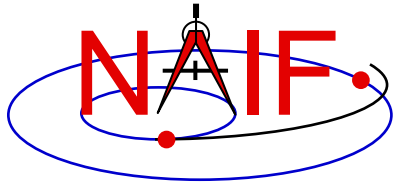




Navigation and Ancillary Information Facility

Other Useful Functions

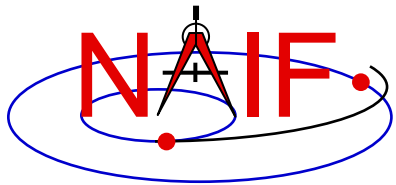
January 2018



Topics

Navigation and Ancillary Information Facility

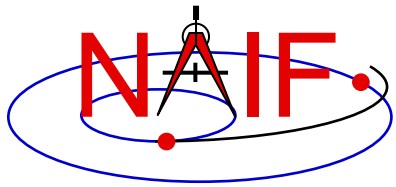
- **Overview**
- **Language-specific status**
- **File Operations**
- **String Manipulation**
- **Searching, Sorting and Other Array Manipulations**
- **Windows**
- **Symbol Tables**
- **Sets and Cells**
- **Constants and Unit Conversion**
- **Numerical Functions**



Overview

Navigation and Ancillary Information Facility

- **The routines described in this tutorial originated in the Fortran version of the the SPICE Toolkit.**
- **Many, but not all, of these routines have implementations in the C, IDL, and MATLAB Toolkits.**
- **The descriptions include a language “identifier” or set of identifiers prefixed to the routine’s name to indicate which Toolkit language(s) include that routine.**
 - [F] available in Fortran (SPICELIB)
 - [C] available in C (CSPICE)
 - [I] available in IDL (Icy)
 - [M] available in MATLAB (Mice)
- **NAIF adds interfaces to the CSPICE, Icy and Mice Toolkits as needed or when requested by a customer.**
- **CSPICE, Icy and Mice do not need all of the functionality implemented in the Fortran Toolkit.**
- **NAIF does not attempt to keep track of which functions are implemented in 3rd party Python toolkits such as SpiceyPy.**



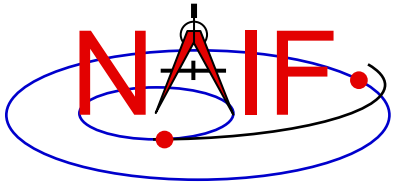
Text I/O (1)

Navigation and Ancillary Information Facility

- **Text files provide a simple, human readable mechanism for sharing data.**
- **The Toolkit contains several utility routines to assist with the creation and parsing of text, and with the reading and writing of text files.**
 - **[F,C] RDTEXT: read a line of text from a text file***
 - **[F] TOSTDO: write a line of text to standard output**
 - **[F,C] PROMPT: display a prompt, wait for and return user's response**
 - **[F] TXTOPN: open a new text file returning a logical unit**
 - **[F] WRITLN: write a line of text to the file attached to a logical unit.**



* The text file must be in native text format for your computer



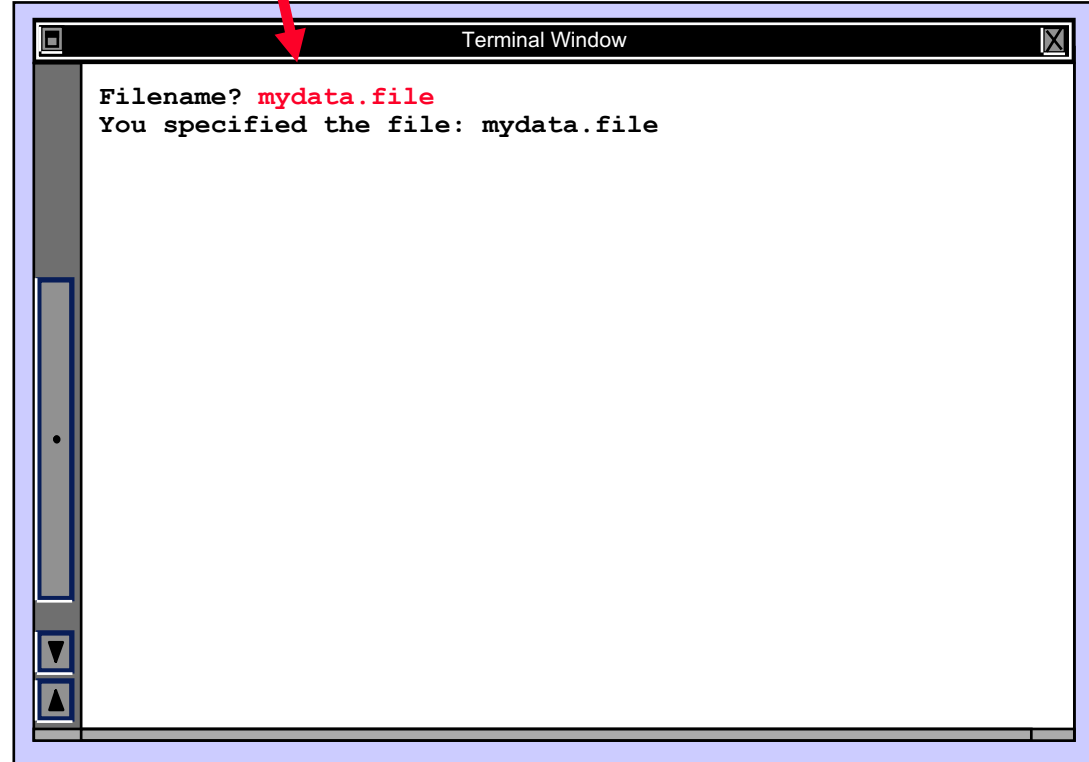
Text I/O (2)

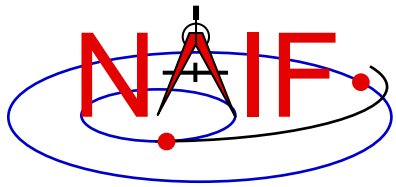
Navigation and Ancillary Information Facility

```
CALL PROMPT ( 'Filename? ', NAME )  
CALL TOSTDO ( 'You specified the file: '// NAME )
```

Now that we have the filename, read
and process its contents

```
CALL RDTEXT ( NAME, LINE, EOF )  
  
DO WHILE ( .NOT. EOF )  
    process the line just read  
    CALL RDTEXT ( NAME, LINE, EOF )  
  
END DO
```

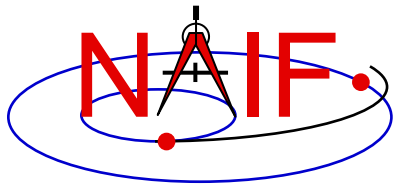




File Operations

Navigation and Ancillary Information Facility

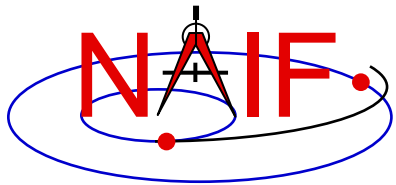
- **Logical unit management - Fortran specific**
 - [F] RESLUN: (reserve logical unit) prohibits SPICE systems from using specified units.
 - [F] FRELUN: (free logical unit) places “reserved” units back into service for SPICE.
 - [F] GETLUN: (get logical unit) locates an unused, unreserved logical unit.
- **Determining whether or not a file exists**
 - [F,C,I] EXISTS
- **Deleting an existing file**
 - [F] DELFIL



String Manipulation - Parsing (1)

Navigation and Ancillary Information Facility

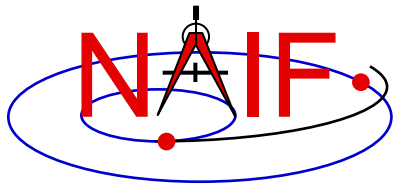
- **Breaking apart a list**
 - [F,C,I] LPARSE: parses a list of items delimited by a single character.
 - [F,C] LPARSM: parses a list of items separated by multiple delimiters.
 - [F] NEXTWD: returns the next word in a given character string.
 - [F] NTHWD: returns the nth word in a string and the location of the word in the string.
 - [F,C] KXTRCT: extracts a substring starting with a keyword.
- **Removing unwanted parts of a string**
 - [F,C,I] CMPRSS: compresses a character string by removing instances of more than N consecutive occurrences of a specified character.
 - [F] ASTRIP: removes a set ASCII characters from a string.
 - [F] REMSUB: removes a substring from a string.



String Manipulation - Parsing (2)

Navigation and Ancillary Information Facility

- **Locating substrings**
 - [F] LTRIM, RTRIM: return the location of the leftmost or rightmost non-blank character.
 - [F,C] POS, CPOS, POSR, CPOSR, NCPOS, NCPOSR: locate substring or member of specified character set searching forward or backward.
- **Pattern matching**
 - [F,C,I] MATCHI: matches a string against a wildcard template, case insensitive.
 - [F,C,I] MATCHW: matches a string against a wildcard template, case sensitive.
- **Extracting numeric and time data**
 - [F] NPARSD, NPARSI, DXTRCT, TPARTV
 - [F,C,I] PRSDP, PRSINT, TPARSE
- **Heavy duty parsing**
 - [F] SCANIT



String Manipulation - Parsing (3)

Navigation and Ancillary Information Facility

`'a dog, a cat, and a cow'`

`lparse or
lparsm`

Split on a comma

`'a dog'`

`'a cat'`

`'and a cow'`

`'Remove extra spaces'`

`cmprss`

`'Remove extra spaces'`

`'Green eggs and ham'`

`'the cat in the hat'`

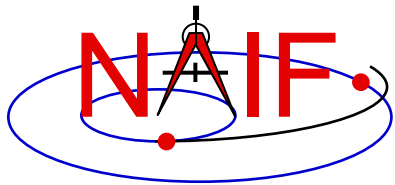
`'how the grinch stole Christmas'`

`matchi(*g*)`

`'green eggs and ham'`

`'how the grinch stole Christmas'`

Match any string containing a 'g'



String Manipulation - Creating (1)

Navigation and Ancillary Information Facility

- **Fill in the “Blank”**

- **[F,C] REPMC: Replace a marker with a character string.**

```
CALL REPMC ( 'The file was: #', '#', 'foo.bar', OUT )
```

OUT has the value “The file was: foo.bar”

- **[F,C] REPMI: Replace a marker with an integer.**

```
CALL REPMI ( 'The value is: #', '#', 7, OUT )
```

OUT has the value “The value is: 7”

- **[F,C] REPMD: Replace a marker with a double precision number.**

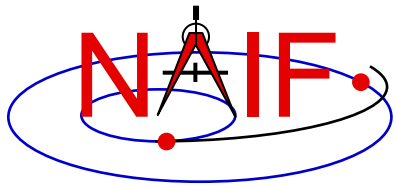
```
CALL REPMD ( 'The value is: #', '#', 3.141592654D0, 10, OUT )
```

OUT has the value “The value is: 3.141592654E+00”

- **[F,C] REPMOT: Replace a marker with the text representation of an ordinal number.**

```
CALL REPMOT ( 'It was the # term.', '#', 'L', 2, OUT )
```

OUT has the value “It was the second term.”



String Manipulation - Creating (2)

Navigation and Ancillary Information Facility

- **Fill in the “Blank” (cont.)**

- **[F,C] REPMCT: Replace a marker with the text representation of a cardinal number.**

```
CALL REPMCT ( 'Hit # errors.', '#', 6, 'L', OUT )
```

OUT becomes ‘Hit six errors.’

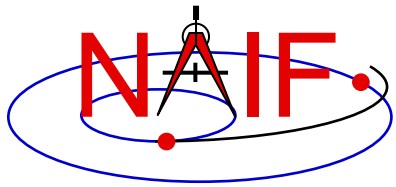
- **Numeric Formatting**

- **[F] DPFMT: Using a format template, create a formatted string that represents a double precision number**

```
CALL DPFMT ( PI(), 'xxx.yyyy', OUT )
```

OUT becomes ‘ 3.1416’

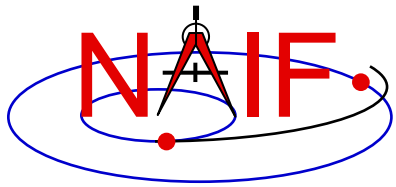
- **[F] DPSTR, INTSTR, INTTXT, INTORD**



String Manipulation - Creating (3)

Navigation and Ancillary Information Facility

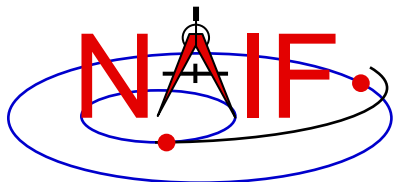
- **Time formatting**
 - [F,C,I,M] TPICTR: Given a sample time string, create a time format picture suitable for use by the routine TIMOUT.
 - [F,C,I,M] TIMOUT: Converts an input epoch to a character string formatted to the specifications of a user's format picture.
- **Changing case**
 - [F,C,I] UCASE: Convert all characters in string to uppercase.
 - [F,C,I] LCASE: Convert all characters in string to lowercase.
- **Building strings**
 - [F] SUFFIX: add a suffix to a string
 - [F] PREFIX: add a prefix to a string
 - [F] LJUCRS: left-justify, upper-case and compress



Searching, Sorting and Other Array Manipulations (1)

Navigation and Ancillary Information Facility

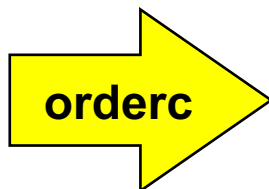
- **Sorting arrays**
 - [F,C] SHELLC, SHELLI, SHELLD, ORDERI, ORDERC, ORDERD, REORDC, REORDI, REORDD, REORDL
- **Searching ordered arrays**
 - [F,C] BSRCHC, BSRCHI, BSRCHD, LSTLEC, LSTLEI, LSTLED, LSTLTC, LSTLTI, LSTLTD, BSCHOI
- **Searching unordered arrays**
 - [F,C] ISRCHC, ISRCHI, ISRCHD, ESRCHC
- **Moving portions of arrays**
 - [F] CYCLAC, CYCLAD, CYCLAI
- **Inserting and removing array elements**
 - [F] INSLAC, INSLAD, INSLAI, REMLAC, REMLAD, REMLAI



Searching, Sorting and Other Array Manipulations (2)

Navigation and Ancillary Information Facility

| Body | A.U. ¹ |
|---------|-------------------|
| sun | 00.0 |
| mercury | 00.455 |
| venus | 00.720 |
| earth | 00.983 |
| mars | 01.531 |
| jupiter | 05.440 |
| saturn | 09.107 |
| uranus | 20.74 |
| neptune | 30.091 |
| pluto | 31.052 |



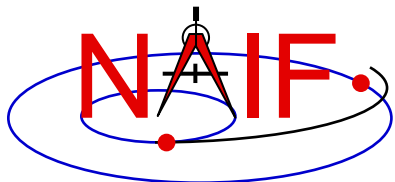
04
06
05
02
09
10
07
01
08
03



| Sorted Body | A.U. ¹ |
|-------------|-------------------|
| earth | 00.983 |
| jupiter | 05.440 |
| mars | 01.531 |
| mercury | 00.455 |
| neptune | 30.091 |
| pluto | 31.052 |
| saturn | 09.107 |
| sun | 00.000 |
| uranus | 20.74 |
| venus | 00.720 |

Vector of "Body" indices representing the list sorted in alphabetical order.

¹ Distance in A.U. at Jan 01, 2006.



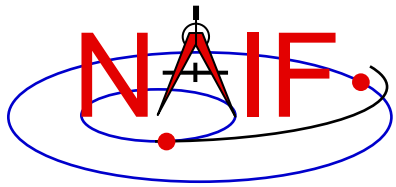
Windows

Navigation and Ancillary Information Facility

- A SPICE window is a list of disjoint intervals arranged in ascending order.

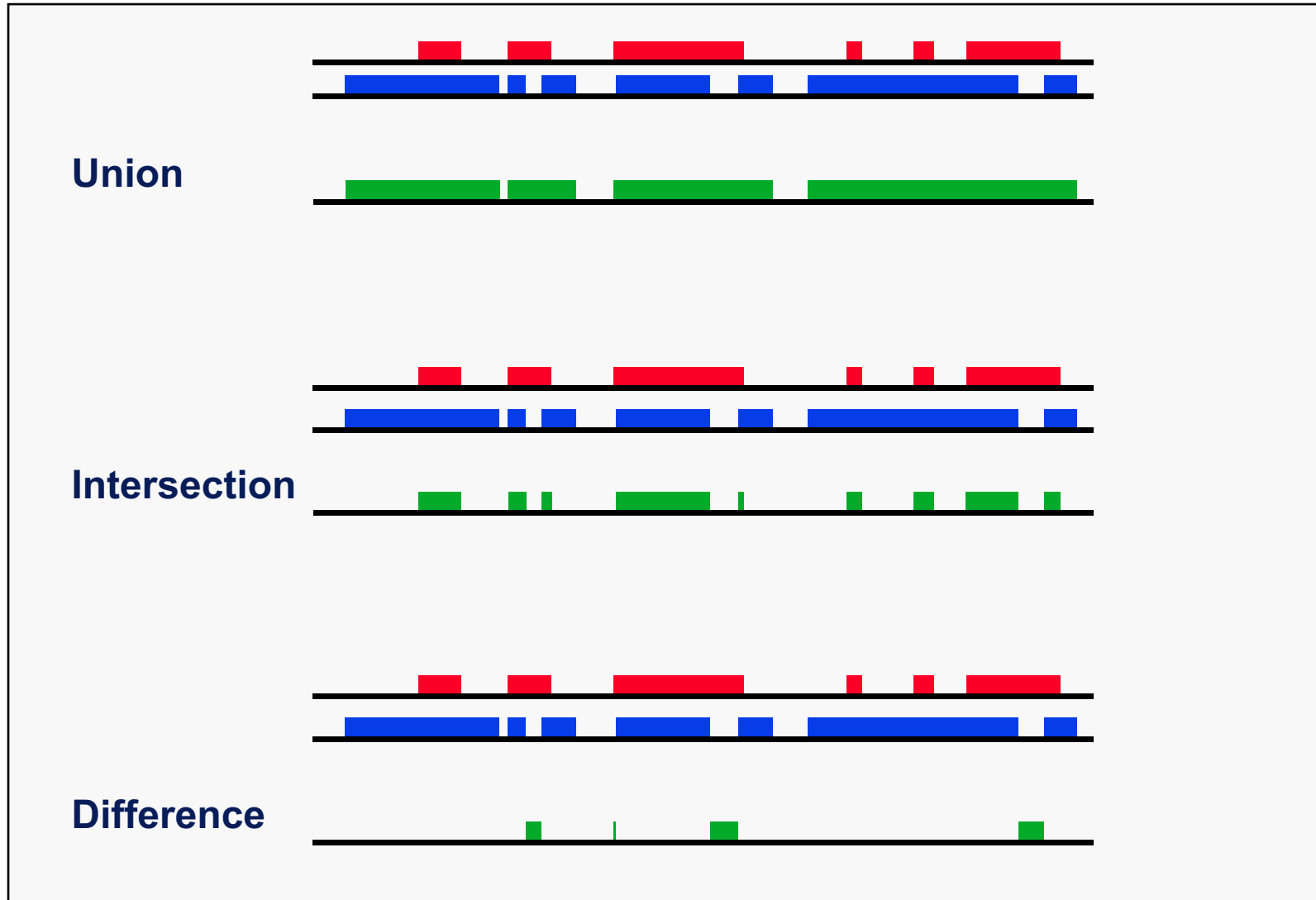


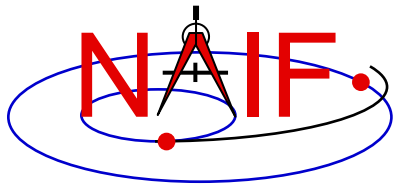
- An interval is specified by a pair of double precision numbers, with the second greater than or equal to the first.
- The Toolkit contains a family of routines for creating windows and performing “set arithmetic” on them.
- SPICE windows are frequently used to specify intervals of time when some set of user constraints are satisfied.
 - Let window *NotBehind* contain intervals of time when Cassini is not behind Saturn as seen from earth.
 - Let window *Goldstone* contain intervals of time when Cassini is above the Goldstone horizon.
 - Cassini can be tracked from Goldstone during the intersection of these two windows ($Track = NotBehind * Goldstone$).
- See *windows.req* for more information.



Windows Math

Navigation and Ancillary Information Facility

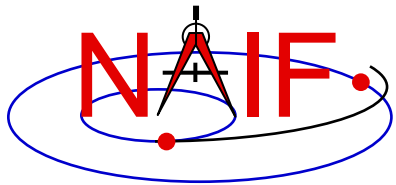




Symbol Tables

Navigation and Ancillary Information Facility

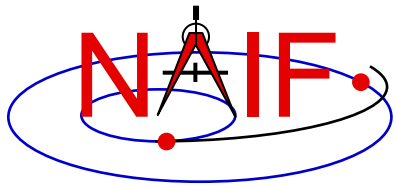
- **SPICELIB (Fortran) supports the use of associative arrays/hashtables through the use of an abstract data type called symbol tables.**
 - These are used to associate a set of names with collections of associated values.
 - Values associated with a name are exclusively character, exclusively integer or exclusively double precision.
 - Routines to manipulate a symbol table have the form **SY***<T>** where **<T>** is the data type of the values (C, D, or I).
- **Operations include:**
 - Insert a symbol
 - Remove a symbol
 - Push/Pop a value onto/off of the list of values associated with a symbol
 - Fetch/Sort values associated with a symbol
- **See *symbols.req* for more information.**



Sets and Cells (1)

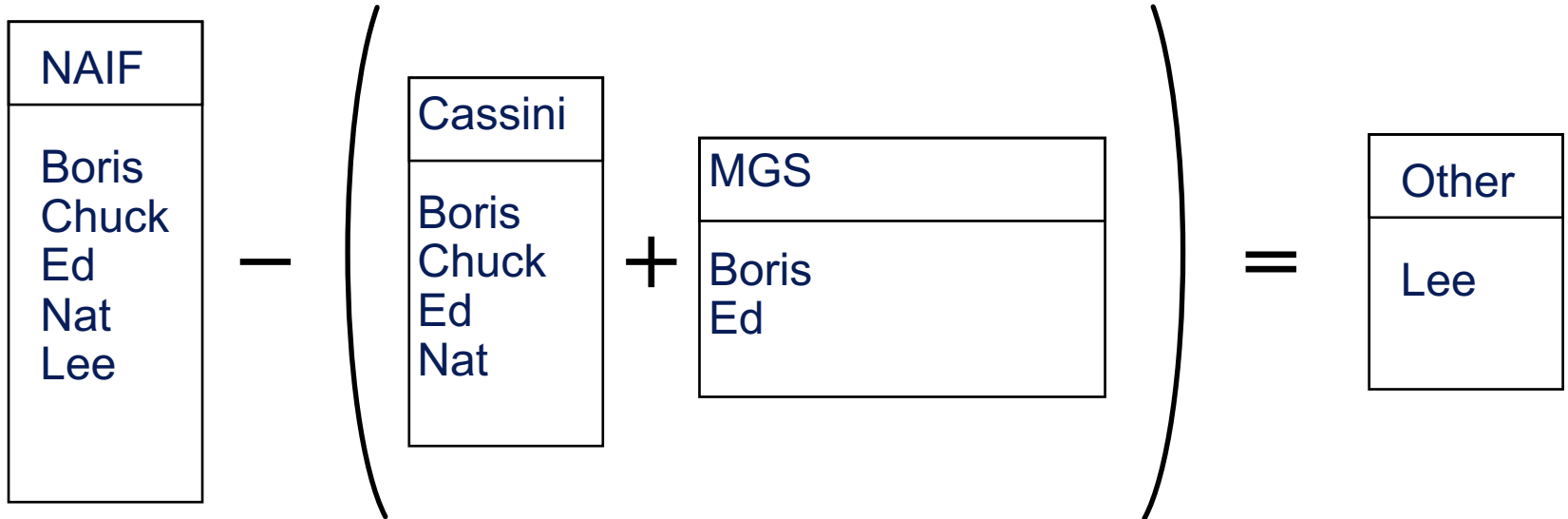
Navigation and Ancillary Information Facility

- **Cells are arrays that “know” how many addresses are available for use and how many are currently used.**
 - Routines that use cells typically have simpler interfaces than routines that use arrays.
 - See *cells.req* for more information.
- **Sets are cells that contain no duplicate elements and whose elements are ordered in ascending order.**
 - Two Sets can be: intersected, unioned, differenced, differenced symmetrically (union - intersection)
 - See *sets.req* for more information.
- **Language support for sets and cells**
 - Double Precision, Integer, and Character string cell types are supported in the Fortran and C Toolkits.
 - Double Precision and Integer cell types are supported in the IDL Toolkits.
 - Sets and cells aren’t currently needed in the MATLAB Toolkits since MATLAB supports set math.

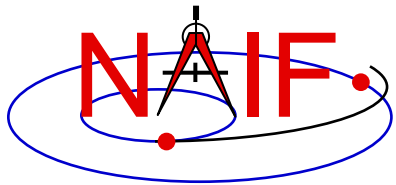


Sets and Cells (2)

Navigation and Ancillary Information Facility



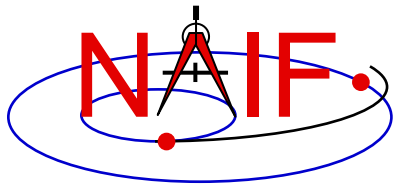
```
CALL UNIONC ( CASSINI, MGS, PROJECTS )
CALL DIFFC ( NAIF, PROJECTS, OTHER )
```



Constants and Unit Conversion

Navigation and Ancillary Information Facility

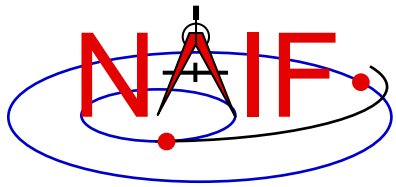
- **Constants are implemented in the Toolkit as functions.**
 - Thus the changing of a constant by NAIF requires only relinking by the Toolkit user—not recompiling.
 - » Users should NOT change constant functions in the Toolkit.
- **System Constants**
 - [F,C,I,M] DPMIN, DPMAX, INTMIN, INTMAX
- **Numeric Constants**
 - [F,C,I,M] PI, HALFPI, TWOPI, RPD (radians/degree), DPR(degrees/radian)
- **Physical Constants**
 - [F,C,I,M] CLIGHT, SPD, TYEAR, JYEAR
- **Epochs**
 - [F,C,I,M] J2000, J1950, J1900, J2100, B1900, B1950
- **Simple Conversion of Units**
 - [F,C,I,M] CONVRT



Numerical Functions (1)

Navigation and Ancillary Information Facility

- **Several routines are provided to assist with numeric computations and comparisons.**
- **Functions**
 - [F] DCBRT: cube root
 - **Hyperbolic Functions:**
 - » [F] DACOSH, DATANH
 - **Polynomial Interpolation and Evaluation:**
 - » [F] LGRESP, LGRINT, LGRIND, POLYDS, HRMESP, HRMINT
 - **Chebyshev Polynomial Evaluation:**
 - » [F] CHBDER, CHBVAL, CHBINT, CHBIGR



Numerical Functions (2)

Navigation and Ancillary Information Facility

- **Numerical Decisions**
 - Same or opposite sign (Boolean):
 - » [F] SMSGND, SMSGNI, OPSGND, OPSGNI
 - Force a value into a range (bracket):
 - » [F,C] BRCKTD, BRCKTI
 - Determine parity of integers (Boolean):
 - » [F] ODD, EVEN
 - Truncate conditionally:
 - » [F] EXACT
- **Arithmetic**
 - Greatest common divisor:
 - » [F] GCD
 - Positive remainder:
 - » [F] RMAINI, RMAIND