EZMAP/EZMAPA, A MAP-DRAWING PACKAGE

David J. Kennison
NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000
email: kennison@ncar.ucar.edu (128.117.64.4)

EZMAP/EZMAPA, A MAP-DRAWING PACKAGE

David J. Kennison
NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000
email: kennison@ncar.ucar.edu (128.117.64.4)

1.0 INTRODUCTION

This document describes an NCAR Graphics package named EZMAP, which allows one to plot maps of the earth according to any of ten different projections, with parallels, meridians, and continental, international, and/or U.S. state outlines. The origin and orientation of the projection may be selected by the user. Points on the earth defined by latitude and longitude are mapped to points in a projection plane, referred to as the u/v plane. A rectangular perimeter whose sides are parallel to the u and v axes is chosen; material within that perimeter (or an inscribed elliptical perimeter) is mapped to the plane of the plotter (referred to as the x/y plane) for plotting. The u and v axes are parallel to the x and y axes, respectively.

Also described in this document are routines from a package originally named EZMAPA, which, among other things, allows the user to produce solid-filled maps of the earth.

Principal sections of this document are as follows:

- The rest of this "**INTRODUCTION**" gives an overview of EZMAP/EZMAPA, with short descriptions of all of the available subroutines.
- The section "**PROJECTIONS**" describes the projections of the earth that may be used.
- The section "**ROUTINES**" gives detailed descriptions of all available subroutines.
- The section "**INTERNAL PARAMETERS**" describes all of the internal parameters of the package; each internal parameter affects some aspect of the behavior of one or more of the routines in the package.
- The section "**ERROR HANDLING**" describes all the conditions that are recognized as errors and what can be done to recover from an error.
- The section "**EXAMPLES**" describes examples that are available to illustrate the use of the package; some users may wish to obtain the code for these examples and run them so as to have them available for reference while reading further.
- The section "**AREA IDENTIFIERS**" lists all the area identifiers that are used by EZMAP/EZMAPA. User knowledge of these is necessary for some purposes.
- The section named "**MISCELLANY**" gives information that will be of interest to a few users.

1.1 Routines Used to Draw a Simple Map

To draw a simple map (one not involving solid fill of areas), as directed by the current values of EZMAP's internal parameters, one need only execute the single FORTRAN statement "CALL MAPDRW". Thus, the principal EZMAP routine is this:

- **MAPDRW** - Draws a complete simple map.

All that MAPDRW does is call four lower-level routines. In some situations, the user may wish to call these routines directly; they are as follows (in the order in which they are called by MAPDRW):

- **MAPINT** - Initializes. MAPINT must be called at least once before calling any routine that depends on mapping lat/lon coordinates into u/v coordinates. After one or more of MAPPOS, MAPROJ, and MAPSET has been called, MAPINT must be called again. MAPINT does the call to the SPPS routine SET that defines the mapping from the u/v (projection) plane to the x/y (plotter) plane.
- **MAPGRD** - Draws selected parallels and meridians.
- **MAPLBL** - Labels the international date line, the equator, the Greenwich meridian, and the poles, and draws the perimeter.
- **MAPLOT** - Draws selected geographic outlines.

1.2 Routines Used to Change the Values of Internal Parameters

The following routines are called to change the values of internal parameters of EZMAP, and thus change the behavior of other EZMAP routines:

- **MAPPOS** - Determines what portion of the plotter frame is to be used.
- **MAPROJ** - Determines the projection to be used.
- **MAPSET** - Determines what portion of the u/v plane is to be viewed.
- **MAPSTC** (or **MPSETC**) - Sets a parameter value of type CHARACTER.
- **MAPSTI** (or **MPSETI**) - Sets a parameter value of type INTEGER.
- **MAPSTL** (or **MPSETL**) - Sets a parameter value of type LOGICAL.
- **MAPSTR** (or **MPSETR**) - Sets a parameter value of type REAL.

1.3 Routines Used to Retrieve the Values of Internal Parameters

The following routines are used to retrieve the current values of EZMAP parameters:

- **MAPGTC** (or **MPGETC**) - Gets a parameter value of type CHARACTER.
- **MAPGTI** (or **MPGETI**) - Gets a parameter value of type INTEGER.
- **MAPGTL** (or **MPGETL**) - Gets a parameter value of type LOGICAL.
- **MAPGTR** (or **MPGETR**) - Gets a parameter value of type REAL.

1.4 Routines Used to Save and Restore Internal Parameters

To save/restore the current values of the internal parameters of EZMAP, use the following:

- **MAPSAV** - Saves the values (by writing a record on a user-specified unit).
- **MAPRST** - Restores saved values (by reading a record from a user-specified unit).

1.5 Routines Used to Draw Objects on a Map

To draw objects on the map, use the following routines:

- **MAPTRA** - Computes the u/v coordinates of a point from its latitude and longitude. If the point is unprojectable or its projection lies outside the current perimeter, a special value is returned to signal this.
- **MAPTRN** - Computes the u/v coordinates of a point from its latitude and longitude. If the point is unprojectable, a special value is returned to signal this, but no check is made for a projected value being outside the perimeter.
- **MAPFST** - Does a "pen-up" move defining the start of a line to be projected and drawn. The line is defined by a series of lat/lon coordinates.
- **MAPVEC** - Does a "pen-down" move defining the continuation of a line to be projected and drawn. The line is defined by a series of lat/lon coordinates.
- **MAPIT** - Does "pen-up" or "pen-down" moves. This routine is called by MAPFST and MAPVEC.
- **MAPIQ** - Signals the end of a string of calls to MAPIT and causes its buffers to be flushed.
- **MAPGCI** - Given the latitudes and longitudes of two points on the surface of the globe, this routine returns the latitudes and longitudes of a specified number of points along the great circle route joining them.

1.6 Routines Used to Do Inverse Transformations

The following routine was added to EZMAP early in 1992:

- **MAPTRI** - Computes the latitude and longitude of a point from its u/v coordinates. If the point is outside the boundary of the map, a special value is returned.

The example named "mpex10" shows one of the ways in which this routine may be used; it draws what is essentially a colored contour plot of a data field defined on the surface of the globe, using an orthographic projection.

1.7 Routines Used to Draw Solid-Filled Maps (EZMAPA)

In late 1986 or early 1987, a package of routines was written allowing a user to draw solid-filled maps of the globe. This package was named EZMAPA and was first advertised in the NCAR Graphics User's Guide (Version 2.00), published in August, 1987. Conceptually, the routines in this package are really part of EZMAP; they use the same common blocks and many of the same underlying low-level routines and they are affected by the same set of internal parameters as the routines in EZMAP proper. The routines of EZMAPA will be described in this document; to use them effectively, it will be necessary to understand also the package AREAS, which is described in a separate document. The EZMAPA routines are as follows:

- **MAPBLA** - Adds boundary lines to an existing area map. Routines in the package AREAS may then be used to process that area map in various ways. (Example: drawing a map of Europe with each country in a different color.)
- **MAPBLM** - Draws boundary lines “masked” by an existing area map. (Example: drawing these lines only where they do not overlay CONPACK labels.)
- **MAPGRM** - Draws selected parallels and meridians “masked” by an existing area map. (Example: drawing these lines over water, but not over land.)
- **MAPITA** and **MAPIQA** - Add to an area map the projections of arbitrary lines defined by lat/lon coordinates of points on the surface of the globe. MAPBLA uses these routines to add boundary lines to an area map; they may be called directly by the user to add his/her own set of boundary lines to the area map.
- **MAPITM** and **MAPIQM** - Draw, masked by an area map, the projections of arbitrary lines defined by lat/lon coordinates of points on the surface of the globe. MAPGRM uses these routines to draw masked lines of latitude and longitude; they may be called directly by the user to draw other masked lines.
- **MAPACI** - A function which, given the “area identifier” for a particular area defined by the boundaries in one of the EZMAP outline datasets, returns a suggested color index for that area; it is guaranteed that, if the suggested color indices are used, no two areas having a boundary in common will have the same color.

1.8 Miscellaneous Other Routines

The following EZMAP routines are used for the purposes stated:

- **MAPEOD** - This routine is called by the EZMAP routine MAPLOT and the EZMAPA routines MAPBLA and MAPBLM; in each case, it is called once for each segment in the outline dataset. The user may supply a version which examines the segment to see if it ought to be used and, if not, to delete it. This can be used (for example) to reduce the clutter in northern Canada.
- **MAPRS** - Re-executes the “CALL SET” done during the last call to MAPINT. This is useful when there has been an intervening call to a utility that calls SET. It is quite common for a background drawn by EZMAP to be placed in a flash buffer (as created by the package “GFLASH”). When the contents of the flash buffer are copied to the metafile being created, if it is desired to draw something on the EZMAP background, MAPRS may first have to be called to ensure that the correct SET call is in effect.

- **MAPUSR** - This routine is called by various EZMAP routines just before and just after drawing parts of the map. By default, grid lines are drawn using software dashed lines and geographical outlines are drawn using either solid lines or dotted lines. The dash pattern used for the grid lines, the flag which says whether outlines are solid or dotted, and the color indices of various parts of the map are all user-settable parameters, but more complete control of color indices, spot size, dash pattern, etc., may be achieved by supplying one's own version of MAPUSR; a user version may be as complicated as is required to achieve a desired effect.
- **SUPMAP** - This is a version of the routine from which EZMAP grew. It allows one to draw a complete map with a single, rather lengthy, call. The routine **SUPCON**, which is the old analogue of MAPTRN, is also implemented.

All of the routines mentioned above are described in detail in the section “ROUTINES”, below.

2.0 PROJECTIONS

EZMAP offers ten different projections, in three groups: conical, azimuthal, and cylindrical.

2.1 Conical Projections

Conical projections map the surface of the earth onto a cone which is either tangent to the earth along a single circle or intersects it along two circles. The cone is cut along some line passing through its vertex and opened up onto a flat surface. The only conical projection offered by EZMAP is the Lambert conformal with two standard parallels (becoming a simple conic with one standard parallel if the two parallels are made equal). The cone intersects the earth along two user-specified standard parallels (lines of latitude), which would normally both be in the Northern Hemisphere or in the Southern Hemisphere; the cone is cut along the line opposite a user-specified central meridian (line of longitude) and laid flat on the u/v plane with either the North Pole or the South Pole (as implied by the standard parallels) at the origin.

If LAT1 and LAT2 are the latitudes of the two standard parallels and LAT1 is not equal to LAT2, the so-called “cone constant” is given by the formula

```
              LOG(COS(LAT1))-LOG(COS(LAT2))
CONE = ------------------------------------------
       LOG(TAN(45-S*LAT1/2))-LOG(TAN(45-S*LAT2/2))
```

where “S” is +1 in the Northern Hemisphere and -1 in the Southern Hemisphere. If LAT1 equals LAT2, then

```
CONE = COS(90-S*LAT1)
```

The value of CONE is between 0 and 1; CONE*360 is the angular separation between the edges of the cut after the cone is opened onto the plane, as measured across the surface of the flattened cone. If (RLAT,RLON) is a point to be projected, then the formulas

```
R = (TAN(45-S*RLAT/2))**CONE
U = R*SIN(CONE*(RLON-CLON))
V = -S*R*COS(CONE*(RLON-CLON))
```

where CLON is the longitude of the central meridian, give the coordinates of the projected point in the u/v plane.

The globe is projected onto the entire u/v plane minus a wedge with its apex at the origin. This projection is normally used to depict mid-latitude regions of limited extent, for which it is relatively distortion-free. It has the property of preserving angles.

See the example named “mpex01”.

2.2 Azimuthal Projections

Azimuthal projections map the earth (or a single hemisphere of it) onto a u/v plane whose origin touches the earth at the user-specified point (PLAT,PLON). The image may be rotated by a user-specified angle ROTA. The azimuthal projections are generated as follows:

- Step 1: Imagine that the earth is placed behind the u/v plane so that the point at latitude 0 and longitude 0 just touches the plane at the point (0,0), the North Pole is at the top, and the South Pole is at the bottom.
- Step 2: Rotate the earth about its polar axis until the v axis is tangent to the meridian identified by PLON (and that meridian is therefore closest to you).
- Step 3: Rotate the earth, tilting one of the poles directly toward you and the other pole directly away from you, until the point (PLAT,PLON) is at the origin of the u/v plane.
- Step 4: Rotate the earth clockwise through the angle ROTA about a line perpendicular to the u/v plane passing through the point (0,0).
- Step 5: Using lines of projection emanating from a central point within or behind the earth (depending on the projection type), project geographical outlines, parallels, and meridians from the earth’s surface onto the u/v plane.
- Step 6: Set up linear scales along the u and v axes. The ranges of u and v depend on the projection type.
- Step 7: Draw a rectangular or elliptical portion of the resulting map.

If A is the angular separation, in degrees, of a point P, to be projected, from the point (PLAT,PLON), SINA is the sine of A, COSA is the cosine of A, and R is the linear distance of the projected point P' from the u/v origin, the various azimuthal projections may be described in terms of the relationship between A and R, as follows:

- Stereographic projection:
  ```
  R = TAN(A/2) = (1-COSA)/SINA
  ```

As A approaches 180 degrees, R approaches infinity. The entire globe is projected to the entire u/v plane. In practice, distortion becomes great beyond R=2, when A is approximately 127 degrees. The center of the projection is the point on the earth's surface opposite the point of tangency with the projection plane.

See the examples named "mpex02", "mpex04", "mpex05", and "mpex07".

- Orthographic projection:

```
R = SINA
```

Points for which A is greater than 90 degrees are treated as invisible. Thus, a hemisphere is projected inside a circle of radius 1. The center of the projection is at infinity; all projection lines are parallel to each other and perpendicular to the u/v plane.

See the example named "mpex05".

- Lambert equal-area projection:

```
R = 2*SINA/SQRT(2*(1+COSA))
```

As A approaches 180 degrees, R approaches 2. The entire globe is projected into a circle of radius 2.

See the example named "mpex05".

- Gnomonic projection:

```
R = TAN(A) = SINA/COSA
```

Points for which A is greater than 90 degrees are invisible. Thus, a hemisphere is projected to the entire u/v plane. In practice, distortion becomes great beyond R=2, when A is approximately 65 degrees. The center of this projection is the center of the earth.

See the example named "mpex05".

- Azimuthal equidistant projection:

```
R = A*pi/180
```

As A approaches 180 degrees, R approaches pi (3.1415926...). Thus, the entire globe is projected within a circle of radius pi.

See the example named "mpex05".

- Basic satellite-view projection:

```
R = SQRT(SA*SA-1)*SINA/(SA-COSA)
```

where "SA" is the distance from the center of the earth to a satellite above the point (PLAT,PLON), in multiples of the earth's radius. Points for which COSA is less than "1/SA" are invisible. The portion of the earth's surface which would be visible from the satellite is projected inside a circle of radius 1. The center of the projection is at the satellite's position. As the satellite moves further and further out, the basic satellite-view projection approaches the orthographic projection. See the examples named "mpex05" and "mpexfi".

The basic satellite-view projection gives a view of the earth as seen by a satellite camera that is looking straight down toward the center of the earth. Note that the model for the "camera" we're talking about here is a pin-hole camera, in which each point of an image is formed by a single straight line emanating from some point in the scene. Real cameras are more complicated than that: each point of an image is formed by focusing infinitely many rays emanating from a point in the scene (all the rays from the point that happen to pass through the lens of the camera). An image formed by a real camera is generally distorted in some fashion and cannot be expected to exactly match an image formed by our hypothetical pin-hole camera.

See the examples named “mpex05” and “mpexfi”.

- General satellite-view projection:

```
R = SOMETHING MESSY THAT I DON’T HAVE IN CLOSED FORM!
```

The user-settable parameters 'S1' and 'S2' affect the satellite-view projection as follows: S1 measures the angle between the line to the center of the earth and the line of sight of the satellite (the line to which the projection plane is perpendicular). The default value of S1 is zero, which gives the basic satellite view. When S1 is non-zero, S2 specifies the angle from the positive u axis of the basic satellite view counter-clockwise to the line OP, where O is the origin of the basic view and P is the projection (a single point), on the basic view, of the desired line of sight from the satellite. When S1 and S2 are used, the part of the earth projected is the same as for the basic view, but the part of the u/v plane covered by the projection may become the interior of an ellipse, the area “inside” a parabola, or the area “inside” one branch of a hyperbola; in each of these cases, the major axis of the curve defining the limb of the projection is at an angle S2 to the u axis.

Basically, 'S1' and 'S2' allow you to aim the satellite camera in any desired direction; however, getting exactly the view that you want is not completely trivial. The example “cpex10” demonstrates a technique that may be quite useful: if, in the call to MAPROJ that requests the satellite-view projection (with a first argument = 'SV' and second and third arguments PLAT and PLON specifying the position of the satellite), you use a value of ROTA (the final argument) such that the point you want to aim the satellite camera at is directly above the center of the basic view and you then use 'S1' non-zero and 'S2' = 90., you get a view with “outer space” at the top, the earth at the bottom, and the horizon line more or less horizontal. It is recommended that the satellite-view projection be used with a call to MAPSET in which the first argument JLTS = 'AN'; in that case, the other four arguments specify the field of view of the camera (as angular distances to the left, to the right, to the bottom, and to the top, of the frame). It is further recommended that you **not** use a first argument JLTS = 'MA' in such a call to MAPSET, since, in that case, you get a view showing the entire part of the globe that is visible from the satellite and distortion is likely to be extreme.

See the examples named “mpex06” and “cpex10”. The latter is actually a CONPACK example, but it demonstrates the aiming technique described in the previous paragraph.

2.3 Cylindrical Projections

Cylindrical projections map the surface of the earth onto a cylinder which is tangent to the earth along a great circle passing through the user-specified point (PLAT,PLON) and tilted at a user-specified angle ROTA. The cylinder is cut along a line parallel to its axis and unrolled onto the plane. The cylindrical projections are generated as follows:

- Step 1: Imagine that the earth is placed behind the u/v plane so that the point at latitude 0 and longitude 0 just touches the plane at the point (0,0), the North Pole is at the top, and the South Pole is at the bottom.

- Step 2: Rotate the earth about its polar axis until the v axis is tangent to the meridian identified by PLON (and that meridian is therefore closest to you).
- Step 3: Rotate the earth, tilting one of the poles directly toward you and the other pole directly away from you, until the point (PLAT,PLON) is at the origin of the u/v plane.
- Step 4: Rotate the earth clockwise through the angle ROTA about a line perpendicular to the u/v plane and passing through the point (0,0).
- Step 5: Wrap the u/v plane around the globe to form a cylinder, with the u axis touching the earth along a great circle.
- Step 6: Using a technique dependent on the projection type, project geographical outlines, parallels, and meridians outward from the earth's surface onto the cylinder.
- Step 7: Cut the cylinder along a line parallel to its axis and opposite the point (0,0).
- Step 8: Unwrap the cylinder again.
- Step 9: Set up linear scales along the u and v axes. The ranges of u and v depend on the projection type.
- Step 10: Draw a rectangular or elliptical portion of the resulting map.

What happens in step 6 above will be described for each of the three types of cylindrical projections provided by EZMAP in the simple case where PLAT, PLON, and ROTA are all zero. Let RLAT and RLON be the latitude and longitude, in degrees, of a point to be projected; RLON must lie between -180 and +180. (If PLAT, PLON, and/or ROTA are non-zero, one must substitute for RLAT and RLON a pseudo-latitude and a pseudo-longitude computed from the real latitude and longitude; this is left as an exercise for the devotee of spherical trigonometry.) The cylindrical projections may then be described as follows:

- Cylindrical Equidistant: U and V are computed using the equations

  ```
  U = RLON
  V = RLAT
  ```

  The entire globe is projected into a rectangle in the u/v plane. U ranges from -180 to +180, V from -90 to +90.

  See the example named "mpex05".

- Mercator: U and V are computed using the equations

  ```
  U = RLON*pi/180 (where pi=3.14159...)
  V = ALOG(COT(45-RLAT/2))
  ```

  The entire globe is projected into an infinite rectangle in the u/v plane. U ranges from -pi to +pi, V from -infinity to +infinity. In practice, distortion becomes unacceptable for latitudes within 5 degrees of the North or South Pole.

  See the examples named "mpex03", "mpex04", "mpex05", "mpex08", and "mpex09".

- Mollweide-type: The projection is not a true Mollweide. U and V are computed using the equations

  ```
  U = RLON/90
  V = COS(90-RLAT)
  ```

The entire surface of the globe is projected into an ellipse. U ranges from -2 to +2, V from -1 to +1.

See the example named “mpex05”.

3.0 ROUTINES

This section describes all of the user-callable routines in the packages EZMAP and EZMAPA. The descriptions are arranged in alphabetical order.

3.1 MAPACI (IAID)

The function MAPACI is normally used to pick colors for the areas created by the boundary lines added to an area map by a call to MAPBLA.

3.1.1 Usage

The FORTRAN statement

```
ICLR = MAPACI(IAID)
```

where IAID is the the area identifier for one of the areas created by the boundary lines in a selected outline dataset, gives ICLR an integer value between 1 and 7. This integer value will normally be used (in a manner determined by the user) to select a color index for that area. It is guaranteed that, if two areas have a portion of their boundaries of non-zero length in common, then MAPACI will return different values for the two areas. For a complete list of area identifiers and the values that MAPACI will return for each, see the section named “AREA IDENTIFIERS”.

See the example named “eezmpa”.

3.1.2 Arguments

IAID (an input expression of type INTEGER) is an area identifier for one of the areas defined by the set of boundary lines added to an area map by a call to MAPBLA.

3.2 MAPBLA (IAMA)

Adds to the area map in the array IAMA the set of boundary lines, of projected geographical entities, determined by the current state of the internal parameters of EZMAP.

3.2.1 Usage

Execute the statement

```
CALL MAPBLA (IAMA)
```

to add boundary lines to the area map in the array IAMA. The area map must previously have been initialized by calling the routine ARINAM, in the package AREAS, using a statement like “CALL ARINAM (IAMA,LAMA)”, where LAMA is the length of the array IAMA. The area map may subsequently be used in various ways; for example, one may call the AREAS routine ARSCAM to draw a solid-filled map.

One or two groups of boundary lines are added to the area map by a call to MAPBLA. The first, having group identifier 'G1' (default value 1), consists of a perimeter (either rectangular or elliptical, depending on the value of the internal parameter 'EL') and the set of projected boundary lines implied by the user’s selection of an outline dataset (some combination of continental, U.S., and world political outlines). For certain projections, a limb line may also be included.

If the parameter 'VS' has a value greater than zero, the group 'G2' is added to the area map; it consists of a copy of the perimeter and the limb line (if any) plus a set of vertical lines splitting the area inside the perimeter into 'VS' vertical strips. (By default, the value of 'VS' is 1.) The object of the group 'G2' is to split areas up, reducing the number of points required to define a typical area below the level at which some target hardware device begins to fail.

See the example named “eezmpa”.

3.2.2 Arguments

IAMA (an input/output array of type INTEGER) is the area map array to which boundary lines are to be added.

3.3 MAPBLM (IAMA,XCRA,YCRA,MCRA, . . .)

(The remaining arguments are IAAI, IAGI, NOGI, and ULPR.)

3.3.1 Usage

The FORTRAN statement

```
CALL MAPBLM (IAMA,XCRA,YCRA,MCRA,IAAI,IAGI,MNOG,ULPR)
```

is used to draw the set of boundary lines of projected geographical entities determined by the current state of the internal parameters of EZMAP, masked against the area map in the array IAMA.

MAPBLM is much like MAPLOT, except that the boundary lines are drawn using calls to MAPITM and MAPIQM, which does the masking of the lines against an area map and passes the pieces resulting from the masking process along to a user-provided line-drawing routine.

See the example named “cpex10”.

3.3.2 Arguments

IAMA (an input/output array of type INTEGER, dimensioned as specified in a call to the AREAS routine ARINAM) is the array containing the area map against which boundary lines are to be masked. The area map must have been initialized by a call to ARINAM; it should contain the edges required to create a desired effect. For example, an area map might be created that defines a region of interest, within which user data is available and within which boundary lines are to be drawn. For more details, see the reference document for the package named AREAS.

XCRA and **YCRA** (scratch arrays, dimensioned at least MCRA, of type REAL) are to be used by MAPBLM in calls to the AREAS routine ARDRLN; they will eventually be used in calls to the user-provided line-processing routine ULPR.

MCRA (an input expression of type INTEGER) is the dimension of the arrays XCRA and YCRA.

IAAI and **IAGI** (scratch arrays, dimensioned at least NOGI, of type INTEGER) are to be used by MAPBLM in calls to the AREAS routine ARDRLN; they will eventually be used in calls to the user-provided line-processing routine ULPR. The mnemonics stand for “Integer Array of Area Identifiers” and “Integer Array of Group Identifiers”, respectively.

NOGI (an input expression of type INTEGER) is the dimension of the arrays IAAI and IAGI. The mnemonic stands for “Number Of Group Identifiers (of edges in the area map)”, which determines the required dimension of IAAI and IAGI.

ULPR is the name of the user-supplied line-processing routine. It must be declared EXTERNAL in the routine that calls MAPBLM, so that the compiler and loader will know that it is the name of a routine to be called instead of a variable. The user routine ULPR will be called once for each piece of a boundary line resulting from the masking process; it may decide to draw (or to not draw) each such piece. ULPR will be called using a FORTRAN statement like

```
CALL ULPR (XCRA,YCRA,NCRA,IAAI,IAGI,NGPS)
```

where XCRA and YCRA are real arrays holding the normalized device coordinates of NCRA points defining a polyline which is part of some boundary line and IAAI and IAGI are integer arrays holding NGPS area-identifier/group-identifier pairs for the area within which that piece of the line lies. In writing ULPR, the user may rely upon a SET call’s having been done which makes it possible to use normalized device coordinates in calls to routines like CURVE, CURVED, GPL, etc. For more details, see the reference document for the package named AREAS and, in particular, the description of the subroutine ARDRLN.

3.4 MAPDRW

Draws a complete map, as described by the current values of the internal parameters of EZMAP.

3.4.1 Usage

Just call it. MAPDRW calls MAPINT (if the value of the internal parameter 'IN' indicates that initialization is required), MAPGRD, MAPLBL, and MAPLOT, in that order.

Users may sometimes wish to call directly the routines that MAPDRW calls. For example, they sometimes want to change the aspect ratio of a map drawn by EZMAP to something other than that implied by the projection selected; in order to do this, it is necessary to put calls to GETSET and SET in between the call to MAPINT and the calls to MAPGRD, MAPLBL, and MAPLOT. This is possible only if the user is calling those routines directly (rather than indirectly, by calling MAPDRW).

See the examples named “mpex01”, “mpex02”, “mpex04”, “mpex05”, “mpex06”, “mpex07”, and “mpex10”.

3.4.2 Arguments

None.

3.5 MAPEOD (NOUT,NSEG,IDLS,IDRS,NPTS,PNTS)

To examine each outline-dataset segment.

3.5.1 Usage

MAPEOD is called by MAPLOT to examine each segment in an outline dataset just before it is plotted and by MAPBLA to examine each segment in an outline dataset just before it is added to an area map. The default version of MAPEOD does nothing. A user-supplied version may cause selected segments to be deleted (to reduce the clutter in northern Canada, for example).

See the examples named “mpex03”, “mpex05”, and “mpex09”.

3.5.2 Arguments

NOUT (an input expression of type INTEGER) is the number of the outline dataset from which the segment comes, as follows:

NOUT	Dataset from which the segment came
1	'CO' - Continental outlines only.
2	'US' - U.S. state outlines only.
3	'PS' - Continental, U.S. state, and international outlines.
4	'PO' - Continental and international outlines.

NSEG (an input expression of type INTEGER) is the number of the segment within the outline dataset. The maps in the example named “mpex09” show segment numbers for the outline dataset 'CO'; the program may be modified to produce maps showing segment numbers for any outline dataset.

IDLS and **IDRS** (input expressions of type INTEGER) are identifiers for the areas to the left and right, respectively, of the segment. (Left and right are defined from the standpoint of a viewer standing at point 1 of the segment and looking toward point 2.) For a complete list of area identifiers, see the section named “AREA IDENTIFIERS”.

NPTS (an input/output variable of type INTEGER), on entry, is the number of points defining the outline segment. NPTS may be zeroed by MAPEOD to suppress any use of the segment on the map.

PNTS (an input/output array, dimensioned 2*NPTS, of type REAL) is an array of coordinates. PNTS(1) and PNTS(2) are the latitude and longitude of the first point, PNTS(3) and PNTS(4) the latitude and longitude of the second point, ... PNTS(2*NPTS-1) and PNTS(2*NPTS) the latitude and longitude of the last point. All values are in degrees. Longitudes are all between -180 and +180; no segment crosses the meridian at -180 (+180) degrees.

3.6 MAPFST (RLAT,RLON)

Draws lines on a map - used in conjunction with MAPVEC.

3.6.1 Usage

The statement

```
CALL MAPFST (RLAT,RLON)
```

is equivalent to the statement

```
CALL MAPIT (RLAT,RLON,0)
```

See the description of MAPIT.

3.6.2 Arguments

RLAT and **RLON** (input expressions of type REAL) are defined in the same way as for MAPIT (which see).

3.7 MAPGCI (ALAT,ALON,BLAT,BLON,NOPI,RLTI,RLNI)

Calculates the latitudes and longitudes of points on the shortest great circle route between two given points on the surface of the globe.

3.7.1 Usage

The statement

```
CALL MAPGCI (ALAT,ALON,BLAT,BLON,NOPI,RLTI,RLNI)
```

defines the positions of two points, A and B, on the globe and the number of equally-spaced points, NOPI, to be interpolated along the great circle route from A to B. The latitudes and longitudes of the interpolated points are returned to the caller in the arrays RLTI and RLNI. If the points A and B are exactly opposite one another on the globe, the code does not fail, but the direction of the great circle route will be somewhat unpredictable (since, in that case, there is more than one great circle route joining the two points).

3.7.2 Arguments

ALAT and **ALON** (input expressions of type REAL) specify the latitude and longitude of the point A.

BLAT and **BLON** (input expressions of type REAL) specify the latitude and longitude of the point B.

NOPI (an input expression of type INTEGER) specifies the number of equally-spaced points to be interpolated along the great circle from A to B.

RLTI and **RLNI** (output arrays of type REAL, each dimensioned at least NOPI) are the latitudes and longitudes of the interpolated points. Each lat/lon pair defines one of the interpolated points; the points are in order of increasing distance from A. The positions of the points A and B themselves are not returned in these arrays; only the positions of the interpolated points are.

3.8 MAPGRD

Draws a lat/lon grid.

3.8.1 Usage

The statement

```
CALL MAPGRD
```

draws a grid consisting of lines of latitude (parallels) and lines of longitude (meridians). If EZMAP needs initialization or if the error flag 'ER' is non-zero or if the parameter 'GR' is less than or equal to zero, MAPGRD does nothing.

The grid is drawn using calls to MAPIT and MAPIQ. By default, MAPGRD forces the value of the internal parameter 'DL' to zero, so that the grid will be drawn using calls to the routines FRSTD and VECTD, in the dash package. MAPGRD also calls the dash-package routine DASHDB to force the current dash pattern to the value specified by the internal parameter 'DA'. Before returning control to the user, MAPGRD restores the original value of 'DL' and the original dash pattern. A user version of the routine MAPUSR may be supplied to change the way in which the grid lines are drawn (as shown in the example "mpex09").

3.8.2 Arguments

None.

3.9 MAPGRM (IAMA,XCRA,YCRA,MCRA, . . .)

(The remaining arguments are IAAI, IAGI, NOGI, and ULPR.)

Draws lines of latitude and longitude masked against an existing area map. The example "eezmpa" described in the section "EXAMPLES" shows how MAPGRM may be used to limit the drawing of these lines to areas over ocean, but there are other possible uses.

3.9.1 Usage

The FORTRAN statement

```
CALL MAPGRM (IAMA,XCRA,YCRA,MCRA,IAAI,IAGI,MNOG,ULPR)
```

is used to draw lines of latitude and longitude masked against the existing area map in the array IAMA.

MAPGRM is just like MAPGRD, except that the grid lines are drawn using calls to MAPITM and MAPIQM, which does the masking of the lines against an area map and passes the pieces resulting from the masking process along to a user-provided line-drawing routine.

See the example named "eezmpa".

3.9.2 Arguments

IAMA (an input/output array, dimensioned as specified in a call to the AREAS routine ARINAM, of type INTEGER) is the array containing the area map against which lines of latitude and longitude are to be masked. The area map must have been initialized by a call to ARINAM; it may contain edges added to it by calling MAPBLA, as is the case in the example named "eezmpa", or it may contain a different set of edges to create some other desired effect. For example, an area map might be created that defines a region of interest, within which user data is available and within which lines of latitude and longitude are to be drawn. For more details, see the reference document for the package named AREAS.

XCRA and **YCRA** (scratch arrays, dimensioned at least MCRA, of type REAL) are to be used by MAPGRM in calls to the AREAS routine ARDRLN; they will eventually be used in calls to the user-provided line-processing routine ULPR.

MCRA (an input expression of type INTEGER) is the dimension of the arrays XCRA and YCRA.

IAAI and **IAGI** (scratch arrays, dimensioned at least NOGI, of type INTEGER) are to be used by MAPGRM in calls to the AREAS routine ARDRLN; they will eventually be used in calls to the user-provided line-processing routine ULPR. The mnemonics stand for "Integer Array of Area Identifiers" and "Integer Array of Group Identifiers", respectively.

NOGI (an input expression of type INTEGER) is the dimension of the arrays IAAI and IAGI. The mnemonic stands for "Number Of Group Identifiers (of edges in the area map)", which determines the required dimension of IAAI and IAGI.

ULPR is the name of the user-supplied line-processing routine. It must be declared EXTERNAL in the routine that calls MAPGRM, so that the compiler and loader will know that it is the name of a routine to be called instead of a variable. The user routine ULPR will be called once for each piece of a latitude/longitude line resulting from the masking process; it may decide to draw (or to not draw) each such piece. ULPR will be called using a FORTRAN statement like

```
CALL ULPR (XCRA,YCRA,NCRA,IAAI,IAGI,NGPS)
```

where XCRA and YCRA are real arrays holding the normalized device coordinates of NCRA points defining a polyline which is part of some latitude/longitude line and IAAI and IAGI are integer arrays holding NGPS area-identifier/group-identifier pairs for the area within which that piece of the line lies. In writing ULPR, the user may rely upon a SET call's having been done which makes it possible to use normalized device coordinates in calls to routines like CURVE, CURVED, GPL, etc. For more details, see the reference document for the package named AREAS and, in particular, the description of the subroutine ARDRLN.

3.10 MAPGTx (PNAM,xVAL) or MPGETx (PNAM,xVAL)

Gets the current value of a specified EZMAP parameter. An alternate name, more nearly consistent with those used in other NCAR Graphics packages, is provided for this routine.

3.10.1 Usage

Use one of the four statements

```
CALL MAPGTC (PNAM,CVAL) or CALL MPGETC (PNAM,CVAL)
CALL MAPGTI (PNAM,IVAL) or CALL MPGETI (PNAM,IVAL)
CALL MAPGTL (PNAM,LVAL) or CALL MPGETL (PNAM,LVAL)
CALL MAPGTR (PNAM,RVAL) or CALL MPGETR (PNAM,RVAL)
```

depending on whether the value to be retrieved is of type CHARACTER, INTEGER, LOGICAL, or REAL. The values of some parameters may be retrieved in more than one way. For example, the value of the initialization flag may be retrieved as a logical quantity (.TRUE. or .FALSE.) or as an integer (non-zero or zero).

See the examples named “mpex07” and “mpex08”.

3.10.2 Arguments

PNAM (an input expression of type CHARACTER) specifies the name of the parameter to get. Only the first two characters of the string are examined. See the section “INTERNAL PARAMETERS” for a list of all the internal parameters whose values may be retrieved.

xVAL (an output variable of type CHARACTER, INTEGER, LOGICAL, or REAL, depending on which routine is called) receives the value of the parameter specified by PNAM.

3.11 MAPINT

Initialization.

3.11.1 Usage

The statement

```
CALL MAPINT
```

initializes the EZMAP package. This is required initially and again after a call to any of the routines MAPPOS, MAPROJ, or MAPSET. The flag 'IN', which may be retrieved by a call to MAPGTI or MAPGTL, indicates whether or not initialization is required at a given time. As of now (April, 1987), no parameter change by means of a call to MAPSTx forces re-initialization; only calls to MAPPOS, MAPROJ, and MAPSET do.

To clarify the use of MAPINT: Once the internal parameters describing a map transformation are completely defined, MAPINT must be called; it computes as many of the transformation parameters as possible, so that, when MAPTRN is called, it can return the u/v coordinates corresponding to a particular lat/lon position as quickly as possible. If the transformation is changed, MAPINT must be re-called before the next call to MAPTRN. Routines which call MAPTRN and therefore depend on MAPINT's having been called are as follows: MAPBLA, MAPFST, MAPGRD, MAPGRM, MAPIQ, MAPIQA, MAPIQM, MAPIT, MAPITA, MAPITM, MAPLBL, MAPLOT, MAPTRA, MAPTRI, MAPVEC, and SUPCON. Routines which change the transformation in some way and therefore require MAPINT to be re-called are as follows: MAPPOS, MAPROJ, and MAPSET. If one is just doing a single plot, one does the parameter-setting calls that define the transformation, calls MAPINT to initialize, and then calls the various routines that draw the plot; in this case, a single call to MAPINT is all that is necessary.

One of the things that MAPINT does is call the SPPS routine SET to define the mapping from the u/v (projection) plane to the x/y (plotter) plane, so that u/v coordinates returned by the routines MAPTRA and MAPTRN can be used directly (as "user" coordinates) in calls to other NCAR Graphics routines. The user must be careful not to interfere with this process by doing an inappropriate call to SET in between calling MAPINT and calling other routines that depend on the proper SET call's having been done; if such a call must be done, the routine MAPRS (q.v.) may be used to restore the SET call done by MAPINT.

See the examples named "mpex09", "mpex10", and "eezmpa".

3.11.2 Arguments

None.

3.12 MAPIQ

Terminate a string of calls to MAPIT.

3.12.1 Usage

The statement

```
CALL MAPIQ
```

flushes MAPIT's buffers. It is particularly important that this be done before a STOP or a CALL FRAME and before changing the color index, dash pattern, etc.

See the description of MAPIT, below.

See the example named "mpexfi".

3.12.2 Arguments

None.

3.13 MAPIQA (IAMA,IGID,IAIL,IAIR)

Terminate a string of calls to MAPITA.

3.13.1 Usage

The statement

```
CALL MAPIQA (IAMA,IGID,IAIL,IAIR)
```

flushes MAPITA's buffers. It is particularly important that this be done before attempting to use the area map in the array IAMA for something.

See the description of MAPITA, below.

See the example named "cpex08", which is descibed in the reference document for the package named CONPACK.

3.13.2 Arguments

IAMA, **IGID**, **IAIL**, and **IAIR** are all as described for the routine MAPITA (described below).

3.14 MAPIQM (IAMA,XCRA,YCRA,MCRA, . . .)

(The remaining arguments are IAAI, IAGI, NOGI, and ULPR.)

Terminates a string of calls to the routine MAPITM.

3.14.1 Usage

The statement

```
CALL MAPIQM (IAMA,XCRA,YCRA,MCRA,IAAI,IAGI,NOGI,ULPR)
```

flushes MAPITM's buffers. It is particularly important that this be done before a STOP, a CALL FRAME, or a call to change line color, width, dash pattern, etc.

See the description of MAPITM, below.

There is currently no example available for the routine MAPIQM.

3.14.2 Arguments

IAMA, **XCRA**, **YCRA**, **MCRA**, **IAAI**, **IAGI**, **NOGI**, and **ULPR** are all as described for the routine MAPITM (described below).

3.15 MAPIT (RLAT,RLON,IFST)

Draws lines on a map.

3.15.1 Usage

MAPIT is used to draw lines on a map; it is called by various EZMAP routines (MAPLOT, MAPGRD, and MAPVEC) and may also be called directly by the user. MAPIT attempts to omit non-visible portions of lines and to handle "cross-over" - a jump from one end of the map to the other caused by the projection's having slit the globe along some half of a great circle and laid it open with the two sides of the slit at opposite ends of the map. Cross-over can occur on cylindrical and conical projections; MAPIT handles it gracefully on the former and not quite so well on the latter.

The EZMAP parameter 'DL' determines whether MAPIT draws solid lines or dotted lines. Dotted lines are drawn using calls to POINTS. Solid lines are drawn using calls to DASHD, FRSTD, and VECTD. The EZMAP parameters 'DD' and 'MV' also affect MAPIT's behavior. See the description of these parameters in the section "INTERNAL PARAMETERS".

A sequence of calls to MAPIT should be followed by a call to MAPIQ (which see, above) to flush its buffers before a STOP, a CALL FRAME, or a call to change the color index, line width, dash pattern, or the like.

Because the entire line segment joining the points in two contiguous pen-down calls to MAPIT is considered visible if both of the points are visible and invisible if both of the points are invisible, such points should not be too far apart on the globe.

See the example named "mpexfi".

3.15.2 Arguments

RLAT and **RLON** (input expressions, of type REAL) specify the latitude and longitude of a point to which the "pen" is to be moved. Both are given in degrees. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

IFST (an input expression, of type INTEGER) is 0 to do a "pen-up" move, 1 to do a "pen-down" move only if the distance from the last point to the new point is greater than 'MV' plotter units, and 2 or greater to do a "pen-down" move regardless of the distance from the last point to the new one.

3.16 MAPITA (RLAT,RLON,IFST,IAMA,IGID,IAIL,IAIR)

3.16.1 Usage

The routines MAPITA and MAPIQA may be used to add lines defined by a set of user-specified latitudes and longitudes to an area map; they attempt to omit non-visible portions of lines and to handle “cross-over”, in the same way that MAPIT does.

MAPITA is called like the EZMAP routine MAPIT (which see, above) but has some additional arguments:

```
CALL MAPITA (RLAT,RLON,IFST,IAMA,IGID,IAIL,IAIR)
```

Additional arguments are the area map array IAMA, a group identifier IGID, and left and right area identifiers IAIL and IAIR. MAPIQA is called like the EZMAP routine MAPIQ, to terminate a series of calls to MAPITA and to flush the buffers; it has the same additional arguments:

```
CALL MAPIQA (IAMA,IGID,IAIL,IAIR)
```

The additional arguments are passed by MAPITA and MAPIQA to the routine AREDAM, in the package named AREAS, for which a reference document is available.

See the example named “cpex08”, which is described in the reference document for the package named CONPACK.

3.16.2 Arguments

RLAT and **RLON** (input expressions, of type REAL) specify the latitude and longitude of a point to which the “pen” is to be moved. Both are given in degrees. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

IFST (an input expression, of type INTEGER) is 0 to do a “pen-up” move, 1 to do a “pen-down” move only if the distance from the last point to the new point is greater than 'MV' plotter units, and 2 or greater to do a “pen-down” move regardless of the distance from the last point to the new one.

IAMA (an input/output array of type INTEGER, dimensioned as specified in a call to the AREAS initialization routine ARINAM) is the area map to which lines in MAPITA’s buffer are to be added.

IGID (an input expression of type INTEGER) is the group identifier to be passed to the AREAS routine AREDAM when the lines in MAPITA’s buffer are added to the area map in IAMA.

IAIL (an input expression of type INTEGER) is the left area identifier to be passed to the AREAS routine AREDAM when the lines in MAPITA's buffer are added to the area map in IAMA.

IAIR (an input expression of type INTEGER) is the right area identifier to be passed to the AREAS routine AREDAM when the lines in MAPITA's buffer are added to the area map in IAMA.

3.17 MAPITM (RLAT,RLON,IFST,IAMA,XCRA,YCRA,MCRA, . . .)

(The remaining arguments are IAAI, IAGI, NOGI, and ULPR.)

3.17.1 Usage

The routines MAPITM and MAPIQM may be used to draw lines defined by a set of user-specified latitudes and longitudes on a map, masked by the areas defined by an area map (perhaps one created by a call to MAPBLA or perhaps one created by a set of calls to routines in the package AREAS); like MAPIT and MAPIQ, they attempt to omit non-visible portions of lines and to handle “cross-over”.

MAPITM is called like the EZMAP routine MAPIT (which see, above) but has some additional arguments:

```
CALL MAPITM (RLAT,RLON,IFST,IAMA,XCRA,YCRA,MCRA,IAAI,IAGI,NOGI,ULPR)
```

Additional arguments are the area-map array IAMA, coordinate arrays XCRA and YCRA, dimensioned MCRA, integer arrays IAAI and IAGI, dimensioned NOGI, and a user-provided line-processing routine named ULPR. MAPIQM is called like the EZMAP routine MAPIQ, to terminate a series of calls to MAPITM and to flush the buffers; it has the same additional arguments:

```
CALL MAPIQM (IAMA,XCRA,YCRA,MCRA,IAAI,IAGI,NOGI,ULPR)
```

The additional arguments are passed by MAPITM and MAPIQM to the routine ARDRLN, in the package named AREAS, for which a reference document is available.

No example is currently available for MAPITM.

3.17.2 Arguments

RLAT and **RLON** (input expressions, of type REAL) specify the latitude and longitude of a point to which the “pen” is to be moved. Both are given in degrees. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

IFST (an input expression, of type INTEGER) is 0 to do a “pen-up” move, 1 to do a “pen-down” move only if the distance from the last point to the new point is greater than 'MV' plotter units, and 2 or greater to do a “pen-down” move regardless of the distance from the last point to the new one.

IAMA (an input/output array of type INTEGER, dimensioned as specified in a call to the AREAS initialization routine ARINAM) is the array containing the area map against which lines drawn by MAPIQM will be masked.

XCRA and **YCRA** (scratch arrays of type REAL, each dimensioned MCRA) are to be passed by MAPIQM to the AREAS routine ARDRLN, which uses them in calls to the user line-processing routine ULPR. They will hold the X and Y coordinates of points in the fractional coordinate system defining some portion of the projection of a user-defined polyline on the globe.

MCRA (an input expression of type INTEGER) is the size of each of the arrays XCRA and YCRA. The value of MCRA must be at least two. For most applications, the value 100 works nicely.

IAAI and **IAGI** (scratch arrays of type INTEGER, each dimensioned NOGI) are to be passed by MAPIQM to the AREAS routine ARDRLN, which uses them in calls to the user line-processing routine ULPR. They will hold area identifier/group identifier pairs for the area containing the polyline fragment defined by XCRA and YCRA.

NOGI (an input expression of type INTEGER) is the size of each of the arrays IAAI and IAGI. The value of NOGI must be greater than or equal to the number of groups of edges placed in the area map in IAMA.

ULPR is the name of a user-provided line-processing routine. This name must appear in an EXTERNAL statement in the routine that calls MAPITM, so that the compiler and loader will know that it is the name of a routine to be called, rather than the name of a variable.

3.18 MAPLBL

Labels the map.

3.18.1 Usage

The statement

```
CALL MAPLBL
```

if the parameter 'LA' is set appropriately, labels the international date line (ID), the equator (EQ), the Greenwich Meridian (GM), and the poles (NP and SP), and, if the parameter 'PE' is set appropriately, draws the perimeter of the map. If EZMAP needs initialization or if the error flag 'ER' is set, MAPLBL does nothing.

See the example named “eezmpa”.

3.18.2 Arguments

None.

3.19 MAPLOT

Draws geographical outlines.

3.19.1 Usage

The statement

```
CALL MAPLOT
```

draws the continental and/or international and/or U.S. state outlines selected by the EZMAP parameter 'OU'; the parameter 'DO' determines whether solid lines or dotted lines are used. If EZMAP currently needs initialization or if the error flag 'ER' is set, MAPLOT does nothing.

The outlines are drawn using calls to MAPIT and MAPIQ. By default, MAPGRD forces the value of the internal parameter 'DL' equal to the value of the internal parameter 'DO'; it also supplies the dash package with a solid-line dash pattern. When 'DO' is zero, the outlines are drawn using calls to the routines FRSTD and VECTD, in the dash package, and this gives solid lines. When 'DO' is non-zero, the outlines are drawn using calls to the SPPS routine POINTS, which gives dotted lines. Before returning control to the user, MAPLOT restores the original value of 'DL' and the original dash pattern. A user version of the routine MAPUSR may be supplied to change the way in which the outlines are drawn (as shown in the example “mpex09”).

See the examples named “mpex09” and “eezmpa”.

3.19.2 Arguments

None.

3.20 MAPPOS (XLOW,XROW,YBOW,YTOW)

Positions the map on the plotter frame.

3.20.1 Usage

The statement

```
CALL MAPPOS (XLOW,XROW,YBOW,YTOW)
```

sets four EZMAP parameters specifying the position of a window in the plotter frame within which maps are to be drawn. Each of the arguments is between 0. and 1., inclusive, and specifies the position of one edge of the window, as a fraction of the distance from left to right, or from bottom to top, across the window. The map is centered within the window and made as large as possible, but maintains its intrinsic shape (aspect ratio).

If it is desired to give a map a different aspect ratio than that implied by the projection being used, it will be necessary to insert, immediately after the call to MAPINT, an appropriate call to SET to override the one done by MAPINT and therefore to specify precisely the mapping of the u/v plane into a desired viewport on the plotter frame. Typically, one might call GETSET to find out what arguments 5-8 should be in one's own call to SET. Note that, if MAPDRW is being called, that call will have to be replaced by calls directly to MAPINT, MAPGRD, MAPLBL, and MAPLOT, in order to allow one's own call to SET to be placed immediately after the call to MAPINT.

See the examples named "mpex04", "mpex05", "mpex06", and "mpex07".

3.20.2 Arguments

XLOW (an input expression, of type REAL) specifies the position of the left edge of the window. The default is .05.

XROW (an input expression, of type REAL) specifies the position of the right edge of the window. The default is .95.

YBOW (an input expression, of type REAL) specifies the position of the bottom edge of the window. The default is .05.

YTOW (an input expression, of type REAL) specifies the position of the top edge of the window. The default is .95.

3.21 MAPROJ (JPRJ,PLAT,PLON,ROTA)

Set the projection to be used.

3.21.1 Usage

The statement

```
CALL MAPROJ (JPRJ,PLAT,PLON,ROTA)
```

sets EZMAP parameters specifying the projection to be used for subsequent maps.

See the examples named “mpex01”, “mpex02”, “mpex04”, “mpex05”, “mpex06”, “mpex07”, “mpex09”, “mpex10”, and “eezmpa”.

3.21.2 Arguments

JPRJ (an input expression, of type CHARACTER) defines the desired projection type. All the possible values are two characters in length; these are the possibilities:

The conic projection:

- '**LC**' - Lambert conformal conic with two standard parallels.

The azimuthal projections:

- '**ST**' - Stereographic.
- '**OR**' - Orthographic. The EZMAP parameter 'SA' will be zeroed. See the note below.
- '**LE**' - Lambert equal area.
- '**GN**' - Gnomonic.
- '**AE**' - Azimuthal equidistant.
- '**SV**' - Satellite-view. If the EZMAP parameter 'SA' is less than or equal to 1., it will be reset to 6.631 (the value for a satellite in a geosynchronous orbit). See the note below.

The cylindrical projections:

- '**CE**' - Cylindrical equidistant.
- '**ME**' - Mercator.
- '**MO**' - Mollweide-type.

Note: The orthographic and satellite-view projections have the same internal identifier. The EZMAP parameter 'SA' determines which will be used. If a call to MAPROJ selecting one or the other is followed by a call to MAPSTR resetting 'SA', it may have the effect of causing the other to be used. See the description of 'SA', below, in the description of the routine MAPSTx.

PLAT, PLON, and **ROTA** (input expressions, of type REAL) are angular quantities, in degrees. How they are used depends on the value of JPRJ, as follows:

- If JPRJ is not equal to 'LC': PLAT and PLON define the latitude and longitude of the pole of the projection - the point on the globe which is to project to the origin of the u/v plane. PLAT must be between -90. and +90., inclusive, positive in the northern hemisphere, negative in the southern. PLON must be between -180. and +180., inclusive, positive to the east, and negative to the west, of Greenwich. ROTA is the angle between the v axis and north at the origin. It is taken to be positive if the angular movement from north to the v axis is counter-clockwise, negative otherwise. If the origin is at the north pole, “north” is considered to be in the direction of PLON+180. If the origin is at the south pole, “north” is considered to be in the direction of PLON. For the cylindrical projections, the axis of the projection is parallel to the v axis.

- If JPRJ is equal to 'LC': PLON defines the central meridian of the projection, and PLAT and ROTA define the two standard parallels. If PLAT and ROTA are equal, a simpler conic projection, with one standard parallel, is used.

For more detailed descriptions of the projections, see the section named “PROJECTIONS”.

3.22 MAPRS

Re-calls SET.

3.22.1 Usage

The statement

```
CALL MAPRS
```

repeats the SET call last done by the routine MAPINT. This might be used when user lines are to be plotted over a map generated in a different overlay (e.g., using a flash buffer), and when the system plot package does not reside in an outer overlay.

No example is available for MAPRS.

3.22.2 Arguments

None.

3.23 MAPRST (IFNO)

Restores the state of EZMAP saved by an earlier call to MAPSAV.

3.23.1 Usage

The statement

```
CALL MAPRST (IFNO)
```

restores EZMAP to a previously saved state (frequently the default state) by reading, from the unit specified by IFNO, values of all user-settable parameters, and then executing MAPINT.

No example is available for MAPRST.

3.23.2 Arguments

IFNO (an input expression, of type INTEGER) is the number of a unit from which a single unformatted record is to be read. It is the user's responsibility to position this unit. MAPRST does not rewind it, either before or after reading the record.

3.24 MAPSAV (IFNO)

Saves the current state of EZMAP for later restoration by MAPRST.

3.24.1 Usage

The statement

```
CALL MAPSAV (IFNO)
```

saves the current state of EZMAP (frequently the default state) by writing, on the unit specified by IFNO, the current values of all the user-settable parameters.

No example is available for MAPSAV.

3.24.2 Arguments

IFNO (an input expression, of type INTEGER) is the number of a unit to which a single unformatted record is to be written. It is the user's responsibility to position this unit. MAPSAV does not rewind it, either before or after writing the record.

3.25 MAPSET (JLTS,PLM1,PLM2,PLM3,PLM4)

To specify the rectangular portion of the u/v plane to be drawn.

3.25.1 Usage

The statement

```
CALL MAPSET (JLTS,PLM1,PLM2,PLM3,PLM4)
```

specifies what portion of the u/v plane is to be plotted.

See the examples named “mpex01”, “mpex02”, “mpex04”, “mpex07”, “mpex09”, “mpex10”, and “eezmpa”.

3.25.2 Arguments

JLTS (an input expression, of type CHARACTER) is a character string specifying how the limits of the map are to be chosen. There are five possibilites, as follows:

- JLTS='MA' (MAXIMUM). The maximum useful area produced by the projection is plotted. PLM1, PLM2, PLM3, and PLM4 are not used.
- JLTS='CO' (CORNERS). The points (PLM1,PLM2) and (PLM3,PLM4) are to be at opposite corners of the map. PLM1 and PLM3 are latitudes, in degrees. PLM2 and PLM4 are longitudes, in degrees. If a cylindrical projection is being used, the first point should be on the left edge of the map and the second point on the right edge; otherwise, the order makes no difference.
- JLTS='PO' (POINTS). PLM1, PLM2, PLM3, and PLM4 are two-element arrays giving the latitudes and longitudes, in degrees, of four points which are to be on the edges of the rectangular map. If a cylindrical projection is being used, the first point should be on the left edge and the second point on the right edge; otherwise, the order makes no difference.
- JLTS='AN' (ANGLES). PLM1, PLM2, PLM3, and PLM4 are positive angles, in degrees, representing angular distances from a point on the map to the left, right, bottom, and top edges of the map, respectively. For most projections, these angles are measured with the center of the earth at the vertex and represent angular distances from the point which projects to the origin of the u/v plane; on a satellite-view projection, they are measured with the satellite at the vertex and represent angular deviations from the line of sight. Angular limits are particularly useful for polar projections and for the satellite-view projection; they are not appropriate for the Lambert conformal conic and an error will result if one attempts to use JLTS='AN' with JPRJ='LC'.
- JLTS='LI' (LIMITS). PLM1, PLM2, PLM3, and PLM4 specify the minimum value of u, the maximum value of u, the minimum value of v, and the maximum value of v, respectively. Knowledge of the projection equations is necessary in order to use this option correctly.

PLM1, PLM2, PLM3, and **PLM4** (input arrays, dimensioned 2, of type REAL) are as described above, depending on the value of JLTS. Note that each is a two-element array. Strictly speaking, the FORTRAN standard requires that they be declared as such, even when only the first element of each array is used.

3.26 MAPSTx (PNAM,xVAL) or MPSETx (PNAM,xVAL)

To set the values of certain EZMAP parameters. An alternate name, more nearly consistent with those used in other NCAR Graphics packages, is provided for this routine

3.26.1 Usage

Use one of the four statements

```
CALL MAPSTC (PNAM,CVAL) or CALL MPSETC (PNAM,CVAL)
CALL MAPSTI (PNAM,IVAL) or CALL MPSETI (PNAM,IVAL)
CALL MAPSTL (PNAM,LVAL) or CALL MPSETL (PNAM,LVAL)
CALL MAPSTR (PNAM,RVAL) or CALL MPSETR (PNAM,RVAL)
```

depending on whether the value to be set is of type CHARACTER, INTEGER, LOGICAL, or REAL. Some parameters may be set in more than one way. For example, the parameter 'GR', which specifies the grid spacing, may be given the value 10.0 in either of two ways:

```
CALL MAPSTI ('GR',10)
CALL MAPSTR ('GR',10.)
```

The flag which controls dotting of outlines may be turned on using either of these calls:

```
CALL MAPSTI ('DO',1)
CALL MAPSTL ('DO',.TRUE.)
```

The important point to remember is that the last character of the routine name implies the type of its second argument.

See the examples named “mpex01”, “mpex02”, “mpex04”, “mpex05”, “mpex06”, “mpex07”, “mpex08”, “mpex09”, “mpex10”, “mpexfi”, and “eezmpa”.

3.26.2 Arguments

PNAM (an input expression, of type CHARACTER*2) specifies the name of an internal parameter whose value is to be set. Only the first two characters of this string are examined.

xVAL (an input expression of type CHARACTER, INTEGER, LOGICAL, or REAL, depending on which routine is called) contains the value to be given to the parameter specified by PNAM.

See the section “INTERNAL PARAMETERS” for descriptions of all the internal parameters.

3.27 MAPTRA (RLAT,RLON,UVAL,VVAL)

To project points.

MAPTRA is just like MAPTRN except that, if a point being projected is outside the boundary of the map (as defined by the last call to MAPSET and by the value of the internal parameter 'EL'), a special value is returned to signal this condition; see the sub-section named “Arguments”, below. (The default version of the CONPACK routine named

CPMPXY now calls MAPTRA, rather than MAPTRN; it was for this purpose that MAPTRA was created, but users may wish to call it, as well.)

3.27.1 Usage

The statement

```
CALL MAPTRA (RLAT,RLON,UVAL,VVAL)
```

is used to find the projection in the u/v plane of a point whose latitude and longitude are known. MAPTRA may be called at any time after EZMAP has been initialized (by calling MAPINT or otherwise).

Currently, no example is available for MAPTRA.

3.27.2 Arguments

RLAT and **RLON** (input expressions, of type REAL) are the latitude and longitude, respectively, of a point on the globe. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

UVAL and **VVAL** (output variables, of type REAL) define the point (UVAL,VVAL) that is the projection in the u/v plane of (RLAT,RLON). The units of UVAL and VVAL depend on the projection.

If the point is not projectable or if it is outside the boundary of the map, as defined by the last call to MAPSET and by the value of the parameter 'EL', UVAL is returned equal to 1.E12.

3.28 MAPTRI (UVAL,VVAL,RLAT,RLON)

To perform inverse transformations.

3.28.1 Usage

The statement

```
CALL MAPTRI (UVAL,VVAL,RLAT,RLON)
```

is used to find the latitude and longitude of a point whose projection in the u/v plane is known. MAPTRI may be called at any time after EZMAP has been initialized (by calling MAPINT or otherwise).

See the example named “mpex10”.

3.28.2 Arguments

UVAL and **VVAL** (input expressions, of type REAL) define the point (UVAL,VVAL) that is the projection in the u/v plane of the point whose latitude and longitude are desired. The units of UVAL and VVAL depend on the projection.

RLAT and **RLON** (output variables, of type REAL) are the latitude and longitude, respectively, of the point, in degrees. RLAT will be between -90. and +90., inclusive; RLON will be between -180. and +180., inclusive.

If the point (UVAL,VVAL) is not the projection of some point of the globe or is outside the boundary of the map, RLAT and RLON are returned equal to 1.E12.

3.29 MAPTRN (RLAT,RLON,UVAL,VVAL)

To project points.

3.29.1 Usage

The statement

```
CALL MAPTRN (RLAT,RLON,UVAL,VVAL)
```

is used to find the projection in the u/v plane of a point whose latitude and longitude are known. MAPTRN may be called at any time after EZMAP has been initialized (by calling MAPINT or otherwise).

See the example named “mpex09”.

3.29.2 Arguments

RLAT and **RLON** (input expressions, of type REAL)are the latitude and longitude, respectively, of a point on the globe. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

UVAL and **VVAL** (output variables, of type REAL) define the point (UVAL,VVAL) that is the projection in the u/v plane of (RLAT,RLON). The units of UVAL and VVAL depend on the projection.

If the point is not projectable, UVAL is returned equal to 1.E12. Note that, if the point is projectable, but outside the boundary of the map, as defined by the last call to MAPSET, its u and v coordinates are still returned by MAPTRN. The user must do the test required to determine if the point is within limits, if that is necessary.

3.30 MAPUSR (IPRT)

To change the appearance of the various parts of a map.

3.30.1 Usage

EZMAP executes the statement

```
CALL MAPUSR (IPRT)
```

just before and just after each portion of a map is drawn. The default version of MAPUSR does nothing.

A user-supplied version of MAPUSR may set/reset the dotting parameter 'DL', the DASHCHAR dash pattern, the color index, etc., so as to achieve a desired effect.

See the example named “mpex08”.

3.30.2 Arguments

IPRT (input expression, of type INTEGER), if positive, says that a particular part of the map is about to be drawn, as follows:

IPRT	Part of map about to be drawn
1	Perimeter.
2	Grid.
3	Labels.
4	Limb lines.
5	Continental outlines.
6	U.S. state outlines.
7	International outlines.

If IPRT is negative, it says that drawing of the last part is complete. The absolute value of IPRT will be one of the above values. Changed quantities should be restored.

3.31 MAPVEC (RLAT,RLON)

Draws lines on a map - used in conjunction with MAPFST.

3.31.1 Usage

The statement

```
CALL MAPVEC (RLAT,RLON)
```

is equivalent to the statement

```
CALL MAPIT (RLAT,RLON,1)
```

See the description of MAPIT.

3.31.2 Arguments

RLAT and **RLON** (input expressions, of type REAL) are defined as for MAPIT (which see).

3.32 MPGETx (PNAM,xVAL) and MPSETx (PNAM,xVAL)

These routines are alternate entry points for MAPGTx and MAPSTx, respectively. See the descriptions of those routines.

3.33 SUPCON (RLAT,RLON,UVAL,VVAL)

An old equivalent of the new routine MAPTRN. Provided for compatibility with earlier versions of EZMAP/SUPMAP. If efficiency is a consideration, bypass this routine and call MAPTRN directly.

3.33.1 Usage

The statement

```
CALL SUPCON (RLAT,RLON,UVAL,VVAL)
```

is exactly equivalent to the statement

```
CALL MAPTRN (RLAT,RLON,UVAL,VVAL)
```

3.33.2 Arguments

RLAT, RLON, UVAL, and **VVAL** are defined as for MAPTRN (which is described above).

3.34 SUPMAP (JPRJ,PLAT,PLON,ROTA, . . .)

(The remaining arguments are PLM1, PLM2, PLM3, PLM4, JLTS, JGRD, IOUT, IDOT, and IERR.)

An implementation of the routine from which EZMAP grew.

3.34.1 Usage

The statement

```
      CALL SUPMAP (JPRJ,PLAT,PLON,ROTA,PLM1,PLM2,PLM3,PLM4,JLTS,
     +             JGRD,IOUT,IDOT,IERR)
```

creates a map of a desired portion of the globe, according to a desired projection, with desired outlines drawn in, and with lines of latitude and longitude at desired intervals. An appropriate call to the routine SET is performed, and the routine SUPCON (which see) is initialized so that the user may map points of known latitude and longitude to points in the u/v plane and use the u/v coordinates to draw objects on the map produced by SUPMAP.

See the examples named “mpex03”, “mpex08”, and “mpexfi”.

3.34.2 Arguments

JPRJ (input expression, of type INTEGER) defines the projection type and indicates whether or not continental outlines are to be plotted, as follows:

- IABS(JPRJ) defines the projection type, as follows (values less than 1 or greater than 10 are treated as 1 or 10, respectively):

IABS(JPRJ)	Projection Type
1	Stereographic.
2	Orthographic.
3	Lambert conformal conic.
4	Lambert equal area.
5	Gnomonic.
6	Azimuthal equidistant.
7	Satellite view.
8	Cylindrical equidistant.
9	Mercator.
10	Mollweide-type.

  Using the value 2 causes the EZMAP parameter 'SA' to be zeroed. ('SA', if greater than 1., says that a satellite-view projection, rather than an orthographic projection, is to be used, and specifies the distance of the satellite from the center of the earth, in units of earth radii.)

  Using the value 7 causes 'SA' to be examined. If it has a non-zero value, the value is left alone. If it has a zero value, its value is reset to 6.631, which is about right for a satellite in a geosynchronous equatorial orbit.

- The sign of JPRJ, when IOUT is -1, 0, or +1, indicates whether the continental outlines are to be plotted or not. See IOUT, below.

PLAT, PLON, and **ROTA** (input expressions, of type REAL) define the origin of the projection and its rotation angle and are used in the same way as they would be in a call to the routine MAPROJ (which see).

JLTS (input expression, of type INTEGER)**,** and **PLM1, PLM2, PLM3,** and **PLM4** (input arrays, dimensioned 2, of type REAL) specify the rectangular limits of the map. These arguments are used in the same way as they would be in a call to MAPSET (which see), except that JLTS is an integer instead of a character string. IABS(JLTS) may take on the values 1 through 5, as follows:

IABS(JLTS)	Equivalent character string in a call to MAPSET
1	'MA' - maximal area desired.
2	'CO' - corner points given in PLM1, PLM2, PLM3, and PLM4.
3	'LI' - U/V limits given in PLM1, PLM2, PLM3, and PLM4.
4	'AN' - angles given in PLM1, PLM2, PLM3, and PLM4.
5	'PO' - edge points given in PLM1, PLM2, PLM3, and PLM4.

At one time, the sign of JLTS specified whether or not a line of text was to be written at the bottom of the plot produced. This line may no longer be written and the sign of JLTS is therefore ignored.

JGRD (input expression, of type INTEGER) is used in the following way: The value of "MOD(IABS(JGRD),1000)" is the value, in degrees, of the interval at which lines of latitude and longitude are to be plotted. If the given interval is zero, grid lines and labels are not plotted. If JGRD is less than zero, the perimeter is not plotted. Set JGRD to -1000 to suppress both grid lines and perimeter and to +1000 to suppress the grid lines, but leave the perimeter. The value -0 may have a meaning on ones' complement machines, but should be avoided; use -1000 instead.

IOUT (input expression, of type INTEGER) has the value 0 to suppress U.S. state outlines, and the value -1 or +1 to plot U.S. state outlines. In both of these cases, the sign of JPRJ indicates whether continental outlines are to be plotted (JPRJ positive) or not (JPRJ negative). Originally, SUPMAP recognized only these values of IOUT; now, if IOUT is less than 0 or greater than 1, the sign of JPRJ is ignored, and IOUT selects an outline group, as follows:

IOUT	Outline group selected
-2 or less	'NO' (no outlines)
2	'CO' (continental outlines)
3	'US' (U.S. state outlines)
4	'PS' (continental outlines plus international outlines plus U.S. state outlines)
5 or greater	'PO' (continental outlines plus international outlines)

At one time, the sign of IOUT specified whether or not a line of text was to be written on the print output unit. This may no longer be done.

IDOT (input expression, of type INTEGER) is a 0 to get continuous outlines, a 1 to get dotted outlines.

IERR (output variable, of type INTEGER) is the only output parameter. A non-zero value indicates that an error has occurred. The section "ERROR CONDITIONS" lists the possible values of IERR.

4.0 INTERNAL PARAMETERS

EZMAP has a large group of internal parameters, each of which affects the behavior of one or more of the EZMAP routines. The current values of the internal parameters may be retrieved using the routines MAPGTC, MAPGTI, MAPGTL, and MAPGTR. The values of most of the internal parameters may be reset using the routines MAPSTC, MAPSTI, MAPSTL, and MAPSTR; some of the parameters are intended for retrieval only and may not be given new values in this way. In the following table are listed, in alphabetical order, all of the internal parameters of EZMAP:

Name	Type	Description
'AR'	C	For retrieval only. The value of the map limits specifier JLTS from the last call to MAPSET. The default value is 'MA'.
'Cn'	I	"n" is an integer between 1 and 7. Each 'Cn', if zero or greater, specifies the color index of some part of the map. The user must do the calls to GKS routines to define the color indices. The default value of each 'Cn' is -1, indicating no change in color index from one part of the map to another.
'C1'	I	Color index for perimeter. See 'Cn', above.
'C2'	I	Color index for grid. See 'Cn', above.
'C3'	I	Color index for labels. See 'Cn', above.
'C4'	I	Color index for limb lines. See 'Cn', above.
'C5'	I	Color index for continent outlines. See 'Cn', above.
'C6'	I	Color index for U.S. state outlines. See 'Cn', above.
'C7'	I	Color index for outlines of countries of the world. See 'Cn', above.
'DA'	I	Dashed-line pattern for the grids. A 16-bit quantity. The default is 21845 (octal 52525 or binary 0101010101010101).
'DD'	I,R	Distance between dots along a dotted line drawn by MAPIT. The default value is 12 (out of 4096; see 'RE', below).
'DL'	I,L	If true (non-zero), user calls to MAPIT draw dotted lines. The default is false (zero); lines drawn by MAPIT are solid or dashed, depending on the current state of the DASHCHAR package. 'DL' may be reset by a user version of MAPUSR to change the way in which the perimeter, the grid, the limb lines, and the outlines are drawn.
'DO'	I,L	If true (non-zero), outlines are dotted. The default is false (zero); outlines are solid.

Name	Type	Description
'EL'	I,L	If true (non-zero), requests an elliptical perimeter: only that part of the map which falls inside an ellipse inscribed within the normal rectangular perimeter is drawn. This is particularly appropriate for use with azimuthal projections and angular limits specifying a square, in which case the ellipse becomes a circle, but it will work for any map. The default value is false (zero).
'ER'	I	For retrieval only. The current value of the error flag. Initially, the value is zero (no error).
'GD'	R	The distance between points used to draw the grid, in degrees. The default value is 1.; user values must fall between .001 and 10.
'GR'	I,R	The desired grid spacing. A zero suppresses the grid. The default is 10 degrees.
'G1'	I	The group identifier to be used by MAPBLA when putting into the area map the group of edges that define the division of the plotter frame into the projected images of geographic entities.
'G2'	I	The group identifier to be used by MAPBLA when putting into the area map the group of edges that define the division of the plotter frame into vertical strips.
'IN'	I,L	For retrieval only. Initialization flag. If true (non-zero), it says that EZMAP is in need of initialization (by a CALL MAPINT). The default value is true (non-zero).
'LA'	I,L	If true (non-zero), label the meridians and the poles. The default is true (non-zero).
'LS'	I	Controls label size. A character width, to be used in a call to PWRIT. The default value is 1, which gives a character width of 12 plotter units.
'MV'	I,R	Minimum vector length for MAPIT. A point closer to the previous point than this is omitted. The default value is 4 (out of 4096; see 'RE', below).
'OU'	C	Says which set of outline data to use. The possible values are 'NO', for no outlines, 'CO', for continental outlines, 'US', for U.S. state outlines, 'PS', for continental outlines plus international outlines plus U.S. outlines, and 'PO', for continental outlines plus international outlines. Default is 'CO'.
'PE'	I,L	If true (non-zero), draw the perimeter (a rectangle or an ellipse, depending on the value of 'EL'). The default is true (non-zero).
'PN'	I,R	For retrieval only. The value of PLON from the last call to MAPROJ. The default value is zero.
'PR'	C	For retrieval only. The value of the projection specifier JPRJ from the last call to MAPROJ. The default value is 'CE'.
'PT'	I,R	For retrieval only. The value of PLAT from the last call to MAPROJ. The default value is zero.
'Pn'	I,R	For retrieval only. “n” is an integer from 1 to 8, inclusive. Retrieves values from the call to MAPSET. 'P1' through 'P4' specify PLM1(1), PLM2(1), PLM3(1), and PLM4(1), while 'P5' through 'P8' specify PLM1(2), PLM2(2), PLM3(2), and PLM4(2). Default values are all zero.
'RE'	I,R	The width of the target plotter, in plotter units. The default value is 4096.
'RO'	I,R	For retrieval only. The value of ROTA from the last call to MAPROJ. The default value is zero.

Name	Type	Description
'SA'	I,R	If 'SA' is greater than 1., a satellite-view projection replaces the orthographic. The value is the distance of the satellite from the center of the earth, in multiples of the earth's radius. The default value is zero. See also 'S1' and 'S2', below.
'S1' and 'S2'	I,R	Used only when 'SA' is greater than 1. Both are angles, in degrees. 'S1' measures the angle between the line to the center of the earth and the line of sight (to which the projection plane is perpendicular). If 'S1' is zero, the projection shows the earth as seen by a satellite looking straight down; call this the "basic view". If 'S1' is non-zero, 'S2' measures the angle from the positive u axis of the basic view to the line OP, where O is the origin of the basic view and P is the projection of the desired line of sight on the basic view. 'S2' is positive if measured counter-clockwise.
'SR'	R	A search radius, in degrees, used by MAPINT in finding the latitude/longitude range of a map. The default value is 1.; user values must lie between .001 and 10. Should not be changed except by advice of a consultant.
'VS'	I	The vertical-stripping parameter, which determines whether MAPBLA puts into the area map edge group 'G2', defining a set of vertical strips. A negative or zero value of 'VS' prevents MAPBLA from doing this. A value greater than zero requests that it be done and specifies the number of vertical strips to be created. The default value of 'VS' is 1.
'XL', 'XR', 'YB', and 'YT'	R	For retrieval only. The values of the arguments XLOW, XROW, YBOW, and YTOW from the last call to MAPPOS. Defaults are .05, .95, .05, and .95, respectively.

5.0 ERROR HANDLING

When an EZMAP routine detects an error condition, it calls the routine SETER, which is the principal routine in the error-handling package for NCAR Graphics. (There is a programmer document describing SETER and associated routines; see that document for complete information about error-handling in NCAR Graphics.)

By default, SETER prints a line and STOPs. The line printed will look something like this:

```
ERROR    3 IN MAPTRN - ATTEMPT TO USE NON-EXISTENT PROJECTION
```

The error number ("3", in the example) may be of use to a consultant (to determine exactly where the error occurred), but is not otherwise meaningful. The actual error message consists of the name of the routine in which the error occurred ("MAPTRN", in the example), a blank, a minus sign, another blank, and, lastly, a short description of the error.

All errors are "recoverable" in the sense that, if the user program puts SETER into "recovery mode", control will be returned to the caller of the EZMAP routine in which the error occurred. In some cases, it is then possible to take remedial action to get around whatever problem has occurred; in any case, the error flag can be cleared and execution of the user's program can continue.

When SETER is in recovery mode (and, occasionally, even when it is not), error messages may have a somewhat more complicated form, like this:

```
SUPCON/MAPTRN - ATTEMPT TO USE NON-EXISTENT PROJECTION
```

What this particular error message says is that SUPCON called MAPTRN, which detected an error condition and called SETER. Upon getting control back from MAPTRN, SUPCON detected the fact that MAPTRN had logged an error. It augmented the error message by prepending its own name, followed by a slash, and then passed control back to the user. Of course, there can be more than two such levels of routine calls indicated in the error message: in a few cases, seven or eight routine names may be listed, each separated from the next by a slash.

The various error conditions in EZMAP are described in the list below. Each bulleted item includes an error message and a thumb-nail description of the error. The items in the list are arranged in alphabetical order. If you get an error message with one or more prefixed subroutine names, as described above, omit them and look up the result in this list. Note that, since EZMAP routines sometimes call other routines, elsewhere in NCAR Graphics, that can detect error conditions and call SETER, the error message you get by calling an EZMAP routine may not be listed here, but in the programmer document for some other package.

- **MAPBLA - UNCLEARED PRIOR ERROR**

  When MAPBLA was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPBLM - UNCLEARED PRIOR ERROR**

  When MAPBLM was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPCHI - ERROR EXIT FROM GQPLCI**

  MAPCHI is an internal routine used to manage color, intensity, dash patterns, etc. The GKS routine GQPLCI, which is called to get the current value of the polyline color index, has returned a non-zero error code.

- **MAPCHI - ERROR EXIT FROM GQPMCI**

  MAPCHI is an internal routine used to manage color, intensity, dash patterns, etc. The GKS routine GQPMCI, which is called to get the current value of the polymarker color index, has returned a non-zero error code.

- **MAPCHI - ERROR EXIT FROM GQTXCI**

MAPCHI is an internal routine used to manage color, intensity, dash patterns, etc. The GKS routine GQTXCI, which is called to get the current value of the text color index, has returned a non-zero error code.

- **MAPDRW - UNCLEARED PRIOR ERROR**

  When MAPDRW was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPFST - UNCLEARED PRIOR ERROR**

  When MAPFST was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGCI - UNCLEARED PRIOR ERROR**

  When MAPGCI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGRD - UNCLEARED PRIOR ERROR**

  When MAPGRD was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGRM - UNCLEARED PRIOR ERROR**

  When MAPGRM was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGTC - UNCLEARED PRIOR ERROR**

  When MAPGTC was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGTC - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPGTC is not recognizable as the name of an internal parameter of EZMAP.

- **MAPGTI - UNCLEARED PRIOR ERROR**

  When MAPGTI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGTI - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPGTI is not recognizable as the name of an internal parameter of EZMAP.

- **MAPGTL - UNCLEARED PRIOR ERROR**

When MAPGTL was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGTL - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPGTL is not recognizable as the name of an internal parameter of EZMAP.

- **MAPGTR - UNCLEARED PRIOR ERROR**

  When MAPGTR was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPGTR - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPGTR is not recognizable as the name of an internal parameter of EZMAP.

- **MAPINT - ANGULAR LIMITS TOO GREAT**

  In a preceding call to MAPSET, JLTS = 'AN' was used, and the accompanying values of PLM1, PLM2, PLM3, and/or PLM4 were too large for the selected projection type.

- **MAPINT - ATTEMPT TO USE NON-EXISTENT PROJECTION**

  Somehow, an internal variable that says what projection type to use has been given an illegal value. Under normal conditions, that's pretty hard to do; it may be that core has been overstored.

- **MAPINT - MAP HAS ZERO AREA**

  The current internal parameters specify a map with zero area.

- **MAPINT - MAP LIMITS INAPPROPRIATE**

  This can happen when you select a Lambert conformal conic projection and then try to use angular limits, which is a no-no. It can also happen when you try to specify the map limits using a point that is not projectable under the current projection.

- **MAPINT - UNCLEARED PRIOR ERROR**

  When MAPINT was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPIO - EOF ENCOUNTERED IN OUTLINE DATASET**

  MAPIO is an internal routine that is used to read the geographic outline dataset. Something is wrong with the dataset.

- **MAPIO - ERROR ON READ OF OUTLINE DATASET**

  MAPIO is an internal routine that is used to read the geographic outline dataset. Something is wrong with the dataset.

- **MAPIQ - UNCLEARED PRIOR ERROR**

When MAPIQ was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPIQA - UNCLEARED PRIOR ERROR**

  When MAPIQA was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPIQM - UNCLEARED PRIOR ERROR**

  When MAPIQM was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPIT - UNCLEARED PRIOR ERROR**

  When MAPIT was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPITA - UNCLEARED PRIOR ERROR**

  When MAPITA was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPITM - UNCLEARED PRIOR ERROR**

  When MAPITM was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPLBL - UNCLEARED PRIOR ERROR**

  When MAPLBL was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPLMB - UNCLEARED PRIOR ERROR**

  When MAPLMB was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPLOT - UNCLEARED PRIOR ERROR**

  When MAPLOT was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPPOS - ARGUMENTS ARE INCORRECT**

  In a call to MAPPOS, either 1) one of the arguments is outside the range from 0. to 1., or 2) argument 1 is greater than or equal to argument 2, or 3) argument 3 is greater than or equal to argument 4.

- **MAPPOS - UNCLEARED PRIOR ERROR**

When MAPPOS was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPROJ - UNCLEARED PRIOR ERROR**

  When MAPROJ was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPROJ - UNKNOWN PROJECTION NAME**

  The first argument in a call to MAPROJ is not recognizable as the name of one of the projections provided by EZMAP.

- **MAPRS - UNCLEARED PRIOR ERROR**

  When MAPRS was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPRST - EOF ON READ**

  An EOF was encountered while attempting to read back parameter values written by MAPSAV. Remember that the user is responsible for all positioning of the file used.

- **MAPRST - ERROR ON READ**

  A FORTRAN read error occurred while attempting to read back parameter values written by MAPSAV. Remember that the user is responsible for all positioning of the file used.

- **MAPRST - UNCLEARED PRIOR ERROR**

  When MAPRST was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSAV - ERROR ON WRITE**

  A FORTRAN write error occurred. This most likely means that the user-provided logical file number is incorrect.

- **MAPSAV - UNCLEARED PRIOR ERROR**

  When MAPSAV was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSET - UNCLEARED PRIOR ERROR**

  When MAPSET was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSET - UNKNOWN MAP AREA SPECIFIER**

  The first argument in a call to MAPSET is not recognizable as the name of one of the map area specification methods provided by EZMAP.

- **MAPSTC - UNCLEARED PRIOR ERROR**

  When MAPSTC was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSTC - UNKNOWN OUTLINE NAME**

  The second argument in a call to MAPSTC is not recognizable as the name of one of the outline datasets provided by EZMAP.

- **MAPSTC - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPSTC is not recognizable as the name of an internal parameter of EZMAP.

- **MAPSTI - UNCLEARED PRIOR ERROR**

  When MAPSTI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSTI - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPSTI is not recognizable as the name of an internal parameter of EZMAP.

- **MAPSTL - UNCLEARED PRIOR ERROR**

  When MAPSTL was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSTL - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPSTL is not recognizable as the name of an internal parameter of EZMAP.

- **MAPSTR - UNCLEARED PRIOR ERROR**

  When MAPSTR was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPSTR - UNKNOWN PARAMETER NAME**

  The first argument in a call to MAPSTR is not recognizable as the name of an internal parameter of EZMAP.

- **MAPTRA - UNCLEARED PRIOR ERROR**

  When MAPTRA was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPTRI - UNCLEARED PRIOR ERROR**

  When MAPTRI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPTRN - ATTEMPT TO USE NON-EXISTENT PROJECTION**

  Somehow, an internal variable that says what projection type to use has been given an illegal value. Under normal conditions, that's pretty hard to do; it may be that core has been overstored.

- **MAPTRN - UNCLEARED PRIOR ERROR**

  When MAPTRN was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MAPVEC - UNCLEARED PRIOR ERROR**

  When MAPVEC was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPGETC - UNCLEARED PRIOR ERROR**

  When MPGETC was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPGETI - UNCLEARED PRIOR ERROR**

  When MPGETI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPGETL - UNCLEARED PRIOR ERROR**

  When MPGETL was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPGETR - UNCLEARED PRIOR ERROR**

  When MPGETR was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPSETC - UNCLEARED PRIOR ERROR**

  When MPSETC was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPSETI - UNCLEARED PRIOR ERROR**

  When MPSETI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPSETL - UNCLEARED PRIOR ERROR**

  When MPSETL was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **MPSETR - UNCLEARED PRIOR ERROR**

  When MPSETR was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.
- **SUPCON - UNCLEARED PRIOR ERROR**

  When SUPCON was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.
- **SUPMAP - UNCLEARED PRIOR ERROR**

  When SUPMAP was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

6.0 EXAMPLES

On a Unix system on which NCAR Graphics has been installed, the command “ncargex” may be used to acquire the code for and run the following examples illustrating various capabilities of EZMAP:

- Example ”mpex01” shows the United States on a Lambert conformal conic.
- Example ”mpex02” shows Africa on a stereographic projection, with an elliptical perimeter.
- Example ”mpex03” shows simplified continental outlines on a Mercator projection.It uses a version of MAPEOD which scrubs all but the principal land masses. The left and right area identifiers are used to determine which segments to keep and which to scrub. Segment-number information, obtained by running the program shown in example 9, could have been used instead.
- Example “mpex04” shows the world in three parts, using stereographic projections for the poles and a Mercator projection for the equatorial belt.
- Example “mpex05” shows maximal-area plots for all types of EZMAP projections.
- Example “mpex06” shows the effect of the rotation angle ROTA on a satellite-view projection.
- Example “mpex07” presents a scheme for labelling the meridians on certain types of maps. It is difficult to write an algorithm for doing this in general.
- Example “mpex08” demonstrates how the routine MAPUSR is used.
- Example “mpex09” shows the numbering of segments in the outline dataset 'CO' and presents a program which may be used to obtain such plots for the other outline datasets. This example is somewhat out of date, since the segment numbers are only of use in a user version of the routine named MAPEOD and it is probably better now to

use the left and right area identifiers, rather than the segment numbers, in deciding which outline segments to keep and which to omit.The technique used to break the maps into rectangular pieces and present each piece individually may be useful in its own right.

- Example “mpex10” shows how to use the routine MAPTRI and the cell-array capability of GKS to obtain what is essentially a contour map of a data field defined on the surface of the globe.
- Example “mpexfi” (the “final” example for EZMAP itself) shows how to draw lines, defined by points for which one has the latitudes and longitudes, on a map produced by EZMAP.
- Example “tezmap” produces a number of frames; it is intended to demonstrate minimal functioning of EZMAP.

Many of the above example programs uses a subroutine named BNDARY to draw a boundary line indicating where the edge of the plotter frame is. This subroutine is not a part of EZMAP or of the NCAR plot package. When you run one of these, the code for the subroutine BNDARY will be put in a file named “mpexcc.f”.

The following examples illustrate the use of routines from the package EZMAPA:

- Example “eezmpa” shows a solid-colored map of the world on a Mercator projection, with lines of latitude and longitude over ocean only.
- Example “tezmpa” shows a solid-colored map of a portion of Northern Europe, with lines of latitude and longitude over ocean only.

The following CONPACK example illustrates the use of the satellite-view projection and various routines from the package EZMAPA:

- Example “cpex10” shows a view of the western United States as seen from a satellite above Washington, D.C. Within seven degrees of Boulder, Colorado, color-filled contour bands are shown. The EZMAPA routine MAPBLM is used with an appropriate area map, allowing state boundaries to be in black over the contoured area and in white elsewhere. The EZMAP routine MAPGRM is used with an appropriate area map, allowing 1-degree grid lines to be drawn outside the contoured area, but not inside it. Mapping capabilities of PLOTCHAR are used to create state labels that appear to be written on the surface of the globe and projected along with it.

7.0 AREA IDENTIFIERS

There are four EZMAP outline datasets, indexed as follows:

Index	Name	Contents
1	'CO'	Continental outlines only.
2	'US'	U.S. state outlines only.

Index	Name	Contents
3	'PS'	Continental, U.S. state, and international outlines.
4	'PO'	Continental and international outlines.

Associated with each outline segment in these datasets are two integers identifying, respectively, the areas to the left and right of the outline segment. These “area identifiers” may be used, in a user version of the subroutine MAPEOD, to cull unwanted segments from a map; they also allow the user to produce solid-color maps.

The following table shows the association between area identifiers and the names of the areas they identify. Column 1 contains an outline dataset index, column 2 an area identifier, column 3 a suggested color index for the area (which will ensure that adjacent areas have different colors), and column 4 the name of the area.

```
1     1  2  Antarctica
1     2  1  Oceans
1     3  2  Africa/Eurasia
1     4  2  USSR - Wrangel Island
1     5  2  Alaska - Atka Island
1     6  2  New Zealand - Chatham Islands
1     7  2  Samoa - Savaii
1     8  2  Samoa - Upolu
1     9  2  Alaska - Saint Lawrence Island
1    10  2  Alaska - Umnak Island
1    11  2  North and South America
1    12  2  Alaska - Unalaska Island
1    13  2  Alaska - Nunivak Island
1    14  2  Antarctica - Roosevelt Island
1    15  2  Hawaiian Islands - Niihau
1    16  2  Hawaiian Islands - Kauai
1    17  2  Hawaiian Islands - Oahu
1    18  2  Hawaiian Islands - Maui
1    19  2  Hawaiian Islands - Hawaii
1    20  2  Alaska - Kodiak Island
1    21  2  Antarctica - Guest Island
1    22  2  Tahiti
1    23  2  Canada - Queen Charlotte Islands
1    24  2  Canada - Vancouver Island
1    25  2  Canada - Banks Island
1    26  1  Canada - Great Bear Lake
1    27  2  Canada - Prince Patrick Island
1    28  2  Canada - Eglinton Island
1    29  2  Canada - Victoria Island
1    30  2  Canada - Melville Island
1    31  1  Canada - Great Slave Lake
1    32  2  Canada - Brock Island
1    33  2  Canada - Borden Island
1    34  2  Canada - Mackenzie King Island
1    35  1  Canada - Lake Athabasca
1    36  2  Antarctica - Bear Island
1    37  2  Canada - Ellef Ringnes and Amund Ringnes Islands
1    38  2  Canada - Lougheed Island
1    39  2  Canada - Bathurst Island
1    40  2  Canada - Byam Martin Island
1    41  1  Canada - Reindeer Lake
1    42  2  Canada - Prince of Wales Island
1    43  1  Canada - Lake Winnipegosis
1    44  1  Canada - Lake Manitoba
1    45  2  Canada - Meighen Island
1    46  1  Canada - Lake Winnipeg
```

```
 1    47  2  Canada - Cornwall Island
 1    48  2  Canada - Cornwallis Island
 1    49  2  Canada - Devon Island
 1    50  2  Canada - Axel Heiberg Island
 1    51  1  Lake Superior
 1    52  2  Galapagos Islands
 1    53  2  Canada - Ellesmere Island
 1    54  2  Canada - Baffin Island
 1    55  1  Lakes Michigan and Huron
 1    56  2  Canada - Southhampton Island
 1    57  2  Cuba
 1    58  2  Canada - Coats Island
 1    59  1  Lake Erie
 1    60  2  Canada - Manitoulin Island
 1    61  2  Cuba - Isla de Pinos
 1    62  2  Canada - Akimiski Island
 1    63  2  Canada - Bylot Island
 1    64  2  Canada - Mansel Island
 1    65  2  Canada - Belcher Islands
 1    66  1  Lake Ontario
 1    67  2  Bahamas
 1    68  2  Bahamas
 1    69  2  Jamaica
 1    70  2  Bahamas
 1    71  2  Canada - Prince Charles Island
 1    72  2  Antarctica - Alexander Island
 1    73  2  Antarctica - Charcot Island
 1    74  2  Tierra del Fuego
 1    75  2  Haiti and the Dominican Republic
 1    76  2  Isla de Chiloe
 1    77  2  Long Island
 1    78  2  Turks and Caicos Islands
 1    79  2  Greenland
 1    80  2  Curacao
 1    81  2  Puerto Rico
 1    82  2  Canada - Anticosti Island
 1    83  2  Antarctica - Anvers Island
 1    84  2  Canada - Prince Edward Island
 1    85  2  Antarctica - Brabant Island
 1    86  2  Trinidad and Tobago
 1    87  2  Guadeloupe - west
 1    88  2  Guadeloupe - east
 1    89  2  Falkland Islands
 1    90  2  Martinique
 1    91  2  Antarctica - Livingston Island
 1    92  2  Antarctica - King George Island
 1    93  2  Newfoundland
 1    94  2  Antarctica - Joinville Island
 1    95  2  Antarctica - Coronation Island
 1    96  2  South Georgia Island
 1    97  2  Iceland
 1    98  2  Greenland - off-shore island
 1    99  2  Africa/Eurasia
 1   100  2  Canary Islands
 1   101  2  Canary Islands
 1   102  2  Canary Islands
 1   103  2  Canary Islands
 1   104  2  Ireland
 1   105  2  Faeroe Islands
 1   106  2  Outer Hebrides
 1   107  2  Scotland/England/Wales
 1   108  2  Mallorca
 1   109  2  Sardinia
 1   110  2  Macias Nguema Biyogo (African island, once called Fernando Po)
 1   111  2  Corsica
```

```
 1   112  2  Denmark - Fyn
 1   113  2  Denmark - Lolland
 1   114  2  Svalbard
 1   115  2  Denmark - Sjaelland
 1   116  2  Sicily
 1   117  1  Lake Chad
 1   118  2  Gotland
 1   119  2  Greece
 1   120  2  USSR - Ostrov Saaremaa
 1   121  2  USSR - Ostrov Hiiumaa
 1   122  2  Crete
 1   123  1  Black Sea
 1   124  2  Rhodes
 1   125  1  Lake Tanganyika
 1   126  1  USSR - Ozero Ladozhskoye (Lake Ladoga)
 1   127  1  Lake Victoria
 1   128  2  Cyprus
 1   129  1  Lake Nyasa
 1   130  1  USSR - Azov Sea
 1   131  1  USSR - Onazhskoye Ozero
 1   132  2  Madagascar
 1   133  2  USSR - Franz Josef Land
 1   134  1  Caspian Sea
 1   135  2  USSR - Ostrov Kolguyev
 1   136  2  USSR - Novaya Zemlya
 1   137  2  USSR - Franz Josef Land
 1   138  2  Yemen (Aden) - Socotra (island)
 1   139  2  USSR - Franz Josef Land
 1   140  2  Reunion Island
 1   141  2  USSR - Franz Josef Land
 1   142  2  Mauritius
 1   143  1  USSR - Aral Sea
 1   144  2  USSR - Franz Josef Land
 1   145  2  USSR - Franz Josef Land
 1   146  2  Ile Kerguelen
 1   147  1  USSR - Lake Balkhash (west)
 1   148  1  USSR - Lake Balkhash (east)
 1   149  2  Sri Lanka (Ceylon)
 1   150  2  USSR - Ostrov Oktyabr'skoy Revolutsii, Komsomolets, Pioner
 1   151  2  Indonesia - Sumatra
 1   152  2  Indonesia - Nias
 1   153  2  Indonesia - Siberut
 1   154  2  USSR - Ostrov Bol'shevik
 1   155  1  USSR - Lake Baikal
 1   156  2  Indonesia - Bangka
 1   157  2  Indonesia - Java
 1   158  2  Indonesia - Belitung
 1   159  2  China - Hainan Dao  (island)
 1   160  2  Borneo
 1   161  2  USSR - Ostrov Bol'shoy Begichev
 1   162  2  Indonesia - Bawean
 1   163  2  Australia
 1   164  2  Indonesia - Bali
 1   165  2  Indonesia - Sumbawa and Lombok
 1   166  2  Philippines - Palawan
 1   167  2  Indonesia - Sulawesi (Celebes)
 1   168  2  Indonesia - Sumba
 1   169  2  Indonesia - Flores
 1   170  2  Phillipines - Luzon
 1   171  2  Taiwan
 1   172  2  Phillipines - Mindoro
 1   173  2  Phillipines - Panay, Negros, and Bohol
 1   174  2  Phillipines - Mindanao
 1   175  2  Indonesia - island
 1   176  2  Indonesia - Timor
```

```
1   177  2  Indonesia - Alor
1   178  2  Phillipines - Samar
1   179  2  Indonesia - Wetar
1   180  2  Indonesia - Buru
1   181  2  Indonesia - Obi
1   182  2  Indonesia - Halmahera
1   183  2  Indonesia - Ceram
1   184  2  Japan - Kyushu
1   185  2  Australia - Melville and Bathurst Islands
1   186  2  Indonesia - Waigeo
1   187  2  New Guinea
1   188  2  Japan - Honshu
1   189  2  Indonesia - Kepulauan Tanimbar
1   190  2  Japan - Shikoku
1   191  2  Indonesia - Kepulauan Aru
1   192  2  Indonesia - Biak and Kepulauan Schouten
1   193  2  Australia - Kangaroo Island
1   194  2  USSR - Ostrov Kotel'nyy and Ostrov Faddeyevskiy
1   195  2  USSR - Shantarskiye Ostrova
1   196  2  Japan - Hokkaido
1   197  2  USSR - Ostrov Malyy Lyakhovskiy
1   198  2  USSR - Ostrov Bol'shoy Lyakhovskiy
1   199  2  USSR - Ostrov Sakhalin
1   200  2  Australia - Tasmania
1   201  2  USSR - Ostrov Novaya Sibir'
1   202  2  USSR - Ostrov Iturup
1   203  2  Bismarck Archipelago - New Britain
1   204  2  USSR - Ostrov Urup
1   205  2  Bismarck Archipelago - New Hanover
1   206  2  Bismarck Archipelago - New Ireland
1   207  2  Solomon Islands - Bougainville
1   208  2  USSR - Ostrov Paramushir
1   209  2  Solomon Islands - Guadalcanal
1   210  2  Solomon Islands - San Cristobal
1   211  2  USSR - Ostrov Karaginskiy
1   212  2  New Caledonia (island)
1   213  2  Antarctica - Ross Island
1   214  2  New Zealand - South Island
1   215  2  New Hebrides - Espiritu Santo
1   216  2  New Hebrides - Malekula
1   217  2  New Zealand - Stewart Island
1   218  2  Alaska - Attu Island
1   219  2  New Zealand - North Island
1   220  2  New Zealand - North Island
1   221  2  Fiji - Viti Levu
1   222  2  Fiji - Vanua Levu
2   223  7  Border
2   224  2  Washington
2   225  3  Oregon
2   226  4  California
2   227  2  Washington - island
2   228  2  Washington - island
2   229  2  Washington - island
2   230  2  Washington - island
2   231  2  Washington - island
2   232  2  Washington - island
2   233  2  Washington - island
2   234  4  California - San Miguel Island
2   235  4  California - Santa Rosa Island
2   236  1  Lake Tahoe
2   237  5  Nevada
2   238  4  California - Santa Cruz Island
2   239  4  California - San Nicholas Island
2   240  4  California - San Clemente Island
2   241  4  California - Santa Catalina Island
```

```
2   242  6  Idaho
2   243  5  Montana
2   244  1  Lake Mead (west)
2   245  2  Arizona
2   246  1  Lake Mead (east)
2   247  3  Utah
2   248  1  Great Salt Lake
2   249  4  Wyoming
2   250  5  New Mexico
2   251  6  Colorado
2   252  2  Texas
2   253  5  Nebraska
2   254  2  South Dakota
2   255  3  North Dakota
2   256  3  Oklahoma
2   257  4  Kansas
2   258  2  Texas - island
2   259  2  Texas - Padre Island
2   260  4  Minnesota
2   261  1  Lake Texoma
2   262  6  Iowa
2   263  2  Missouri
2   264  1  Minnesota - Lake of the Woods
2   265  1  Minnesota - Shoal Lake
2   266  4  Minnesota - Lake of the Woods - land west of lake
2   267  2  Texas - island
2   268  4  Minnesota - Lake of the Woods - island
2   269  4  Minnesota - Lake of the Woods - island
2   270  5  Arkansas
2   271  4  Minnesota - Lake of the Woods - island
2   272  4  Louisiana
2   273  1  Louisiana - Sabine Lake
2   274  1  Minnesota - Rainy Lake
2   275  1  Louisiana - Calcasieu Lake
2   276  1  Minnesota - Namakan Lake
2   277  3  Wisconsin
2   278  1  Louisiana - Grand Lake
2   279  4  Minnesota - little piece
2   280  1  Louisiana - White Lake
2   281  1  Lake Superior
2   282  4  Louisiana - Marsh Island
2   283  2  Mississippi
2   284  4  Illinois
2   285  4  Louisiana - island
2   286  2  Lake Superior - Madeline Island
2   287  4  Louisiana - island
2   288  2  Lake Superior - Oak Island
2   289  2  Lake Superior - Stockton Island
2   290  1  Louisiana - Lake Maurepas
2   291  4  Louisiana - Timbalier Island
2   292  2  Lake Superior - Outer Island
2   293  2  Michigan (upper)
2   294  1  Louisiana - Lake Pontchartrain
2   295  3  Tennessee
2   296  4  Louisiana - island
2   297  4  Louisiana - island
2   298  4  Louisiana - island
2   299  6  Kentucky
2   300  4  Louisiana - island
2   301  4  Louisiana - island
2   302  4  Louisiana - island
2   303  4  Louisiana - island
2   304  4  Louisiana - island
2   305  4  Louisiana - island
2   306  4  Louisiana - island
```

2	307	2	Lake Superior - Isle Royale
2	308	2	Lake Superior - Pie Island
2	309	4	Louisiana - Cat Island
2	310	5	Alabama
2	311	5	Indiana
2	312	1	Lakes Michigan and Huron
2	313	4	Florida
2	314	2	Lake Michigan - Washington Island
2	315	2	Michigan (lower)
2	316	2	Lake Superior - Grand Island
2	317	2	Lake Michigan - Manitou Island
2	318	2	Lake Superior - Michipicoten Island
2	319	1	Florida - lake
2	320	6	Georgia
2	321	2	Lake Michigan - Beaver Island
2	322	4	Florida - island
2	323	1	Lake Seminole
2	324	4	Ohio
2	325	2	Lake Huron - Bois Blanc Island
2	326	2	Lake Superior - island
2	327	1	Sault Ste. Marie
2	328	4	North Carolina
2	329	1	Lake George
2	330	2	Lake Huron - Drummond Island
2	331	2	Virginia
2	332	2	Lake Huron - Cockburn Island
2	333	1	Lake Erie
2	334	2	South Carolina
2	335	2	Lake Huron - Manitoulin Island
2	336	1	Lake St. Clair
2	337	2	Lake Huron - Barrie Island
2	338	3	West Virginia
2	339	1	Clark Hill Reservoir
2	340	4	Florida - Sanibel Island
2	341	4	Florida - Pine Island
2	342	2	Lake Huron - Fitzwilliam Island
2	343	4	Florida - Keys
2	344	4	Florida - Keys
2	345	4	Florida - Keys
2	346	4	Florida - Keys
2	347	1	Florida - Lake Okeechobee
2	348	4	Florida - island
2	349	4	Florida - island
2	350	4	Florida - island
2	351	4	Florida - island
2	352	6	Pennsylvania
2	353	4	Florida - island
2	354	2	Lake Erie - Long Point (land)
2	355	2	Lake Huron - Christian Island
2	356	3	New York
2	357	1	Lake Ontario
2	358	4	Maryland
2	359	4	Bahamas - Grand Bahama
2	360	4	Bahamas - Andros
2	361	4	North Carolina - Cape Fear
2	362	4	Bahamas - Andros
2	363	4	Bahamas - Andros
2	364	4	Bahamas - Great Abaco and Little Abaco
2	365	4	Bahamas - Andros
2	366	4	Bahamas - Grand Bahama
2	367	4	Bahamas - Grand Bahama
2	368	4	Bahamas - Andros
2	369	4	Bahamas - Mores Island
2	370	4	Bahamas - New Providence Island
2	371	2	Lake Ontario - island

```
   2   372  4  Bahamas - Eleuthera Island
   2   373  4  Bahamas - Eleuthera Island
   2   374  4  Bahamas - Eleuthera Island
   2   375  2  Lake Ontario - island
   2   376  4  Maryland - island
   2   377  2  Lake Ontario - island
   2   378  4  Maryland - island
   2   379  2  Virginia - island
   2   380  4  Maryland - island
   2   381  4  Maryland - island
   2   382  2  Virginia
   2   383  4  North Carolina - island
   2   384  4  North Carolina - island
   2   385  2  Virginia - island
   2   386  1  Maryland - inlet
   2   387  2  Virginia - island
   2   388  4  North Carolina - island
   2   389  2  Virginia
   2   390  2  Virginia
   2   391  4  North Carolina - island
   2   392  4  North Carolina - island
   2   393  5  Delaware
   2   394  4  Bahamas - Cat Island
   2   395  4  North Carolina - island
   2   396  4  North Carolina - island
   2   397  4  North Carolina - island
   2   398  4  North Carolina - island
   2   399  4  North Carolina - island
   2   400  2  New Jersey
   2   401  2  Virginia - island
   2   402  1  Delaware - bay
   2   403  4  Maryland
   2   404  4  Maryland
   2   405  2  New Jersey - island
   2   406  3  Staten Island
   2   407  1  New Jersey - Newark Bay
   2   408  3  New York - Long Island
   2   409  5  Connecticut
   2   410  4  Massachusetts
   2   411  1  Lake Champlain
   2   412  2  Vermont
   2   413  2  Lake Champlain - Grand Isle
   2   414  1  New Hampshire - lake
   2   415  1  Canada - lake
   2   416  1  New Hampshire - lake
   2   417  1  Long Island - Great South Bay
   2   418  6  New Hampshire
   2   419  3  New York - Shelter Island
   2   420  3  New York - Fishers Island
   2   421  6  Rhode Island
   2   422  6  Rhode Island - Block Island
   2   423  6  Rhode Island - island
   2   424  6  Rhode Island - island
   2   425  6  Rhode Island - island
   2   426  1  Massachusetts - bay
   2   427  5  Maine
   2   428  4  Massachusetts - Marthas Vineyard
   2   429  4  Massachusetts - Nantucket Island
   2   430  1  Massachusetts - inlet
   2   431  5  Maine - island
   2   432  5  Maine - island
   2   433  5  Maine - island
   2   434  5  Maine - island
   2   435  5  Maine - island
   2   436  5  Maine - island
```

2	437	5	Maine - island
2	438	5	Maine - island
2	439	5	Maine - island
3	440	7	Antarctica
3	441	1	Oceans
3	442	6	USSR
3	443	6	USSR - Wrangel Island
3	444	3	Alaska - Atka Island
3	445	6	New Zealand - Chatham Islands
3	446	3	Samoa - Savaii
3	447	3	Samoa - Upolu
3	448	3	Alaska - Saint Lawrence Island
3	449	3	Alaska - Umnak Island
3	450	3	Alaska
3	451	3	Alaska - Unalaska Island
3	452	3	Alaska - Nunivak Island
3	453	7	Antarctica - Roosevelt Island
3	454	3	Hawaiian Islands - Niihau
3	455	3	Hawaiian Islands - Kauai
3	456	3	Hawaiian Islands - Oahu
3	457	3	Hawaiian Islands - Maui
3	458	3	Hawaiian Islands - Hawaii
3	459	3	Alaska - Kodiak Island
3	460	7	Antarctica - Guest Island
3	461	3	Tahiti
3	462	6	Canada
3	463	6	Canada - Queen Charlotte Islands
3	464	6	Canada - Vancouver Island
3	465	6	Canada - Banks Island
3	466	1	Canada - Great Bear Lake
3	467	2	US - Washington
3	468	3	US - Oregon
3	469	4	US - California
3	470	6	Canada - Prince Patrick Island
3	471	2	US - Washington - island
3	472	2	US - Washington - island
3	473	2	US - Washington - island
3	474	2	US - Washington - island
3	475	2	US - Washington - island
3	476	2	US - Washington - island
3	477	2	US - Washington - island
3	478	4	US - California - San Miguel Island
3	479	4	US - California - Santa Rosa Island
3	480	1	US - Lake Tahoe
3	481	5	US - Nevada
3	482	4	US - California - Santa Cruz Island
3	483	4	US - California - San Nicholas Island
3	484	6	Canada - Eglinton Island
3	485	6	Canada - Victoria Island
3	486	4	US - California - San Clemente Island
3	487	4	US - California - Santa Catalina Island
3	488	6	Canada - Melville Island
3	489	6	US - Idaho
3	490	1	Canada - Great Slave Lake
3	491	6	Mexico
3	492	5	US - Montana
3	493	6	Canada - Brock Island
3	494	1	US - Lake Mead (west)
3	495	2	US - Arizona
3	496	1	US - Lake Mead (east)
3	497	3	US - Utah
3	498	6	Canada - Borden Island
3	499	6	Canada - Mackenzie King Island
3	500	1	US - Great Salt Lake
3	501	1	Canada - Lake Athabasca

```
  3   502  4  US - Wyoming
  3   503  6  Antarctica - Bear Island
  3   504  5  US - New Mexico
  3   505  6  US - Colorado
  3   506  2  US - Texas
  3   507  6  Canada - Ellef Ringnes and Amund Ringnes Islands
  3   508  6  Canada - Lougheed Island
  3   509  6  Canada - Bathurst Island
  3   510  6  Canada - Byam Martin Island
  3   511  5  US - Nebraska
  3   512  2  US - South Dakota
  3   513  3  US - North Dakota
  3   514  1  Canada - Reindeer Lake
  3   515  3  US - Oklahoma
  3   516  6  Canada - Prince of Wales Island
  3   517  4  US - Kansas
  3   518  1  Canada - Lake Winnipegosis
  3   519  1  Canada - Lake Manitoba
  3   520  6  Canada - Meighen Island
  3   521  1  Canada - Lake Winnipeg
  3   522  2  US - Texas - island
  3   523  2  US - Texas - Padre Island
  3   524  4  US - Minnesota
  3   525  6  Canada - Cornwall Island
  3   526  1  US - Lake Texoma
  3   527  6  US - Iowa
  3   528  6  Canada - Cornwallis Island
  3   529  6  Canada - Devon Island
  3   530  2  US - Missouri
  3   531  1  US - Minnesota - Lake of the Woods
  3   532  1  US - Minnesota - Shoal Lake
  3   533  4  US - Minnesota - Lake of the Woods - land west of lake
  3   534  2  US - Texas - island
  3   535  6  Canada - Axel Heiberg Island
  3   536  4  US - Minnesota - Lake of the Woods - island
  3   537  4  US - Minnesota - Lake of the Woods - island
  3   538  5  US - Arkansas
  3   539  4  US - Minnesota - Lake of the Woods - island
  3   540  4  US - Louisiana
  3   541  1  US - Louisiana - Sabine Lake
  3   542  1  US - Minnesota - Rainy Lake
  3   543  1  US - Louisiana - Calcasieu Lake
  3   544  1  US - Minnesota - Namakan Lake
  3   545  3  US - Wisconsin
  3   546  1  US - Louisiana - Grand Lake
  3   547  4  US - Minnesota - little piece
  3   548  1  US - Louisiana - White Lake
  3   549  3  Guatemala
  3   550  1  US - Lake Superior
  3   551  4  US - Louisiana - Marsh Island
  3   552  2  US - Mississippi
  3   553  3  Galapagos Islands
  3   554  4  US - Illinois
  3   555  4  US - Louisiana - island
  3   556  2  US - Lake Superior - Madeline Island
  3   557  4  US - Louisiana - island
  3   558  2  US - Lake Superior - Oak Island
  3   559  2  US - Lake Superior - Stockton Island
  3   560  1  US - Louisiana - Lake Maurepas
  3   561  4  US - Louisiana - Timbalier Island
  3   562  6  Canada - Ellesmere Island
  3   563  2  US - Lake Superior - Outer Island
  3   564  2  US - Michigan (upper)
  3   565  1  US - Louisiana - Lake Pontchartrain
  3   566  3  US - Tennessee
```

```
3   567  4  El Salvador
3   568  6  Canada - Baffin Island
3   569  4  US - Louisiana - island
3   570  4  US - Louisiana - island
3   571  4  US - Louisiana - island
3   572  5  Honduras
3   573  6  US - Kentucky
3   574  4  US - Louisiana - island
3   575  4  US - Louisiana - island
3   576  4  US - Louisiana - island
3   577  4  US - Louisiana - island
3   578  4  US - Louisiana - island
3   579  4  US - Louisiana - island
3   580  4  US - Louisiana - island
3   581  2  US - Lake Superior - Isle Royale
3   582  2  Belize
3   583  2  US - Lake Superior - Pie Island
3   584  4  US - Louisiana - Cat Island
3   585  5  US - Alabama
3   586  5  US - Indiana
3   587  1  US - Lakes Michigan and Huron
3   588  6  Nicaragua
3   589  4  US - Florida
3   590  6  Canada - Southhampton Island
3   591  2  US - Lake Michigan - Washington Island
3   592  2  US - Michigan (lower)
3   593  2  US - Lake Superior - Grand Island
3   594  2  US - Lake Michigan - Manitou Island
3   595  2  US - Lake Superior - Michipicoten Island
3   596  2  Costa Rica
3   597  1  US - Florida - lake
3   598  6  US - Georgia
3   599  2  US - Lake Michigan - Beaver Island
3   600  4  US - Florida - island
3   601  1  US - Lake Seminole
3   602  4  Cuba
3   603  4  US - Ohio
3   604  2  US - Lake Huron - Bois Blanc Island
3   605  2  US - Lake Superior - island
3   606  1  US - Sault Ste. Marie
3   607  4  US - North Carolina
3   608  1  US - Lake George
3   609  2  US - Lake Huron - Drummond Island
3   610  6  Canada - Coats Island
3   611  2  US - Virginia
3   612  2  US - Lake Huron - Cockburn Island
3   613  1  US - Lake Erie
3   614  2  US - South Carolina
3   615  2  US - Lake Huron - Manitoulin Island
3   616  4  Cuba - Isla de Pinos
3   617  5  Panama
3   618  1  US - Lake St. Clair
3   619  2  US - Lake Huron - Barrie Island
3   620  3  US - West Virginia
3   621  1  US - Clark Hill Reservoir
3   622  4  US - Florida - Sanibel Island
3   623  4  US - Florida - Pine Island
3   624  6  Canada - Akimiski Island
3   625  2  US - Lake Huron - Fitzwilliam Island
3   626  4  US - Florida - Keys
3   627  4  US - Florida - Keys
3   628  4  US - Florida - Keys
3   629  4  US - Florida - Keys
3   630  5  Peru
3   631  1  US - Florida - Lake Okeechobee
```

```
3   632  3  Ecuador
3   633  4  US - Florida - island
3   634  6  Canada - Bylot Island
3   635  4  US - Florida - island
3   636  4  US - Florida - island
3   637  4  US - Florida - island
3   638  6  US - Pennsylvania
3   639  4  US - Florida - island
3   640  2  US - Lake Erie - Long Point (land)
3   641  2  US - Lake Huron - Christian Island
3   642  6  Canada - Mansel Island
3   643  6  Canada - Belcher Islands
3   644  3  US - New York
3   645  1  US - Lake Ontario
3   646  4  US - Maryland
3   647  2  Colombia
3   648  4  US - Bahamas - Grand Bahama
3   649  4  US - Bahamas - Andros
3   650  5  Jamaica
3   651  4  US - North Carolina - Cape Fear
3   652  4  US - Bahamas - Andros
3   653  4  US - Bahamas - Andros
3   654  4  US - Bahamas - Great Abaco and Little Abaco
3   655  4  US - Bahamas - Andros
3   656  4  US - Bahamas - Grand Bahama
3   657  4  US - Bahamas - Grand Bahama
3   658  4  US - Bahamas - Andros
3   659  4  US - Bahamas - Mores Island
3   660  4  US - Bahamas - New Providence Island
3   661  6  Canada - Prince Charles Island
3   662  2  US - Lake Ontario - island
3   663  4  US - Bahamas - Eleuthera Island
3   664  4  US - Bahamas - Eleuthera Island
3   665  4  US - Bahamas - Eleuthera Island
3   666  7  Antarctica - Alexander Island
3   667  2  US - Lake Ontario - island
3   668  4  US - Maryland - island
3   669  7  Antarctica - Charcot Island
3   670  2  US - Lake Ontario - island
3   671  4  US - Maryland - island
3   672  2  US - Virginia - island
3   673  4  US - Maryland - island
3   674  4  US - Maryland - island
3   675  2  US - Virginia
3   676  4  US - North Carolina - island
3   677  4  US - North Carolina - island
3   678  2  US - Virginia - island
3   679  1  US - Maryland - inlet
3   680  2  US - Virginia - island
3   681  4  US - North Carolina - island
3   682  2  US - Virginia
3   683  2  US - Virginia
3   684  4  US - North Carolina - island
3   685  4  US - North Carolina - island
3   686  5  US - Delaware
3   687  4  US - Bahamas - Cat Island
3   688  4  US - North Carolina - island
3   689  4  US - North Carolina - island
3   690  4  US - North Carolina - island
3   691  4  US - North Carolina - island
3   692  4  US - North Carolina - island
3   693  2  Chile
3   694  2  US - New Jersey
3   695  2  US - Virginia - island
3   696  1  US - Delaware - bay
```

3	697	4	US - Maryland
3	698	4	US - Maryland
3	699	2	Chile - Tierra del Fuego
3	700	6	Haiti
3	701	2	US - New Jersey - island
3	702	2	Chile - Isla de Chiloe
3	703	3	US - Staten Island
3	704	1	US - New Jersey - Newark Bay
3	705	6	Brazil
3	706	3	US - New York - Long Island
3	707	5	US - Connecticut
3	708	4	Turks and Caicos Islands
3	709	4	US - Massachusetts
3	710	3	Venezuela
3	711	1	US - Lake Champlain
3	712	2	US - Vermont
3	713	3	Argentina
3	714	2	US - Lake Champlain - Grand Isle
3	715	1	US - New Hampshire - lake
3	716	1	Canada - lake
3	717	1	US - New Hampshire - lake
3	718	2	Greenland
3	719	1	US - Long Island - Great South Bay
3	720	3	US - New Hampshire
3	721	3	US - New York - Shelter Island
3	722	3	US - New York - Fishers Island
3	723	6	US - Rhode Island
3	724	3	Dominican Republic
3	725	6	US - Rhode Island - Block Island
3	726	6	US - Rhode Island - island
3	727	6	US - Rhode Island - island
3	728	6	US - Rhode Island - island
3	729	1	US - Massachusetts - bay
3	730	5	US - Maine
3	731	4	US - Massachusetts - Marthas Vineyard
3	732	4	US - Massachusetts - Nantucket Island
3	733	1	US - Massachusetts - inlet
3	734	4	Bolivia
3	735	3	Curacao
3	736	5	US - Maine - island
3	737	5	US - Maine - island
3	738	5	US - Maine - island
3	739	3	Argentina - Tierra del Fuego
3	740	5	US - Maine - island
3	741	5	US - Maine - island
3	742	5	US - Maine - island
3	743	5	US - Maine - island
3	744	5	US - Maine - island
3	745	5	US - Maine - island
3	746	3	Puerto Rico
3	747	6	Canada - Anticosti Island
3	748	7	Antarctica - Anvers Island
3	749	6	Canada - Prince Edward Island
3	750	7	Antarctica - Brabant Island
3	751	5	Paraguay
3	752	6	Trinidad and Tobago
3	753	3	Guadeloupe - west
3	754	3	Guadeloupe - east
3	755	4	Guyana
3	756	5	Falkland Islands
3	757	3	Martinique
3	758	7	Antarctica - Livingston Island
3	759	7	Antarctica - King George Island
3	760	6	Canada - Newfoundland
3	761	2	Uruguay

```
  3   762  5  Surinam
  3   763  7  Antarctica - Joinville Island
  3   764  3  French Guiana
  3   765  7  Antarctica - Coronation Island
  3   766  5  South Georgia Island
  3   767  7  Iceland
  3   768  2  Greenland - off-shore island
  3   769  3  Senegal
  3   770  6  Western Sahara
  3   771  4  Mauritania
  3   772  2  Canary Islands
  3   773  4  Guinea-Bissau
  3   774  2  Gambia
  3   775  2  Canary Islands
  3   776  5  Guinea
  3   777  2  Canary Islands
  3   778  2  Canary Islands
  3   779  6  Sierra Leone
  3   780  5  Morocco
  3   781  6  Mali
  3   782  3  Liberia
  3   783  4  Ireland
  3   784  4  Portugal
  3   785  2  Spain
  3   786  2  Algeria
  3   787  4  Ivory Coast
  3   788  5  Northern Ireland
  3   789  2  Faeroe Islands
  3   790  2  Outer Hebrides
  3   791  2  Scotland
  3   792  5  England
  3   793  2  Upper Volta
  3   794  3  Wales
  3   795  3  France
  3   796  5  Ghana
  3   797  6  Togo
  3   798  5  Niger
  3   799  3  Benin
  3   800  4  Andorra
  3   801  2  Mallorca
  3   802  6  Belgium
  3   803  4  Nigeria
  3   804  3  Netherlands
  3   805  4  Norway
  3   806  5  Luxembourg
  3   807  5  Switzerland
  3   808  4  West Germany
  3   809  6  Italy
  3   810  5  Tunisia
  3   811  2  Denmark - Jutland
  3   812  6  Sardinia
  3   813  3  Cameroon
  3   814  3  Macias Nguema Biyogo (African island, once called Fernando Po)
  3   815  3  Corsica
  3   816  5  Gabon
  3   817  6  Equatorial Guinea
  3   818  4  Libya
  3   819  4  Liechtenstein
  3   820  3  Austria
  3   821  2  Denmark - Fyn
  3   822  3  East Germany
  3   823  2  Denmark - Lolland
  3   824  4  Svalbard
  3   825  2  Denmark - Sjaelland
  3   826  5  Sweden
```

```
  3   827  2  Congo
  3   828  2  Namibia
  3   829  4  Angola - exclave called Cabinda
  3   830  4  Angola
  3   831  2  Czechoslovakia
  3   832  6  Zaire
  3   833  6  Sicily
  3   834  1  Lake Chad
  3   835  4  Yugoslavia
  3   836  2  Chad
  3   837  5  Poland
  3   838  5  Central African Republic
  3   839  5  Hungary
  3   840  5  South Africa
  3   841  2  Gotland
  3   842  3  Albania
  3   843  6  Malaysia - Malay Peninsula
  3   844  2  Greece
  3   845  6  Botswana
  3   846  3  Romania
  3   847  3  Finland
  3   848  2  Greece
  3   849  6  USSR - Ostrov Saaremaa
  3   850  3  Sudan
  3   851  5  Zambia
  3   852  6  USSR - Ostrov Hiiumaa
  3   853  5  Bulgaria
  3   854  2  Crete
  3   855  6  Egypt
  3   856  3  Zimbabwe
  3   857  3  Turkey
  3   858  6  Lesotho
  3   859  1  Black Sea
  3   860  2  Rhodes
  3   861  5  Rwanda
  3   862  1  Lake Tanganyika
  3   863  4  Burundi
  3   864  2  Uganda
  3   865  3  Tanzania
  3   866  1  USSR - Ozero Ladozhskoye (Lake Ladoga)
  3   867  4  Mozambique
  3   868  6  Swaziland
  3   869  1  Lake Victoria
  3   870  3  Cyprus
  3   871  2  Malawi
  3   872  4  Ethiopia
  3   873  5  Kenya
  3   874  1  Lake Nyasa
  3   875  2  Israel
  3   876  1  USSR - Azov Sea
  3   877  1  USSR - Onazhskoye Ozero
  3   878  3  Saudi Arabia
  3   879  5  Jordan
  3   880  3  Lebanon
  3   881  4  Syria
  3   882  6  Iraq
  3   883  2  Somalia
  3   884  3  Djibouti
  3   885  5  Yemen (Sana)
  3   886  3  Madagascar
  3   887  6  Yemen (Aden)
  3   888  5  Iran
  3   889  6  USSR - Franz Josef Land
  3   890  4  Unnamed neutral zone between Iraq and Saudi Arabia
  3   891  1  Caspian Sea
```

3	892	6	USSR – Ostrov Kolguyev
3	893	2	United Arab Emirates
3	894	4	Qatar
3	895	6	USSR – Novaya Zemlya
3	896	4	Oman – northern section
3	897	6	USSR – Franz Josef Land
3	898	6	Yemen (Aden) – Socotra (island)
3	899	6	USSR – Frantz Josef Land
3	900	3	Reunion Island
3	901	6	USSR – Franz Josef Land
3	902	4	Oman – southern section
3	903	3	Mauritius
3	904	1	USSR – Aral Sea
3	905	6	USSR – Franz Josef Land
3	906	3	Afghanistan
3	907	4	Pakistan
3	908	6	USSR – Franz Josef Land
3	909	2	India
3	910	3	Ile Kerguelen
3	911	1	USSR – Lake Balkhash (west)
3	912	5	China
3	913	1	USSR – Lake Balkhash (east)
3	914	4	Disputed area between India and China
3	915	5	Sri Lanka (Ceylon)
3	916	3	Nepal
3	917	3	Mongolia
3	918	4	Bangladesh
3	919	3	Bhutan
3	920	6	USSR – Ostrov Oktyabr’skoy Revolutsii, Komsomolets, Pioner
3	921	3	Burma
3	922	5	Indonesia – Sumatra
3	923	5	Indonesia – Nias
3	924	5	Thailand
3	925	5	Indonesia – Siberut
3	926	6	USSR – Ostrov Bol’shevik
3	927	6	Laos
3	928	2	Cambodia
3	929	4	Viet Nam
3	930	1	USSR – Lake Baikal
3	931	5	Indonesia – Bangka
3	932	5	Indonesia – Java
3	933	5	Indonesia – Belitung
3	934	5	China – Hainan Dao (island)
3	935	5	Indonesia – Borneo – Kalimantan
3	936	6	Malaysia – Borneo – Malaysia Timor
3	937	6	USSR – Ostrov Bol’shoy Begichev
3	938	5	Indonesia – Bawean
3	939	3	Australia
3	940	3	Borneo – Brunei
3	941	5	Indonesia – Bali
3	942	5	Indonesia – Sumbawa and Lombok
3	943	4	Philippines – Palawan
3	944	5	Indonesia – Sulawesi (Celebes)
3	945	5	Indonesia – Sumba
3	946	5	Indonesia – Flores
3	947	4	Phillipines – Luzon
3	948	2	Taiwan
3	949	4	Phillipines – Mindoro
3	950	4	Phillipines – Panay, Negros, and Bohol
3	951	4	Phillipines – Mindanao
3	952	5	Indonesia – island
3	953	5	Indonesia – Timor
3	954	5	Indonesia – Alor
3	955	4	Phillipines – Samar
3	956	2	North Korea

3	957	5	Indonesia – Wetar
3	958	5	Indonesia – Buru
3	959	4	South Korea
3	960	5	Indonesia – Obi
3	961	5	Indonesia – Halmahera
3	962	5	Indonesia – Ceram
3	963	3	Japan – Kyushu
3	964	3	Australia – Melville and Bathurst Islands
3	965	5	Indonesia – Waigeo
3	966	5	Indonesia – New Guinea – Irian Jaya
3	967	3	Japan – Honshu
3	968	5	Indonesia – Kepulauan Tanimbar
3	969	3	Japan – Shikoku
3	970	5	Indonesia – Kepulauan Aru
3	971	5	Indonesia – Biak and Kepulauan Schouten
3	972	3	Australia – Kangaroo Island
3	973	6	USSR – Ostrov Kotel’nyy and Ostrov Faddeyevskiy
3	974	6	USSR – Shantarskiye Ostrova
3	975	3	Japan – Hokkaido
3	976	6	USSR – Ostrov Malyy Lyakhovskiy
3	977	6	USSR – Ostrov Bol’shoy Lyakhovskiy
3	978	3	New Guinea – Papua
3	979	6	USSR – Ostrov Sakhalin
3	980	3	Australia – Tasmania
3	981	6	USSR – Ostrov Novaya Sibir’
3	982	6	USSR – Ostrov Iturup
3	983	3	Bismarck Archipelago – New Britain
3	984	6	USSR – Ostrov Urup
3	985	3	Bismarck Archipelago – New Hanover
3	986	3	Bismarck Archipelago – New Ireland
3	987	3	Solomon Islands – Bougainville
3	988	6	USSR – Ostrov Paramushir
3	989	3	Solomon Islands – Guadalcanal
3	990	3	Solomon Islands – San Cristobal
3	991	6	USSR – Ostrov Karaginskiy
3	992	3	New Caledonia (island)
3	993	7	Antarctica – Ross Island
3	994	6	New Zealand – South Island
3	995	3	New Hebrides – Espiritu Santo
3	996	3	New Hebrides – Malekula
3	997	6	New Zealand – Stewart Island
3	998	3	Alaska – Attu Island
3	999	6	New Zealand – North Island
3	1000	6	New Zealand – North Island
3	1001	3	Fiji – Viti Levu
3	1002	3	Fiji – Vanua Levu
4	1003	2	Kuwait
4	1004	7	Antarctica
4	1005	1	Oceans
4	1006	6	USSR
4	1007	6	USSR – Wrangel Island
4	1008	3	Alaska – Atka Island
4	1009	6	New Zealand – Chatham Islands
4	1010	3	Samoa – Savaii
4	1011	3	Samoa – Upolu
4	1012	3	Alaska – Saint Lawrence Island
4	1013	3	Alaska – Umnak Island
4	1014	3	Alaska
4	1015	3	Alaska – Unalaska Island
4	1016	3	Alaska – Nunivak Island
4	1017	7	Antarctica – Roosevelt Island
4	1018	3	Hawaiian Islands – Niihau
4	1019	3	Hawaiian Islands – Kauai
4	1020	3	Hawaiian Islands – Oahu
4	1021	3	Hawaiian Islands – Maui

```
4  1022  3  Hawaiian Islands - Hawaii
4  1023  3  Alaska - Kodiak Island
4  1024  7  Antarctica - Guest Island
4  1025  3  Tahiti
4  1026  6  Canada
4  1027  6  Canada - Queen Charlotte Islands
4  1028  6  Canada - Vancouver Island
4  1029  6  Canada - Banks Island
4  1030  1  Canada - Great Bear Lake
4  1031  3  US
4  1032  6  Canada - Prince Patrick Island
4  1033  6  Canada - Eglinton Island
4  1034  6  Canada - Victoria Island
4  1035  6  Canada - Melville Island
4  1036  1  Canada - Great Slave Lake
4  1037  6  Mexico
4  1038  6  Canada - Brock Island
4  1039  6  Canada - Borden Island
4  1040  6  Canada - Mackenzie King Island
4  1041  1  Canada - Lake Athabasca
4  1042  7  Antarctica - Bear Island
4  1043  6  Canada - Ellef Ringnes and Amund Ringnes Islands
4  1044  6  Canada - Lougheed Island
4  1045  6  Canada - Bathurst Island
4  1046  6  Canada - Byam Martin Island
4  1047  1  Canada - Reindeer Lake
4  1048  6  Canada - Prince of Wales Island
4  1049  1  Canada - Lake Winnipegosis
4  1050  1  Canada - Lake Manitoba
4  1051  6  Canada - Meighen Island
4  1052  1  Canada - Lake Winnipeg
4  1053  6  Canada - Cornwall Island
4  1054  6  Canada - Cornwallis Island
4  1055  6  Canada - Devon Island
4  1056  6  Canada - Axel Heiberg Island
4  1057  3  Guatemala
4  1058  1  Lake Superior
4  1059  3  Galapagos Islands
4  1060  6  Canada - Ellesmere Island
4  1061  4  El Salvador
4  1062  6  Canada - Baffin Island
4  1063  5  Honduras
4  1064  2  Belize
4  1065  1  Lakes Michigan and Huron
4  1066  6  Nicaragua
4  1067  6  Canada - Southhampton Island
4  1068  2  Costa Rica
4  1069  4  Cuba
4  1070  6  Canada - Coats Island
4  1071  1  Lake Erie
4  1072  6  Canada - Manitoulin Island
4  1073  4  Cuba - Isla de Pinos
4  1074  5  Panama
4  1075  6  Canada - Akimiski Island
4  1076  5  Peru
4  1077  3  Ecuador
4  1078  6  Canada - Bylot Island
4  1079  6  Canada - Mansel Island
4  1080  6  Canada - Belcher Islands
4  1081  1  Lake Ontario
4  1082  2  Colombia
4  1083  6  Bahamas
4  1084  6  Bahamas
4  1085  5  Jamaica
4  1086  6  Bahamas
```

```
4  1087  6  Canada - Prince Charles Island
4  1088  7  Antarctica - Alexander Island
4  1089  7  Antarctica - Charcot Island
4  1090  2  Chile
4  1091  2  Chile - Tierra del Fuego
4  1092  6  Haiti
4  1093  2  Chile - Isla de Chiloe
4  1094  6  Brazil
4  1095  3  Long Island
4  1096  4  Turks and Caicos Islands
4  1097  3  Venezuela
4  1098  3  Argentina
4  1099  2  Greenland
4  1100  3  Dominican Republic
4  1101  4  Bolivia
4  1102  3  Curacao
4  1103  3  Argentina - Tierra del Fuego
4  1104  3  Puerto Rico
4  1105  6  Canada - Anticosti Island
4  1106  7  Antarctica - Anvers Island
4  1107  6  Canada - Prince Edward Island
4  1108  7  Antarctica - Brabant Island
4  1109  5  Paraguay
4  1110  6  Trinidad and Tobago
4  1111  3  Guadeloupe - west
4  1112  3  Guadeloupe - east
4  1113  4  Guyana
4  1114  5  Falkland Islands
4  1115  3  Martinique
4  1116  7  Antarctica - Livingston Island
4  1117  7  Antarctica - King George Island
4  1118  6  Canada - Newfoundland
4  1119  2  Uruguay
4  1120  5  Surinam
4  1121  7  Antarctica - Joinville Island
4  1122  3  French Guiana
4  1123  7  Antarctica - Coronation Island
4  1124  5  South Georgia Island
4  1125  7  Iceland
4  1126  2  Greenland - off-shore island
4  1127  3  Senegal
4  1128  6  Western Sahara
4  1129  4  Mauritania
4  1130  2  Canary Islands
4  1131  4  Guinea-Bissau
4  1132  2  Gambia
4  1133  2  Canary Islands
4  1134  5  Guinea
4  1135  2  Canary Islands
4  1136  2  Canary Islands
4  1137  6  Sierra Leone
4  1138  5  Morocco
4  1139  6  Mali
4  1140  3  Liberia
4  1141  4  Ireland
4  1142  4  Portugal
4  1143  2  Spain
4  1144  2  Algeria
4  1145  4  Ivory Coast
4  1146  5  Northern Ireland
4  1147  2  Faeroe Islands
4  1148  2  Outer Hebrides
4  1149  2  Scotland
4  1150  5  England
4  1151  2  Upper Volta
```

```
4  1152  3  Wales
4  1153  3  France
4  1154  5  Ghana
4  1155  6  Togo
4  1156  5  Niger
4  1157  3  Benin
4  1158  4  Andorra
4  1159  2  Mallorca
4  1160  6  Belgium
4  1161  4  Nigeria
4  1162  3  Netherlands
4  1163  4  Norway
4  1164  5  Luxembourg
4  1165  5  Switzerland
4  1166  4  West Germany
4  1167  6  Italy
4  1168  5  Tunisia
4  1169  2  Denmark - Jutland
4  1170  6  Sardinia
4  1171  3  Cameroon
4  1172  3  Macias Nguema Biyogo (African island, once called Fernando Po)
4  1173  3  Corsica
4  1174  5  Gabon
4  1175  6  Equatorial Guinea
4  1176  4  Libya
4  1177  4  Liechtenstein
4  1178  3  Austria
4  1179  2  Denmark - Fyn
4  1180  3  East Germany
4  1181  2  Denmark - Lolland
4  1182  4  Svalbard
4  1183  2  Denmark - Sjaelland
4  1184  5  Sweden
4  1185  2  Congo
4  1186  2  Namibia
4  1187  4  Angola - exclave called Cabinda
4  1188  4  Angola
4  1189  2  Czechoslovakia
4  1190  6  Zaire
4  1191  6  Sicily
4  1192  1  Lake Chad
4  1193  4  Yugoslavia
4  1194  2  Chad
4  1195  5  Poland
4  1196  5  Central African Republic
4  1197  5  Hungary
4  1198  5  South Africa
4  1199  2  Gotland
4  1200  3  Albania
4  1201  6  Malaysia - Malay Peninsula
4  1202  2  Greece
4  1203  6  Botswana
4  1204  3  Romania
4  1205  3  Finland
4  1206  2  Greece
4  1207  6  USSR - Ostrov Saaremaa
4  1208  3  Sudan
4  1209  5  Zambia
4  1210  6  USSR - Ostrov Hiiumaa
4  1211  5  Bulgaria
4  1212  2  Crete
4  1213  6  Egypt
4  1214  3  Zimbabwe
4  1215  3  Turkey
4  1216  6  Lesotho
```

```
4  1217  1  Black Sea
4  1218  2  Rhodes
4  1219  5  Rwanda
4  1220  1  Lake Tanganyika
4  1221  4  Burundi
4  1222  2  Uganda
4  1223  3  Tanzania
4  1224  1  USSR - Ozero Ladozhskoye (Lake Ladoga)
4  1225  4  Mozambique
4  1226  6  Swaziland
4  1227  1  Lake Victoria
4  1228  3  Cyprus
4  1229  2  Malawi
4  1230  4  Ethiopia
4  1231  5  Kenya
4  1232  1  Lake Nyasa
4  1233  2  Israel
4  1234  1  USSR - Azov Sea
4  1235  1  USSR - Onazhskoye Ozero
4  1236  3  Saudi Arabia
4  1237  5  Jordan
4  1238  3  Lebanon
4  1239  4  Syria
4  1240  6  Iraq
4  1241  2  Somalia
4  1242  3  Djibouti
4  1243  5  Yemen (Sana)
4  1244  3  Madagascar
4  1245  6  Yemen (Aden)
4  1246  5  Iran
4  1247  6  USSR - Franz Josef Land
4  1248  4  Unnamed neutral zone between Iraq and Saudi Arabia
4  1249  1  Caspian Sea
4  1250  6  USSR - Ostrov Kolguyev
4  1251  2  United Arab Emirates
4  1252  4  Qatar
4  1253  6  USSR - Novaya Zemlya
4  1254  4  Oman - northern section
4  1255  6  USSR - Franz Josef Land
4  1256  6  Yemen (Aden) - Socotra (island)
4  1257  6  USSR - Franz Josef Land
4  1258  3  Reunion Island
4  1259  6  USSR - Franz Josef Land
4  1260  4  Oman - southern section
4  1261  3  Mauritius
4  1262  1  USSR - Aral Sea
4  1263  6  USSR - Franz Josef Land
4  1264  3  Afghanistan
4  1265  4  Pakistan
4  1266  6  USSR - Franz Josef Land
4  1267  2  India
4  1268  3  Ile Kerguelen
4  1269  1  USSR - Lake Balkhash (west)
4  1270  5  China
4  1271  1  USSR - Lake Balkhash (east)
4  1272  4  Disputed area between India and China
4  1273  5  Sri Lanka (Ceylon)
4  1274  3  Nepal
4  1275  3  Mongolia
4  1276  4  Bangladesh
4  1277  3  Bhutan
4  1278  6  USSR - Ostrov Oktyabr'skoy Revolutsii, Komsomolets, Pioner
4  1279  3  Burma
4  1280  5  Indonesia - Sumatra
4  1281  5  Indonesia - Nias
```

```
4  1282  5  Thailand
4  1283  5  Indonesia - Siberut
4  1284  6  USSR - Ostrov Bol'shevik
4  1285  6  Laos
4  1286  2  Cambodia
4  1287  4  Viet Nam
4  1288  1  USSR - Lake Baikal
4  1289  5  Indonesia - Bangka
4  1290  5  Indonesia - Java
4  1291  5  Indonesia - Belitung
4  1292  5  China - Hainan Dao (island)
4  1293  5  Indonesia - Borneo - Kalimantan
4  1294  6  Malaysia - Borneo - Malaysia Timor
4  1295  6  USSR - Ostrov Bol'shoy Begichev
4  1296  5  Indonesia - Bawean
4  1297  3  Australia
4  1298  3  Borneo - Brunei
4  1299  5  Indonesia - Bali
4  1300  5  Indonesia - Sumbawa and Lombok
4  1301  4  Philippines - Palawan
4  1302  5  Indonesia - Sulawesi (Celebes)
4  1303  5  Indonesia - Sumba
4  1304  5  Indonesia - Flores
4  1305  4  Phillipines - Luzon
4  1306  2  Taiwan
4  1307  4  Phillipines - Mindoro
4  1308  4  Phillipines - Panay, Negros, and Bohol
4  1309  4  Phillipines - Mindanao
4  1310  5  Indonesia - island
4  1311  5  Indonesia - Timor
4  1312  5  Indonesia - Alor
4  1313  4  Phillipines - Samar
4  1314  2  North Korea
4  1315  5  Indonesia - Wetar
4  1316  5  Indonesia - Buru
4  1317  4  South Korea
4  1318  5  Indonesia - Obi
4  1319  5  Indonesia - Halmahera
4  1320  5  Indonesia - Ceram
4  1321  3  Japan - Kyushu
4  1322  3  Australia - Melville and Bathurst Islands
4  1323  5  Indonesia - Waigeo
4  1324  5  Indonesia - New Guinea - Irian Jaya
4  1325  3  Japan - Honshu
4  1326  5  Indonesia - Kepulauan Tanimbar
4  1327  3  Japan - Shikoku
4  1328  5  Indonesia - Kepulauan Aru
4  1329  5  Indonesia - Biak and Kepulauan Schouten
4  1330  3  Australia - Kangaroo Island
4  1331  6  USSR - Ostrov Kotel'nyy and Ostrov Faddeyevskiy
4  1332  6  USSR - Shantarskiye Ostrova
4  1333  3  Japan - Hokkaido
4  1334  6  USSR - Ostrov Malyy Lyakhovskiy
4  1335  6  USSR - Ostrov Bol'shoy Lyakhovskiy
4  1336  3  New Guinea - Papua
4  1337  6  USSR - Ostrov Sakhalin
4  1338  3  Australia - Tasmania
4  1339  6  USSR - Ostrov Novaya Sibir'
4  1340  6  USSR - Ostrov Iturup
4  1341  3  Bismarck Archipelago - New Britain
4  1342  6  USSR - Ostrov Urup
4  1343  3  Bismarck Archipelago - New Hanover
4  1344  3  Bismarck Archipelago - New Ireland
4  1345  3  Solomon Islands - Bougainville
4  1346  6  USSR - Ostrov Paramushir
```

```
4  1347  3  Solomon Islands - Guadalcanal
4  1348  3  Solomon Islands - San Cristobal
4  1349  6  USSR - Ostrov Karaginskiy
4  1350  3  New Caledonia (island)
4  1351  7  Antarctica - Ross Island
4  1352  6  New Zealand - South Island
4  1353  3  New Hebrides - Espiritu Santo
4  1354  3  New Hebrides - Malekula
4  1355  6  New Zealand - Stewart Island
4  1356  3  Alaska - Attu Island
4  1357  6  New Zealand - North Island
4  1358  6  New Zealand - North Island
4  1359  3  Fiji - Viti Levu
4  1360  3  Fiji - Vanua Levu
4  1361  2  Kuwait
```

8.0 MISCELLANY

8.1 History

About 1963, R. L. Parker of UCSD wrote the original code, named SUPERMAP, using outline data generated by Hershey. This was adapted for use at NCAR by Lee, in 1968. Revisions occurred in January, 1969, and in May, 1971. The code was put in standard NSSL form in October, 1973. Further revisions occurred in July, 1974, in August, 1976, and in July, 1978. In late 1984 and early 1985, the code was heavily revised to achieve compatibility with FORTRAN-77 and GKS, to remove errors, to augment the outline datasets, and to add enough controls to make user modification of the code unnecessary. In 1987, certain changes were made as part of preparing to introduce the solid-color routine named MAPBLA: A routine named MAPEOS was removed and a routine named MAPEOD was implemented in its place; intensity parameters 'I1', 'I2', etc., were removed and the color-index parameters 'C1', 'C2', etc. were implemented instead. Later in 1987, the routines of EZMAPA were made available. In 1992, an inverse-transformation routine was added to the package.

Cicely Ridley, Jay Chalmers, and Dave Kennison (the current curator) have all had a hand in the creation of EZMAP.

8.2 Resolution of Datasets

Data points in the continental outlines are about one degree apart and the coordinates are accurate to .01 degree. Data points in U.S. state outlines are about .05 degrees apart and the coordinates are accurate to .001 degree. Both the spacing and the accuracy of the international boundaries falls somewhere between these two extremes.

8.3 References

Hershey, A.V., “The Plotting of Maps on a CRT Printer.” NWL Report no. 1844, 1963.

Lee, Tso-Hwa, “Students’ Summary Reports, Work-Study Program in Scientific Computing”. NCAR, 1968.

Parker, R.L., “2UCSD SUPERMAP: World Plotting Package”.

Steers, J.A., “An Introduction to the Study of Map Projections”. University of London Press, 1962.