRANT PERIOD
1. Salaries and wages	$ 20000	$ 8000	$ 0	=	$ 28000
2. Fringe benefits	$ 0	$ 0	$ 0	=	$ 0
3. Consultant fees	$ 0	$ 0	$ 0	=	$ 0
4. Travel	$ 2000	$ 0	$ 0	=	$ 2000
5. Supplies and materials	$ 0	$ 0	$ 0	=	$ 0
6. Services	$ 0	$ 0	$ 0	=	$ 0
7. Other costs	$ 0	$ 0	$ 0	=	$ 0
8 Total direct costs (Items 1-7)	$ 22000	$ 8000	$ 0	=	$ 30000
9. Indirect costs	$ 0	$ 0	$ 0	=	$ 0
10. Total project costs (direct and indirect)	$ 22000	$ 8000	$ 0	=	$ 30000

PROJECT FUNDING FOR ENTIRE GRANT PERIOD

1. Indicate the amount of outright and/or federal matching funds that is requested from NEH.

2. Indicate the amount of cash contributions that will be made by the applicant and cash and in-kind contributions made by third parties to support project expenses that appear in the budget. Cash gifts that will be raised to release federal matching funds should be included under "Third-party contributions." (Consult the program guidelines for information on cost sharing requirements.) When a project will generate income that will be used during the grant period to support expenses listed in the budget, indicate the amount of income that will be expended on budgeted project activities. Indicate funding received from other federal agencies.

3. Total Project Funding should equal Total Project Costs.

1. REQUESTED FROM NEH		**2. COST SHARING**	
Outright	$ 30000	Applicant's contributions	$
Federal Matching	$ 0	Third-party contributions	$
		Project income	$
		Other federal agencies	$
TOTAL NEH FUNDING	$ 30000	TOTAL COST SHARING	$ 0

3. TOTAL PROJECT FUNDING (Total NEH Funding + Total Cost Sharing) : **$** 30000

6.0) CDL Project Application Appendices

[Use appendices to provide essential supplementary materials. Include a brief résumé (two-page maximum) for each principal project participant and letters of commitment from other participants and cooperating institutions. Descriptive material from preliminary work or previous periods of support may be included in an appendix, but should be limited to essential information.]

The elements of the CDL system were first publicly specified in two core documents (Bishop & Cook, 2003; these are included in this appendix, for the convenience of reviewers reading paper print-outs of this application):

- **"A Specification for CDL: Character Description Language"** .
  <http://www.wenlin.com/cdl/cdl_spec_2003_10_31.pdf >

- **"Character Description Language (CDL): The Set of Basic CJK Unified Stroke Types"**
  <http://www.wenlin.com/cdl/cdl_strokes_2004_05_23.pdf>

Additionally, we attach a copy of the recently approved encoding proposal **N3063**, adding 20 stroke types to the **CJK Strokes** block of Unicode (ISO/IEC 10646):

- **"Proposed additions to the CJK Strokes block of the UCS"** .
  <http://www.dkuug.dk/jtc1/sc2/wg2/docs/n3063.pdf>

The CJK Strokes block, encoding a total 36 strokes (U+31C0 .. U+31E3) resulted from the proposers' work with the US and international standards bodies (Unicode and ISO/IEC 10646/WG2/IRG), and the proposal itself was completed in IRG sub-committee work hosted at UC Berkeley in November, 2005 (with financial assistance from the UC Berkeley Townsend Center for the Humanities, the UC Berkeley Dept. of Linguistics, the UC Berkeley Institute of East Asian Studies, and the UC Berkeley East Asian Library).

For more information on the domestic and international standards bodies and other organizations contributing to this work, please see the following links:

- **The Unicode Consortium**
  <http://www.unicode.org/>

- **The Ideographic Rapporteur Group (ISO/IEC 10646/WG2/IRG)**
  <http://www.cse.cuhk.edu.hk/~irg/>
  <http://www.cse.cuhk.edu.hk/~irg/irg/irg25/IRG25.htm>

- **The Script Encoding Initiative**
  <http://www.linguistics.berkeley.edu/sei/>

- **The Sino-Tibetan Etymological Dictionary and Thesaurus Project**
  <http://stedt.berkeley.edu/>

- **Wenlin Institute, Inc.**
  <http://www.wenlin.com/>

The final two appendices include resumés for the CDL project leaders.

A Specification for CDL

Character Description Language

Source: Tom Bishop <tbishop@wenlin.com> and Richard Cook <rscook@unicode.org>

Status: Expert Contribution

Date: 2003-10-31 (slightly corrected 2003-12-21)

Action: For consideration by UTC and IRG

Contents

INTRODUCTION

Character Description Language (CDL) is for accurately describing and displaying the forms of all Han (CJKV) characters. This document, which is the first public specification of CDL, presents the key features and syntax of the language, and discusses some of its applications, especially to character encoding standards work. We propose adoption of CDL as a data management tool for ensuring accuracy and long-term stability in the public character encoding process.

The acute need for CDL is predicated upon the fact that the set of Han characters is truly open-ended, rather like the set of English words. Historical and idiosyncratic spelling differences present a vast quantity of data, and a large number of forms not easily related to currently encoded forms. Witness the tens of thousands of characters being evaluated by the IRG for inclusion in CJK Unified Ideographs Extension C1.

CDL is based on Unicode, XML, and a few well-known characteristics of Han characters:

- Most characters are formed by combining two or more simpler characters or components and fitting them into a square.
- Basic characters or components are composed of strokes, which are classified into distinct types in accordance with modern orthographic conventions.
- Identification of stroke types underlies consistent counting of strokes.
- Stroke types, stroke counts, and component analysis are essential to the learning process, character recognition, indexing, and comparison of variant forms.

A set of less than fifty stroke types is sufficient for the construction of practically all characters in a modern printed style, as demonstrated by the existence of CDL descriptions for over 40,000 characters, including all BMP Han characters and over 12,000 in Extension B.

THE CDL FONT DATABASE

A CDL description of a character encodes an analysis of the character into its constituent components and/or strokes, and simultaneously provides instructions for displaying the character. A collection of CDL descriptions can therefore serve as both a database and a font.

When CDL is used as a font format, a software interpreter converts the descriptions into glyphs in real time. For Han characters, CDL has some advantages over conventional font formats. It is much smaller — only about 12 bytes per character, on average, when compressed. It is a kind of “meta-font” in the sense that it has variable parameters so that the same descriptions can produce different styles of glyphs.[1] New glyphs can be added to the font relatively quickly and easily. Consistency between the forms of related characters is easier to ensure as a consequence of the sharing of components.

As a database language, CDL encodes essential information for categorizing, indexing, learning, and recognizing Chinese characters. This information includes stroke count, stroke types, stroke order, component analysis, radicals and residual strokes, and coordinates of strokes and components. While some of this information is, or could be, stored in an ordinary database, CDL is better for enforcing consistency. For example, the stroke count of a character is calculated algorithmically from actual CDL instructions for writing the character stroke-by-stroke; it is not merely a personal impression, or gathered from one of various dictionaries that may not be mutually consistent (or even individually self-consistent) in counting the strokes of a particular component.

EXAMPLES

Here is a description for 行, as a combination of the components 彳 and 亍:

```
<cdl char="行">
     <comp char="彳" points="0,0 40,128" />
     <comp char="亍" points="60,12 128,128" />
</cdl>
```

Positions are given as points with two-dimensional coordinates. The square enclosing the entire character has (x, y) coordinates ranging from (0, 0) for the top left corner, to (128, 128) for the bottom right corner.[2] The numbers after 彳 describe its bounding rectangle on the left side of 行: (0, 0) is its top left corner, and (40, 128) is its bottom right corner.[3] Similarly, a rectangle is given for 亍 on the right side of 行.

In order for the above CDL description to be carried out as a set of instructions (e.g., for displaying the character or counting its strokes), it is necessary for the interpreter to refer to the separate descriptions of the components, 彳 and 亍, as sequences of particular stroke types with specific coordinates. Here is a description[4] for 彳:

```
<cdl char="彳">
     <stroke type="p" points="107,0 10,46" />
     <stroke type="p" points="128,38 0,83" />
     <stroke type="s" points="86,70 86,128" />
</cdl>
```

There are three strokes in 彳. The first two (from top to bottom) are both type ‘p’, which stands for 撇 *piě*, a curved stroke falling to the left. The third stroke is type ‘s’, which stands for 竖 *shù*, a vertical falling stroke. For each of these simple stroke types, only two points are needed. For example, the first stroke starts at (107, 0) and ends at (10, 46).

Some descriptions combine components and strokes. Here, the character 太 is described as a combination of the component 大 (which itself is a character, and should have its own description), and a stroke of type ‘d’ (点 *diǎn*, dot):

```
<cdl char="太">
     <comp char="大" points="0,0 128,128" />
     <stroke type="d" points="45,104 66,128" />
</cdl>
```

LANGUAGE DETAILS

CDL is an XML application, which means that it conforms to a widely-used standard syntax (usage of angle brackets < >, et cetera). We have already introduced most of the elements of the language: each description is contained in a `cdl` element, which can contain any number of `comp` (component) and/or `stroke` elements. There is another element, `cdl-list`, for enclosing a list (or file, font, or database) of descriptions. The only CDL elements currently defined are these four: `cdl-list`, `cdl`, `comp`, and `stroke`.

Both the `cdl` and `comp` elements have `char` (character) attributes. The value of the `char` attribute is simply a character: typically a Han character, which might be encoded with UTF-8 or any other XML-supported encoding. Any character can, in principle, be used as a component.[5]

The `stroke` element has a `type` attribute, whose value is one of less than fifty names of stroke types that are defined for CDL. This article has already introduced ‘p’ for 撇 *piě* and a few others. One of the most complex stroke types is ‘hzzzg’, which stands for 横折折折钩 *héng-zhé-zhé-zhé-gōu*, and is exemplified by the character 乃. It has six reference points, including four points of inflection between the starting and ending points. The essential features of each stroke type are: its name; the number of reference points it uses; and the directions and curvatures between the reference points. The complete set of CDL stroke types is documented in another article.[6, 7]

There is a form of recursion implied by CDL. For example, a description of 龍 may refer (with a `comp` tag) to a description of 立, which in turn may refer (with another `comp` tag) to a description of 亠, which describes two individual strokes. A CDL interpreter will therefore typically process components within components within components, using recursive algorithms (and scaling coordinates according to bounding rectangles). Recursion stops when `stroke` elements are reached.[8]

Any CDL description that uses `comp` elements can be transformed automatically into a description that uses only `stroke` elements. For example, 行 is described as a sequence of two components 彳 and 亍, each of which is in turn described as a sequence of three strokes. Alternatively, 行 could be described directly as a sequence of six strokes. A straightforward recursive algorithm can transform the component description into the “strokes-only” description. The reverse transformation might be more difficult. Component descriptions are more generally useful as well as more concise.[9]

EXTENDING THE PRECISION AND SCOPE OF CHARACTER SETS

Compared with even the largest standard character set, CDL provides more precision: the ability to distinguish between unified variants. It also provides wider scope: a potentially infinite number of Han characters.

CDL can describe and display particular variants of characters that are "unified" (treated as equivalent) in standard character sets. For example, in Unicode the forms 者 (eight strokes) and 者 (nine strokes) are both `U+8005`, but can be made distinct using CDL.[10]

CDL can also be used for describing and displaying characters that are not in any standard character set. Some such characters might simply not have been encoded yet; some might be new; some might have extremely limited and special usages, and therefore might not even be suitable for inclusion in a standard character set.

The CDL instructions for displaying a character can be composed whenever the need arises (preferably using a graphical user interface), and included directly in a document using XML syntax. Of course, the program displaying the text needs to have the capability of interpreting the language, possibly by means of a "plug-in" or "helper" application; people reading the text simply see the resulting image of the character, not the CDL tags.

MANAGING DATA FOR CHARACTER SET STANDARDIZATION

By simultaneously producing both a (meta-)font and a database, CDL can enable standards organizations to publish representative glyphs and stroke counts (etc.), that are consistent with each other. Furthermore, the language can facilitate systematic treatment of the complex and difficult problems of unification and variation. Currently, such systematic treatment is held back by the absence of an intermediate representation of character forms, between abstract "characters" and concrete "images" (or particular written/printed instances) of characters. Each Unicode codepoint represents an abstract character, which corresponds to a potentially infinite number of graphic images. Graphic images are useful as examples of characters, but it is practically impossible, in general, for an algorithm to determine the stroke count of an image, or to measure the degree of similarity between two images according to the principles of Han unification. Consequently, with over 70,000 Han characters already encoded, it has become difficult to determine whether a given glyph corresponds to any of the characters that have already been encoded. Really there are two difficulties: first, to find all the likely candidates for codepoints that might correspond to the glyph in question; second, to decide for each of those codepoints whether the glyph belongs to that codepoint's implicit equivalence class according to the unification principles.

CDL can help resolve both of the difficulties just mentioned. A CDL database could be built for all encoded Han characters. Each character could potentially have multiple CDL descriptions, corresponding to variants[11] that have been unified. Then, when confronted with a glyph, if one were uncertain whether it was already encoded, one could construct a CDL description for it, and run a program to compare that description with those already in the database, to find the closest matches. (Several comparison algorithms could be applied for the same character, some based on strokes, some based on components.) Of course, a perfect match would be unlikely, but trivial differences in coordinates or stroke order would easily be recognized as falling within the scope of unification. Less trivial differences would still require judgment by experts, but CDL would make it far easier for the experts to apply the unification rules consistently. For example, all the characters containing a given component could be examined to discover any precedent for unifying two variants of that component. If the decision were made to unify the new glyph with an already encoded character, in spite of some difference, then the CDL for the new glyph could be added to the database as a variant, thus providing a precedent, making the unification rules more explicit, and facilitating future usage of the database.[12, 13]

ORIGIN AND CURRENT STATUS

CDL was originally designed and implemented (in the C programming language) by one of the authors, and is an integral part of *Wenlin Software for Learning Chinese,* published by Wenlin Institute, Inc. Its original application was Wenlin's *Stroking Box*, which illustrates for learners how to write a character stroke-by-stroke in slow motion. It turned out to be fast enough for use as a general-purpose scalable font. It also provides stroke-count and stroke-type information, and is even applied to handwriting recognition. However, the CDL language itself is hidden from the user, and only the resulting stroked characters are visible. Wenlin actually uses a compressed binary format, which is equivalent to the XML format, but very compact and fast for machine processing. Wenlin's CDL was used to create printed radical and stroke-order indexes for 9,638 characters in the *ABC Chinese-English Comprehensive Dictionary,* published in 2003 by University of Hawaii Press (ISBN 0-8248-2766-X).

Currently (October 2003) over 40,000 characters have CDL descriptions, including all the Han characters in Unicode 3.0 (with Extension A) and many more that are in Unicode 4.0 (Extension B). These descriptions were made by the authors.

CONCLUSION

Experience has shown CDL to be a useful language for systematic treatment of Han characters. While it undoubtedly still has room for improvement, the authors have become convinced (with the encouragement of several members of the Unicode Technical Committee) that it should be made public for the benefit of the international community, especially standards organizations. Comments, questions, and suggestions are welcome.

REFERENCES

The latest revision of this article, and other information about CDL (including the list of stroke types and a DTD[14]), may be found at http://www.wenlin.com/cdl.

The Unicode Consortium website is http://www.unicode.org. The International Standards Organization (ISO) website is http://www.iso.org. The Ideographic Rapporteur Group (IRG) website is http://www.cse.cuhk.edu.hk/~irg.

XML (Extensible Mark-up Language) is described at http://www.xml.org and http://www.w3.org/XML.

NOTES

1. The concept of a "meta-font" originated with the METAFONT language (documented in The METAFONTbook by Donald Knuth, 1986, ISBN 0-201-13445-4). Although CDL is not closely related to METAFONT, there is a procedure for converting CDL into METAFONT, but currently only at a low-level in which the glyph outline is exactly specified. A similar procedure exists for converting CDL into the PostScript language (PostScript is a trademark of Adobe; see http://www.adobe.com).

2. All coordinates are decimal integers in the range 0 through 128. CDL could easily be extended to allow floating-point numbers and/or different ranges of coordinates. However, the use of small integers and a power of two like 128 leads to compact storage and fast rendering even on slow machines, and has been found to give plenty of precision. More sophisticated versions of the language should allow symbolic variable names, and even algebraic expressions, to stand for coordinates. It should be possible to convert automatically from such "higher level" versions of CDL into the basic "low-level" version of CDL that uses only numerical coordinates. For some purposes, it is likely to be convenient to describe component and stroke positions with less precision, with rough indications such as top, left, top-left, middle, etc.; there should be utilities to support conversion back and forth between such rough indications and precise coordinates.

3. A clarification is needed regarding coordinates and bounding rectangles. In general, the reference points for a stroke are inside the stroke, roughly at the center of the tip of an imaginary brush. For a thick stroke, the fat tip of the brush may extend the radius of "ink" a considerable distance in all directions from the reference point. The precise flow of ink depends on the particular font style, and the same CDL description could be displayed differently by different interpreters (or by the same interpreter, given a different set of preferences). What we mean by the bounding rectangle of a component is based only on the reference points; "extra ink" might extend about half the thickness of a stroke in any direction beyond that rectangle.

4. The description for 彳 has been simplified slightly to make it easier to understand. A better description might use the optional `points` attribute of the `cdl` tag. Rather than simply `<cdl char="彳">`, the opening tag might be `<cdl char="彳" points="24,0 104,128">`. This means that when 彳 is displayed by itself, it does not take up the entire square, but instead has some space on both sides, making it relatively tall and narrow. When 彳 (or any character) is used as a component, however, this `points` attribute is ignored, since the `comp` tag has its own `points` attribute. The stroke points should always make a component touch all four edges of its grid, so that its bounding rectangle is `0,0 128,128` before any scaling is applied. The `points` attribute is even more important for 口 ("mouth"), which has

a large amount of space on all four sides; characters like 因 look best with a smaller amount of space on all four sides. In general, any character with a stroke running along an outside edge tends to look better (especially in juxtaposition with other characters) with some space on that edge; so, 相 might have `points="0,0 124,128"`.

5. Instead of, or in addition to, the char attribute, CDL supports a `uni` attribute, whose value is a hexadecimal Unicode scalar value (USV). For example, `uni="592A"` has the same meaning as `char="太"`. Simultaneous use of `char` and `uni` attributes is redundant but sometimes convenient. If both are used, they should be consistent. An optional `variant` attribute can be used in addition to either `char` or `uni`, to associate identification strings for distinguishing multiple descriptions for the same USV.

6. There is a widely-used system, commonly known as the 札 *zhá* system, which puts all strokes into only five stroke categories: 一 *héng*, 丨 *shù*, 丿 *piě*, 丶 *diǎn*, and 乛 *zhé*. Each of the less than fifty CDL stroke types belongs to one of the five 札 *zhá* stroke categories. Thus, 札 *zhá* classification can easily be obtained from a CDL description.

7. There are `head` and `tail` attributes for stroke elements, which describe minor changes to beginning and end points of strokes, respectively. Such changes are important for some typeface styles, especially where strokes join; however, they can be ignored for some simple styles, and for many applications of CDL. They are documented in another article, along with the list of stroke types.

8. A more explicit form of recursion could be supported, with one `cdl` tag allowed to occur inside of another, acting as an anonymous component. This would be one solution to the problem of unencoded components. Another solution is to assign private-use codes to unencoded components, give them separate descriptions in the same database, and use `comp` tags. The latter solution has the advantage that the same component can be used in more than one character without duplicating its description. Ideally, however, there should be standard (not private-use) codes for many components that are useful in CDL.

9. There are a few more optional attributes (such as a `radical` attribute for specifying which strokes in a character are considered to be its radical), which are beyond the scope of this article.

10. Actually, Unicode includes two compatibility characters, related to `U+8005` 者, namely `U+FA5B` and `U+2F97A`. The difference between them seems to involve a slight difference in the position of the extra dot.

11. In this context, we only distinguish "variants" if they have nontrivial differences in their CDL descriptions. (Slight coordinate differences can be regarded as trivial.) If there are two or more distinct CDL descriptions of a unified character, we call them all "variants" of each other, without any implication about relative correctness or deviance, since in general those qualities depend on the locale, the context, and/or the eye of the beholder.

12. While CDL can't solve all the difficulties of Han unification and variation, it can go a long way toward making the principles and procedures more rational. Just the ability to

produce self-consistent radical and stroke-count indexes of the currently encoded Han characters will be an advance. It would be a mistake to assume that stroke count will always be fuzzy and ill-defined, and that when looking up a character, people will always have to be prepared to add or subtract one or two from the stroke count when their first guess fails. On the contrary, within particular locales, such as the PRC, a tremendous amount of careful work has been done, and official publications such as 现代汉语通用字笔顺规范 (ISBN 7-80126-201-8) have standardized not only the stroke counts but also the stroke orders and stroke categories for thousands of characters. The stroke count of one character is generally related to the stroke counts of other characters. Most characters are built from components, and as long as the stroke counts of those components are defined, there is rarely any difficulty in adding them together to obtain the combined stroke count. Therefore, if a standard defines the strokes of a few thousand characters, it implicitly defines the strokes of many thousands of additional characters.

13. There are conflicting conventions (in different countries, or even in the same country) for the strokes of some characters that are nevertheless unified in Unicode. Using a `variant` attribute in addition to a `char` (or `uni`) attribute, a standard CDL database can include several variants of a unified character, possibly with different strokes or components. Some kind of "variant selectors" could in this way be given very precise meanings. Whether to associate certain variants with certain locales is another question, perhaps best decided separately by implementations for particular locales; an international standard would simply specify which variant selectors correspond to which CDL descriptions.

14. Here is a minimal DTD (Document Type Definition); it omits a few optional or experimental attributes that were not mentioned in this document:

```
<?xml encoding="UTF-8"?>
<!ELEMENT cdl-list (cdl)+>
<!ELEMENT cdl (comp|stroke)+>
<!ELEMENT comp EMPTY>
<!ELEMENT stroke EMPTY>
<!ATTLIST cdl
     char            CDATA  #IMPLIED
     uni             CDATA  #IMPLIED
     variant         CDATA  #IMPLIED
     points          CDATA  #IMPLIED
>
<!ATTLIST comp
     char            CDATA  #IMPLIED
     uni             CDATA  #IMPLIED
     variant         CDATA  #IMPLIED
     points          CDATA  #IMPLIED
>
<!ATTLIST stroke
     type            CDATA           #IMPLIED
     points          CDATA           #IMPLIED
     head  (cut|corner|vertical)     #IMPLIED
     tail  (cut|long)                #IMPLIED
>
```

Character Description Language (CDL):
The Set of Basic CJK Unified Stroke Types

Source: Tom Bishop <tbishop@wenlin.com> and Richard Cook <rscook@unicode.org>
Status: Expert Contribution
Date: May 23, 2004, 4:28 pm [revision of 20040523:16:27:07]
Action: For UTC and IRG consideration

This document is part of the *Specification of CDL* outlined in L2/03-404. See also L2/03-387 for additional discussion and examples of CDL usage. For information on the CDL specification and its implementations, see <http://www.wenlin.com/cdl/>.

1) The Set of Types

Table 1 below lists the set of 39 Basic Stroke Types currently implemented in the CDL descriptions of more than 40,000 ISO/IEC 10646 “CJK Unified Ideographs” (including all BMP, and 12,000 SIP forms).

The eleven headers *A*..*K* in Table 1 are as follows:

- ***A*** Sequential numbering [1..39] of all current types;
- ***B*** Numeric index for the 5 札 *zhá* types [1..5], with alphabetic sub-types [a..z];
- ***C*** Total number of 折 *zhé* ‘transitional bends’ (+1 = number of segments) in the type;
- ***D*** Total number of control points currently implemented for the type;
- ***E*** Frequency (non-recursive) of this type in current descriptions, as a percentage of total;
- ***F*** Glyph exemplifying the type in isolation (outside of compounds);
- ***G*** Provisional assignment of an ISO/IEC 10646 *UCS Scalar Value* for each exemplar in F, or *PUA* (Private Use Area) for unencoded forms;
- ***H*** Name of the type in Han characters;
- ***I*** Romanization in *pīnyīn* of H;
- ***J*** Abbreviation for the *pīnyīn* name of the type in I (acronymic, except for 39);
- ***K*** Notes on the type, including structural analysis (not necessarily tied to the actual implementation), unified variants of the type, examples of usage in compounds, and cross-references to similar types.

Table 1: Set of Basic CJK Unified Stroke Types

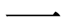
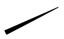
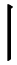
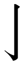
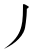
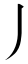
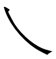
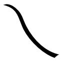
#	札	折	點	分	體	碼	名	拼	縮	注
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*
1	1a	0	2	26.87		U+4e00	橫	*héng*	**h**	horizontal; as in 三十卅; or as 一 in 七乇戈; *cp*. ㇀ **t**
2	1b	0	2	03.45		*PUA*	提	*tí*	**t**	一 **h** + taper; 3rd stroke of 扌 as in 地; stroke 5 of 虫
3	2a	0	2	15.77		U+4e28	竖	*shù*	**s**	vertical; as in 中卜上; or as / in 五 and 丑
4	2b	1	3	01.13		U+4e85	竖钩	*shù-gōu*	**sg**	丨 **s** + left hook; as in 小丁亍才; *cp*. ㇙ **st**
5	3a	0	2	12.54		U+4e3f	撇	*piě*	**p**	falling to left, not very curved; as in 八彳行; *cp*. ㇓ **wp** and 丿 **sp**
6	3b	0	2	03.95		*PUA*	弯撇	*wān-piě*	**wp**	curve + ㇒ **p**; as in 大; *cp*. ㇒ **p** and 丿 **sp**
7	3c	1	3	03.22		*PUA*	竖撇	*shù-piě*	**sp**	丨 **s** + ㇓ **wp**; as in 厂; *cp*. ㇓ **wp** and ㇒ **p**
8	4a	0	2	09.59		U+4e36	点	*diǎn*	**d**	taper + clockwise curve; as in 主; sometimes to left, as ㇒ 1st in 火
9	4b	0	2	03.52		*PUA*	捺	*nà*	**n**	falling right counter-clockwise curve; as in 人; *cp*. ㇒ **p** and ◡ **pn**
10	4c	0	3	00.03		*PUA*	点捺	*diǎn-nà*	**dn**	丶 **d** + ㇏ **n**; 2nd stroke in 入, only in 入 (=⼊) and its compounds, *e.g.* 仚

Table 1: Set of Basic CJK Unified Stroke Types

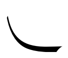
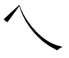
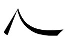
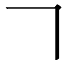
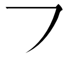
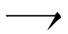
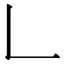
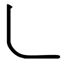
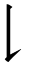
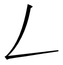
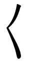
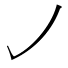
#	札	折	點	分	體	碼	名	拼	縮	注
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*
11	4d	1	3	00.43		*PUA*	平捺	*píng-nà*	**pn**	㇏ **n** + 一 **h**; last stroke in 是走廴; *cp*. ㇏ **n** and ㇄ **sw**
12	4e	1	3	00.11		U+4e40	提捺	*tí-nà*	**tn**	㇀ **t** + ㇏ **n**; last stroke in 夂 (=夊); or as ㇝ in 八入吏; *cp*. ⌣ **tpn**
13	4f	1	4	00.08		*PUA*	提平捺	*tí-píng-nà*	**tpn**	㇀ **t** + ⌣ **pn**; last stroke in 辶之; *cp*. 乀 **tn**
14	5a	1	3	03.28		U+200cd	橫折	*héng-zhé*	**hz**	一 **h** + 丨 **s**; 2nd stroke in 口; *cp*. ㇆ **hzg**
15	5b	1	3	00.90		*PUA*	橫撇	*héng-piě*	**hp**	一 **h** + ㇓ **p**; 1 in 又; 4 in 今; *cp*. ㇖ **hg**
16	5c	1	3	01.36		U+4e5b	横钩	*héng-gōu*	**hg**	一 **h** + left hook; 2 in 冖写; *cp*. ㇇ **hp**
17	5d	1	3	02.54		U+200ca	竖折	*shù-zhé*	**sz**	丨 **s** + 一 **h**; as in 山; or as ㇜ (㇒ **s** + 一 **h**) in 乐牛东互; *cp*. ㇜ **pz**
18	5e	1	4	00.17		*PUA*	竖弯	*shù-wān*	**sw**	丨 **s** + ⌣ **pn**; stroke 4 in 四
19	5f	1	3	01.36		U+2010c	竖提	*shù-tí*	**st**	丨 **s** + ㇀ **t** (right hook); as in 民辰; *cp*. 亅 **sg**
20	5g	1	3	00.51		*PUA*	撇折	*piě-zhé*	**pz**	㇓ **p** + ㇀ **t**; 3 in 公; stroke 1 in 厶; *cp*. ㇗ **sz**
21	5h	1	3	00.11		U+21fe8	撇点	*piě-diǎn*	**pd**	㇓ **p** + 丶 **d**; stroke 1 in 女巜巛𢀖
22	5i	1	3	00.00		*PUA*	撇钩	*piě-gōu*	**pg**	㇓ **p** + left-rising hook; as in 乂; *cp*. ㇓ **p**

Table 1: Set of Basic CJK Unified Stroke Types

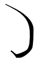
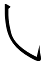
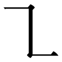
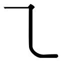
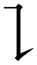
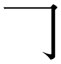
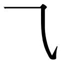
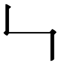
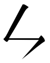
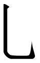
#	札	折	點	分	體	碼	名	拼	縮	注
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*
23	5j	1	4	00.24		*PUA*	弯钩	*wān-gōu*	**wg**	curving ㇚ **sg**; 3 in 豕, 6 in 家
24	5k	1	3	01.81		*PUA*	斜钩	*xié-gōu*	**xg**	㇏ **n** + up hook; 5 in 我; 2 in 弋; *cp.* ㇟ **swg**
25	5l	2	4	00.14		*PUA*	横折折	*héng-zhé-zhé*	**hzz**	㇐ **h** + ㇗ **sz** or ㇕ **hz** + ㇐ **h**; 2nd stroke in 凹; *cp.* ㇍ **hzw**, ㇈ **hzwg**
26	5m	2	5	00.03		*PUA*	横折弯	*héng-zhé-wān*	**hzw**	㇐ **h** + ㇄ **sw**; 2 in 朵殳; *cp.* ㇈ **hzwg**
27	5n	2	4	00.18		*PUA*	横折提	*héng-zhé-tí*	**hzt**	㇐ **h** + ㇙ **st**; 2 in 讠, as in 记鳩; *cp.* ㇅ **hzz**, ㇍ **hzw**
28	5o	2	4	02.22		U+200cc	横折钩	*héng-zhé-gōu*	**hzg**	㇐ **h** + ㇚ **sg**; 2 in 月丹; or as 𠃌 in 勺万; or as ㇆ in 也乜; *cp.* ㇕ **hz**
29	5p	2	4	00.28		U+2e84	横斜钩	*héng-xié-gōu*	**hxg**	㇐ **h** + ㇂ **xg**; 1st stroke in 飞卂; also in 凬气; *cp.* ㇈ **hzwg**
30	5q	2	4	00.44		U+200d1	竖折折	*shù-zhé-zhé*	**szz**	㇑ **s** + ㇕ **hz** or ㇗ + ㇑; as 4 in 亞, 6 in 鼎, 11 in 龍; *cp.* ㄣ **szp**, ㇉ **szzg**
31	5r	2	4	00.11		*PUA*	竖折撇	*shù-zhé-piě*	**szp**	㇒ **s** + ㇖ **hg** / ㇇ **hp**; as in 专呉; or as ㄣ in 艮詪[illegible]冄; *cp.* ㇉ **szzg**
32	5s	2	5	01.84		U+4e5a	竖弯钩	*shù-wān-gōu*	**swg**	㇄ **sw** + up hook; as in 儿礼心; *cp.* ㇄ **sw**

Table 1: Set of Basic CJK Unified Stroke Types

#	札	折	點	分	體	碼	名	拼	縮	注
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*
33	5t	3	5	00.06	㇎	*PUA*	横折折折	*héng-zhé-zhé-zhé*	**hzzz**	㇕ **hz** + ㇕ **hz**; 4th stroke in 凸凸; *cp.* ㇋ **hzzp**
34	5u	3	5	00.09	㇋	*PUA*	横折折撇	*héng-zhé-zhé-piě*	**hzzp**	㇖ **hg** + ㇇ **hp**; 1 in 廴建; 2 in 及; *cp.* ㇎ **hzzz**, ㇡ **hzzzg**
35	5v	3	6	00.60	乙	U+4e59	横折弯钩	*héng-zhé-wān-gōu*	**hzwg**	㇐ **h** + ㇟ **swg**; stroke 19 in 亁; or as ㇈ stroke 2 in 九几風; *cp.* ㇍ **hzw**, ㇅ **hzz**, ㇠ **hxg**
36	5w	3	6	00.03	㇌	*PUA*	横撇弯钩	*héng-piě-wān-gōu*	**hpwg**	㇖ **hg** + ㇁ **wg**; 1 in 阝队; *cp.* ㇡ **hzzzg**
37	5x	3	5	00.92	㇉	*PUA*	竖折折钩	*shù-zhé-zhé-gōu*	**szzg**	/ **s** + ㇆ **hzg**; 2 in 马丂; *cp.* ㇞ **szz**, ㇜ **szp**
38	5y	4	6	00.11	㇡	U+2010e	横折折折钩	*héng-zhé-zhé-gōu*	**hzzzg**	㇖ **hg** + ㇆ **hzg**; 1 in 乃仍; *cp.* ㇌ **hpwg**, ㇎ **hzzz**, ㇋ **hzzp**
39	5z	1	2	00.06	〇	U+3007	圈	*quān*	**o**	circle; bottom of 캉흥창; points are for bounding rectangle

2) Analysis of the set of Unified Basic Stroke Types

Table 2 below presents multi-dimensional feature analysis of the set of basic types. This analysis is given in terms of basic *segments* and transitional *junctures* between segments, and in terms of *vertical* (X), *horizontal* (Y), and *curvature* (Z) dimensions. For each of the *X*, *Y*, *Z* dimensions, the *directionality* of the stroke is indicated as follows:

X => **lr** 'left-to-right', **rl** 'right-to-left, **0** 'zero lateral movement';

Y => **tb** 'top-to-bottom', **bt** 'bottom-to-top', **0** 'zero longitudinal movement';

Z => **cw** 'clockwise', **ccw** 'counter-clockwise', **0** 'zero curvature'.

The total number of segments (T) for a given type may be written as T = *C* + 1, where *C* is equal to the number of junctures (column *C*). Each type is described with T elements in each of the *X*, *Y* and *Z* columns, where "+" indicates the juncture. Junctures are of two types, *curved* (gradual) and *sharp* (corner), and all curved junctures are associated with curvature of at least one of the conjoined segments. The relation between the number of transitions (column *C*) and the number of points (column *D*) is *D* = *sharp* + (*curved* * 2) + 2; when *C* = 0, *D* = 2. Elements separated by " | " indicate unified variants (parenthesized for T > 1), and column *F* includes several examples of such unifications (*i.e.* 2a, 4a, 4e, 5d, 5o, 5r, also given with examples in the notes in column K of Table 1). A trailing "+" in column *Z* indicates additional curvature, differentiating two pairs of types (3a,b and 4e,f).

See column K of Table 1 for analysis of the complex types into basic segments. The set of 7 basic segmental elements (those with zero transitions) is as follows:

㇐ **h**, ㇀ **t**, ㇑ **s**, ㇒ **p**, ノ **wp**, ㇔ **d**, ㇏ **n**

This set may be reduced by 2 to a set of 5, by applying transformations to the **t** (+taper) and **wp** (+curve) types, relative to base types **h** and **p**, respectively.

Note that combination of segments, basic or not, always results in a number of transitions equal to one less than the number of combined segments. So, for example, the 乀 **pn** stroke has 1 transition (it is composed of ㇏ **n** + ㇐ **h**), while ㇝ **tpn** also has 1 transition (from ㇀ **t** to 乀 **pn**) rather than 2 (㇀ **t** + ㇏ **n** + ㇐ **h**), which would ignore the higher level grouping for 乀 **pn** (= ㇏ **n** + ㇐ **h**). Similarly, it should perhaps be emphasized that stroke count for the each of the 39 types is always 1, no matter how many junctures.

Future treatment of the 3 撇 *piě* types ㇒ **p**, ノ **wp**, and ㇓ **sp** might involve unification, using a variable number of control points, though these are currently distinct in the implementation (note that there is at present only one encoded *piě* type, U+4e3f). Other unifications within the overall set might be possible as well, *e.g.* ㇏ **n** with 乀 **pn**, and ㇝ **tn** with ㇝ **tpn**. The set of 39 given here does however seem to have general validity and wide acceptance, in terms of modern orthographic practices, especially as evident in the representative forms appearing in modern typography, and in the ISO/IEC 10646 codecharts. Refinements to the set of types will likely involve additions needed to accommodate very rare forms.

Table 2: Analysis of the set of Unified Basic Stroke Types

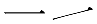
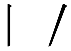
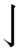
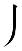
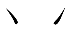
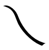
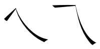
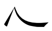
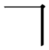
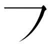
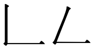
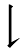
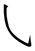
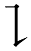
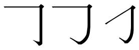
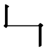
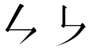
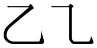
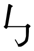
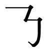
札	折	點	縮	體	橫	竖	弯
B	*C*	*D*	*J*	*F*	***X***	***Y***	***Z***
1a	0	2	**`h`**		lr	0|bt	0
1b	0	2	**`t`**		lr	bt	0
2a	0	2	**`s`**		0|rl	tb	0
2b	1	3	**`sg`**		0+lr	tb+bt	0+(0|cw)
3a	0	2	**`p`**		rl	tb	cw
3b	0	2	**`wp`**		rl	tb	cw+
3c	1	3	**`sp`**		0+rl	tb+tb	0+cw
4a	0	2	**`d`**		lr|rl	tb	cw
4b	0	2	**`n`**		lr	tb	ccw
4c	1	3	**`dn`**		lr+lr	tb+tb	cw+ccw
4d	1	3	**`pn`**		lr+lr	tb+0	ccw+0
4e	1	3	**`tn`**		lr+lr	(bt|0)+tb	0+ccw
4f	1	4	**`tpn`**		lr+lr	bt+tb	0+ccw+
5a	1	3	**`hz`**		lr+0	0+tb	0+0
5b	1	3	**`hp`**		lr+rl	0+tb	0+cw
5c	1	3	**`hg`**		lr+rl	0+tb	0+0
5d	1	3	**`sz`**		(0|rl)+lr	tb+0	0+0
5e	1	4	**`sw`**		0+lr	tb+0	0+ccw
5f	1	3	**`st`**		0+lr	tb+bt	0+0
5g	1	3	**`pz`**		rl+lr	tb+bt	cw+0

Table 2: Analysis of the set of Unified Basic Stroke Types

札	折	點	縮	體	橫	竖	弯
B	*C*	*D*	*J*	*F*	***X***	***Y***	***Z***
5h	1	3	**pd**		rl+lr	tb+tb	cw+cw
5i	1	3	**pg**		rl+rl	tb+bt	cw+0
5j	1	4	**wg**		lr+rl	tb+bt	cw+cw
5k	1	3	**xg**		lr+0	tb+bt	ccw+0
5*l*	2	4	**hzz**		lr+0+lr	0+tb+0	0+0+0
5m	2	5	**hzw**		lr+0+lr	0+tb+0	0+0+ccw
5n	2	4	**hzt**		lr+0+lr	0+tb+bt	0+0+0
5o	2	4	**hzg**		lr+(0|rl)+rl	(0|bt)+tb+bt	0+(0|cw)+0
5p	2	4	**hxg**		lr+lr+(0|lr)	0+tb+bt	0+ccw+(0|ccw)
5q	2	4	**szz**		0+lr+0	tb+0+tb	0+0+0
5r	2	4	**szp**		(0|rl)+lr+rl	tb+0+tb	0+0+(0|cw)
5s	2	5	**swg**		0+lr+0	tb+0+bt	0+ccw+0
5t	3	5	**hzzz**		lr+0+lr+0	0+tb+0+tb	0+0+0+0
5u	3	5	**hzzp**		lr+rl+lr+rl	0+tb+0+tb	0+(0|cw)+0+cw
5v	3	6	**hzwg**		lr+(0|rl)+lr+0	0+tb+0+bt	0+(0|ccw)+ccw+0
5w	3	6	**hpwg**		lr+rl+lr+(0|rl)	0+tb+tb+bt	0+(0|cw)+cw+(0|cw)
5x	3	5	**szzg**		rl+lr+rl+(0|lr)	tb+0+tb+bt	0+0+cw+(0|cw)
5y	4	5	**hzzzg**		lr+rl+lr+rl+(0|rl)	0+tb+0+tb+bt	0+(0|cw)+0+cw+(0|cw)
5z	4	2	**o**		lr+rl+rl+lr	tb+tb+bt+bt	cw+cw+cw+cw ?

SC2/WG2 N3063

ISO/IEC JTC 1/SC 2/WG 2 N ______

ISO/IEC JTC 1/SC 2/WG 2/IRG N 1180

ISO/IEC JTC1/SC2/WG2/IRG

Ideographic Rapporteur Group (IRG)

TITLE:	Proposed additions to the CJK Strokes block of the UCS
SOURCE:	IRG Rapporteur
STATUS:	Submission from the IRG to WG2
DISTRIBUTION:	Members of ISO/IEC JTC1/SC2/WG2
DATE:	2006-4-3
REFERENCE:	WG2 N2807R, N2808R, IRG N1174
ATTACHMENTS:	IRG N1181 (“Summary for Stroke submission”)

Summary

This document proposes the addition of twenty new CJK Strokes to the CJK Strokes block of the UCS.

WG2 resolution M45.34 (N2754R) expanded the scope of IRG work to include CJK Strokes. The CJK Strokes block (U+31CO..U+31EF) now contains a total of sixteen CJK Strokes (U+31C0..U+31CF), derived from HKSCS (ISO/IEC 10646:2003/Amd.1).

The IRG formed ad-hoc groups to complete the repertoire of common CJK Strokes, and finalized the repertoire in the IRG#25 meeting held in Berkeley, California, U.S.A. The twenty new CJK Strokes are proposed for code points in the range (U+31D0..U+31E3). Information on naming conventions and collation is also provided.

List of the proposed characters, character names and code positions

	31C	31D	31E
0	㇀	㇐	㇠
1	㇁	㇑	㇡
2	㇂	㇒	㇢
3	㇃	㇓	㇣
4	㇄	㇔	
5	㇅	㇕	
6	㇆	㇖	
7	㇇	㇗	
8	㇈	㇘	
9	㇉	㇙	
A	㇊	㇚	
B	㇋	㇛	
C	㇌	㇜	
D	㇍	㇝	
E	㇎	㇞	
F	㇏	㇟	

31C0	CJK STROKE T
31C1	CJK STROKE WG
31C2	CJK STROKE XG
31C3	CJK STROKE BXG
31C4	CJK STROKE SW
31C5	CJK STROKE HZZ
31C6	CJK STROKE HZG
31C7	CJK STROKE HP
31C8	CJK STROKE HZWG
31C9	CJK STROKE SZWG
31CA	CJK STROKE HZT
31CB	CJK STROKE HZZP
31CC	CJK STROKE HPWG
31CD	CJK STROKE HZW
31CE	CJK STROKE HZZZ
31CF	CJK STROKE N
31D0	CJK STROKE H
31D1	CJK STROKE S
31D2	CJK STROKE P
31D3	CJK STROKE SP
31D4	CJK STROKE D
31D5	CJK STROKE HZ
31D6	CJK STROKE HG
31D7	CJK STROKE SZ
31D8	CJK STROKE SWZ
31D9	CJK STROKE ST
31DA	CJK STROKE SG
31DB	CJK STROKE PD
31DC	CJK STROKE PZ
31DD	CJK STROKE TN
31DE	CJK STROKE SZZ
31DF	CJK STROKE SWG
31E0	CJK STROKE HXWG
31E1	CJK STROKE HZZZG
31E2	CJK STROKE PG
31E3	CJK STROKE Q

Stroke Type Naming Conventions

Stroke types have traditionally been named using combinations of Han characters meaning ‘horizontal’, ‘vertical’, ‘bent’, ‘curved’, etc.

These characters are conveniently represented by single letters of the alphabet, which are abbreviations for the Mandarin pronunciations of these characters. The following table shows each abbreviation, in alphabetical order, followed by the corresponding Hanyu Pinyin, Han character, and approximate meaning in English. In all cases, the final alphabetic string in the character name may be converted to the sequence of Han characters for that name, simply by substituting a Latin letter in the following list for the corresponding Han character.

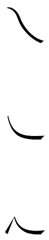
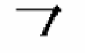
For example, "HZZZG" becomes 横折折折鉤.

B	biǎn	扁	‘flat’	Q	quān	圈	‘circle’
D	diǎn	點	‘dot’	S	shù	豎	‘vertical’
G	gōu	鉤	‘hook’	T	tí	提	‘rising’
H	héng	横	‘horizontal’	W	wān	彎	‘curved’
N	nà	捺	‘right-falling’	X	xié	斜	‘slant’
P	piě	撇	‘left-falling’	Z	zhé	折	‘bent’

Variants and usage examples of CJK Strokes

The following table illustrates typical variant forms of each stroke type and Han characters encoded in ISO/IEC 10646 as examples of usage.

Stroke	Name	Variants	Usage examples
㇐	**H** (横)		一，三，丁，丞，丈，世，不，上，十，卅，七
㇀ U+31C0	**T** (提)		冰，淋，治，冶，冽，暴，氾，录，地，虫
㇑	**S** (豎)		丩，中，串，讧，乍，上，五，丑

㇚	**SG** (竪鉤)		爭，事，求，水
㇒	**P** (撇)		乂，爻，禾，毛，乏，乖，采，衣，八，行
㇓	**SP** (竪撇)		乃，月，用，齊，几，人，班，大
㇔	**D** (點)		丸，义，永，冰，凡，丹，主，求，火，刃
㇏ U+31CF	**N** (捺)		大，人，天，入，走，边，廷，尺
㇝	**TN** (提捺)		尐，八，入，廻
㇕	**HZ** (横折)		口，囗，田，品，吕，申，甲，圆，㠯
㇇ U+31C7	**HP** (横撇)		又，双，叒，今
㇖	**HG** (横鉤)		疋，了，予，矛，子，字，疏，写，冖
㇗	**SZ** (竪折)		断，继，山，互，彙，牙，乐，东

㇘	**SWZ** (豎彎左)		肅嘯蕭簫
㇄ U+31C4	**SW** (豎彎)		區，亡，妄，四
㇙	**ST** (豎提)		食，良，艮，很，狠，鄉，民
㇜	**PZ** (撇折)	㇜	弘，公，翁
㇛	**PD** (撇點)		巡，獵，災，甾，女，巛
㇢	**PG** (撇鉤)		㐅
㇁ U+31C1	**WG** (彎鉤)		狐，嶽，貓，家，逐
㇂ U+31C2	**XG** (斜鉤)		戈，弋，戰，我
㇃ U+31C3	**BXG** (扁斜鉤)		心，必，沁，惢，蕊
㇅ U+31C5	**HZZ** (橫折折)		卍
㇍ U+31CD	**HZW** (橫折彎)		殳，投，朵
㇊ U+31CA	**HZT** (橫折提)		讠，计，鳩

㇆ U+31C6	**HZG** (橫折鉤)	𠃌	羽，习，包，匀，葡，用，青，甫，勺，月，也，乜
㇈ U+31C8	**HZWG** (橫折彎鉤)	⺄	飞，风，瘋，九，几，气，虱
㇞	**SZZ** (豎折折)	ㄥ ㄣ	亞，鼎，卐，吳，专，設，㞋，奏
㇟	**SWG** (豎彎鉤)		乱，己，已，巳
㇎ U+31CE	**HZZZ** (橫折折折)		凸
㇋ U+31CB	**HZZP** (橫折折撇)		建，及
乙	**HXWG** (橫斜彎鉤)		乙，氹，乞
㇌ U+31CC	**HPWG** (橫撇彎鉤)		阝，队，邮
㇉ U+31C9	**SZWG** (豎折彎鉤)		号，亏，弓，强，丐，马，丂
㇡	**HZZZG** (橫折折折鉤)		乃，孕，仍
○	**Q** (圈)		〇，召，古，咎

Unicode Character Properties

All proposed characters should have the following Unicode properties so that they match those of the encoded sixteen CJK Strokes.
So;0;ON;;;;;N;;;;;

Collation of CJK Strokes

Because the HKSCS strokes were added to the block before the more complete repertoire was developed, the strokes in the block are out of order, in terms of the traditional stroke classification. CJK Strokes are divided into five broad types, in Mandarin called the five 札 zhá types, since the Han character "札" exhibits each of the types: ㇐㇑㇒㇔乙 (héng shù piě diǎn zhé). The collation order proposed here assigns each stroke to one and only one of the five types, and further sub-classifies each stroke according to its specific features (see IRG N987). The following is an XML representation of the collation data for the proposed CJK Strokes block, in accordance with CLDR guidelines.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ldml SYSTEM "http://www.unicode.org/cldr/dtd/1.4/ldml.dtd">
<ldml>
    <identity>
        <version number="$Revision: 1.00 $"/>
        <generation date="$Date: 2005/12/12 12:12:12 $"/>
        <language type="root" />   <!-- since this is a pan-language ordering of CJK -->
    </identity>
        <collations> <!-- removing the valid sublocales until a complete list is available -->
            <collation type="IRG-Strokes" > <!-- or whatever name would be most descriptive -->
                <rules>
                <!-- 札 zhá type 1 -->
                <!-- h; t -->
                <pc> ㇐㇀</pc>
                <!--札 zhá type 2-->
                <!-- s; sg -->
                <pc>㇑㇚</pc>
                <!--札 zhá type 3-->
                <!-- p; sp -->
                <pc>㇒㇓</pc>
                <!--札 zhá type 4-->
                <!-- d; n; tn -->
              <pc>㇔ ㇏㇝</pc>
                <!--札 zhá type 5-->
                <!-- hz; hp; hg; sz; swz; sw; st; pz; pd; pg; wg; xg; bxg; hzz; hzw; hzt; hzg; hxg; szz; swg; hzzz; hzzp;
hxwg; hpwg; szwg; hzzzg; q -->
                <pc>㇕㇇㇖㇗㇘㇄㇙㇜㇛㇢㇁㇂㇃㇅㇍㇊㇆㇈㇞㇟㇎㇋㇠㇌㇉㇡㇣</pc>
                </rules>
             </collation   >
          </collations>
</ldml>
```

Rationale for proposed repertoire

The repertoire of strokes proposed for addition to the existing CJK Strokes block is derived from the stroke types occurring in representative forms of currently encoded UCS Ideographs. All of these proposed strokes are currently missing from the CJK Strokes block. Representative forms of some proposed CJK Strokes are similar in appearance to representative forms of some single-stroke CJK Ideographs currently encoded in various UCS blocks (URO, Ext A, Ext B, Kangxi Radicals, CJK Radicals Supplement). However, single-stroke CJK Ideographs do not have the properties of CJK Strokes, and single-stroke CJK Ideographs may in some cases exhibit a range of variation in their representative glyphs which conflates necessary distinctions for the CJK Strokes block. For example, the proposed CJK Strokes ㇒ "CJK STROKE P" (U+31D2) vs. ㇓ "CJK STROKE SP" (U+31D3) are conflated in the representative forms currently used for ⼃"KANGXI RADICAL SLASH" (U+2F03) and 丿 "CJK UNIFIED IDEOGRAPHS-4E3F" (U+4E3F). The proposed strokes complete the set of most commonly seen stroke types. The precise definition of the CJK Strokes block and clear differentiation of it from the blocks of CJK Ideographs and Radicals serves an essential purpose in the indexing and collation of encoded and unencoded CJK Ideographs and Radicals. It is anticipated that some very rare strokes may be proposed for future addition to this set.

ATTACHEMENTS: IRG N1181 “Summary for Stroke submission”

A. Administrative

1. **Title:** Summary for Stroke submission
2. Requester's name: *IRG Rapporteur*
3. Requester type (Member body/Liaison/Individual contribution): *IRG*
4. Submission date: *2006-4-3*
5. Requester's reference (if applicable): *IRG N1174*
6. Choose one of the following:
   This is a complete proposal: *Yes*
   (or) More information will be provided later:

B. Technical – General

1. Choose one of the following:
   a. This proposal is for a new script (set of characters):
   Proposed name of script:
   b. The proposal is for addition of character(s) to an existing block: *Yes*
   Name of the existing block: *CJK Strokes*
2. Number of characters in proposal: *20*
3. Proposed category (select one from below - see section 2.2 of P&P document):
   A-Contemporary ____ B.1-Specialized (small collection) *X* B.2-Specialized (large collection) ____
   C-Major extinct ____ D-Attested extinct ____ E-Minor extinct ____
   F-Archaic Hieroglyphic or Ideographic ____ G-Obscure or questionable usage symbols ____
4. Proposed Level of Implementation (1, 2 or 3) (see Annex K in P&P document): *1*
   Is a rationale provided for the choice? *No*
   If Yes, reference:
5. Is a repertoire including character names provided? *Yes*
   a. If YES, are the names in accordance with the “character naming guidelines”
   in Annex L of P&P document? *Yes*
   b. Are the character shapes attached in a legible form suitable for review? *Yes*
6. Who will provide the appropriate computerized font (ordered preference: True Type, or PostScript format) for publishing the standard? *Beijing Founder Electronic Co.,Ltd.*
   If available now, identify source(s) for the font (include address, e-mail, ftp-site, etc.) and indicate the tools used: *9, No. 5 street, Shangdi, Information Industry Base, Haidian District, Beijing 100085, P.R. China; yjh@founder.com.cn; http://www.foundertype.com*
7. References:
   a. Are references (to other character sets, dictionaries, descriptive texts etc.) provided? *Yes (IRG N1174)*
   b. Are published examples of use (such as samples from newspapers, magazines, or other sources) of proposed characters attached? *Yes; see “Variants and usage examples of CJK Strokes” section in IRG N1180.*
8. Special encoding issues:
   Does the proposal address other aspects of character data processing (if applicable) such as input, presentation, sorting, searching, indexing, transliteration etc. (if yes please enclose information)? *Yes*

9. Additional Information:

Submitters are invited to provide any additional information about Properties of the proposed Character(s) or Script that will assist in correct understanding of and correct linguistic processing of the proposed character(s) or script. Examples of such properties are: Casing information, Numeric information, Currency information, Display behaviour information such as line breaks, widths etc., Combining behaviour, Spacing behaviour, Directional behaviour, Default Collation behaviour, relevance in Mark Up contexts, Compatibility equivalence and other Unicode normalization related information. See the Unicode standard at http://www.unicode.org for such information on other scripts. Also see http://www.unicode.org/Public/UNIDATA/UCD.html and associated Unicode Technical Reports for information needed for consideration by the Unicode Technical Committee for inclusion in the Unicode Standard.

C. Technical - Justification

1. Has this proposal for addition of character(s) been submitted before? *No*
   If YES explain
2. Has contact been made to members of the user community (for example: National Body, user groups of the script or characters, other experts, etc.)? *Yes*
   If YES, with whom? *IRG and Ideographic experts*
   If YES, available relevant documents: *IRG N987, IRG N1081, IRG N1086A, IRG N1096, IRG*

N1097, IRG N1138

3. Information on the user community for the proposed characters (for example:
size, demographics, information technology use, or publishing use) is included? Users of Han characters

Reference:

4. The context of use for the proposed characters (type of use; common or rare) common

Reference:

5. Are the proposed characters in current use by the user community? Yes

If YES, where? Reference: Yes; see the representative glyphs in the published ISO/IEC 10646 Standard.

6. After giving due considerations to the principles in the P&P document must the proposed characters be entirely in the BMP? Yes

If YES, is a rationale provided? Yes

If YES, reference: IRG N1180

7. Should the proposed characters be kept together in a contiguous range (rather than being scattered)? Yes

8. Can any of the proposed characters be considered a presentation form of an existing character or character sequence? No

If YES, is a rationale for its inclusion provided?

If YES, reference:

9. Can any of the proposed characters be encoded using a composed character sequence of either existing characters or other proposed characters? No

If YES, is a rationale for its inclusion provided?

If YES, reference:

10. Can any of the proposed character(s) be considered to be similar (in appearance or function) to an existing character? Yes

If YES, is a rationale for its inclusion provided? Yes

If YES, reference: See “Rationale for proposed repertoire” section in IRG N1180

11. Does the proposal include use of combining characters and/or use of composite sequences? No

If YES, is a rationale for such use provided?

If YES, reference:

Is a list of composite sequences and their corresponding glyph images (graphic symbols) provided?

If YES, reference:

12. Does the proposal contain characters with any special properties such as control function or similar semantics? No

If YES, describe in detail (include attachment if necessary)

13. Does the proposal contain any Ideographic compatibility character(s)? No

If YES, is the equivalent corresponding unified ideographic character(s) identified?

If YES, reference: