U. S. and Foreign Alloy Cross-Reference Database

Contract NAS8-36166

Final Report

by

Dr. John M. Springer and Dr. Steven H. Morgan

May 31, 1991

Submitted by the Department of Physics

Fisk University

Nashville, Tennessee 37208

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

U. S. and Foreign Alloy Cross-Reference Database

Contract NAS8-36166

Final Report

by

Dr. John M. Springer and Dr. Steven H. Morgan

May 31, 1991

Submitted by the Department of Physics

Fisk University

Nashville, Tennessee 37208

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TABLE OF CONTENTS

APPENDICES

I. INTRODUCTION

A. Overview: Marshall Space Flight Center and other NASA installations have a continuing requirement for materials data from other countries involved with the development of joint international Spacelab experiments and other hardware. This need includes collecting data for common alloys to ascertain composition, physical properties, specifications, and designations. This data is scattered throughout a large number of specification statements, standards, handbooks, and other technical literature which make a manual search both tedious and often limited in extent. In recognition of this problem, a contract was awarded to Fisk University to develop a computerized database of information on alloys along with the software necessary to provide the desired functions to access this data. The intention was to produce an initial database covering aluminum alloys, along with the program to provide a user-interface to the data, and then later extend and refine the database to include other nonferrous and ferrous alloys.

B. Type of Data Gathered: The data elements for each alloy record to be delivered to NASA included the items noted in the list below.

Designation
UNS number if available
Originating organization
Specification number or standard
Composition (minimum and maximum values, when specified)
Form
Alloy type
Condition
Yield strength (Minimum, maximum and typical values)
Tensile strength (Minimum, maximum and typical values)
SCC rating per MSFC specification 522A
Temper
U. S. equivalent alloy (for foreign designations)

In addition, included in the records are indications of where the equivalence, chemical composition, and mechanical property values were found or how they were determined.

C. Software: The original scope of the contract and the collection and analysis of the data itself required a software package to be developed to make it possible to add records, search the database, modify data, and perform the other usual operations expected on a functioning database. It was determined that the Digital Equipment database query language Datatrieve (TM) would be used and imbedded in a more user-friendly environment coded in FORTRAN. As the project developed, the decision was made to combine this database with other materials databases at MSFC using the ORACLE (TM) database language. After this point program development at Fisk concentrated on improving the FORTRAN program to use on-site for database maintenance.

D. Hardware requirements:

The alloy cross-reference database was stored in a Digital Equipment VAX-Datatrieve accessible format at Fisk, and delivered to NASA/MSFC on 1/2" magnetic tape for transfer to the ORACLE system.

II. DATA COLLECTION

A. Sources: Alloy data was mainly collected by evaluating

specifications issued by standards organizations, government publications and, to a lesser extent, manufacturers. A list of these sources is given in Appendix A. The data records themselves give the individual standards and specifications that were referenced in their preparation. It was originally planned that sources of collected data already available from technical associations would be purchased when they meet the needs of this project, but it was found impossible to do this except on a yearly licensing basis, which was not acceptable to MSFC. For reference, some cross-referencing collections available from various publishers are listed in Appendix B.

B. Procedures: Once appropriate metals standards or other data sources were identified and obtained, the limited data needed for the database was extracted and compiled on hand-written datasheets. This information was then added as new records on the database, printed out, and verified by comparing the printout to the original sources. Two internal status fields were used to keep track of whether the data had been verified, was sufficiently complete to put on the NASA/MSFC database, and whether the US/Foreign equivalency had been either established or determined to in all probability not exist. A more detailed description of this process, as given in an extracct from the instruction manual created for our student workers, is presented in Appendix C. At the end of this manual are illustrations of each of the standard forms that were developed during the preparation of the database.

These forms included the basic datasheet, a multiple-entry form for recording records that differed only in a few characteristics, a verification form, a country form for summarizing standard terms used for fields such as form and condition, and a standards assessment form used to note relevant information about individual standards and specifications.

CHAPTER III - ALLOY CROSS-REFERENCING

A. General Problem: A major task at hand in preparing the database was to furnish cross-referencing between equivalent US and foreign alloys. While superficially a simple task, in practice it was not so straighforward due to two major difficulties. These difficulties were that (1) the meaning of the term "equivalent" when applied to alloys is so greatly subject to the final application that alloys will be used for, and (2) that different national standards organizations use differing philosophies in determinging standards criteria. The latter problem meant that in principle one had to compare standards in which the key indicators of alloy identification were different. For instance, a German specification might use mechanical properties as the basis for determining compliance with the standard while composition was given considerable latitude. The comparable US standard might hold to a strict compositional requirement while considering mechanical properties as a derivative specification. Such problems requiring a detailed standards analysis have been addressed in a publication

by the NSTI, in which very few nominally equivalent materials were found to be truly equivalent for the purpose of ship building.

The former problem of dealing with the ramifications of nominally equivalent alloys in specific applications is to some extent even more of an intractable problem than that of differing rationales in designing standards. This is because it would require foreknowledge of the use to which an alloy will be put, which is impossible in an open access database. Thus while two alloys might be nominally equivalent for general purpose use, some specific difference in properties or tolerances might render them incompatible for a specific application.

B. Existing References: Several compilations of US/Foreign alloy designations with cross-referencing have been published, but very little is published on the rationale for considering different alloys to be equivalent. These may be purchased from the publishers, but were of little direct use in this effort due to copyright problems. A summary of them is furnished in Appendix B. In some cases, foreign standards are self-cross-referencing to US or non-US standards. Although even these are not always unambiguous, when such internal cross-referencing was provided in the standard it was generally used. In addition, certain foreign standards have been written whose sole purpose is to provide cross-referencing.

C. Composition Matching: The system used for most cross-

referencing is composition matching. This may fail, of course, where the standard is specific for mechanical properties and allows considerable latitude in composition. For this database, however, it was the primary means of either finding matching alloys or verifying that alloys linked by other sources were reasonable matches. A program was written to search for compositionally equivalent alloys that had been placed on the database to aid in the cross-referencing. This program allowed matching tolerances in accordance with ASTM standards but was not entirely successful as the database grew due to the long searching time that it required. Another technique was simply to sort the records of a given alloy type by composition and manually compare adjacent groups of records. This was useful, but depended greatly on the order of the sort (in terms of the elements) on the classes of alloys that would be grouped together. All of these techniques required individual attention to each alloy at some point to verify the matching that was done.

C. Heat Treatments/Conditions: A subsidiary part of the alloy matching process was to also match heat treatments and other conditioning methods. Since the behavior of a metal and its mechanical properties depend so much on the specific conditions of its manufacture, we attempted to provide an equivalent condition in terms of its US nomenclature for each alloy record on the database. In some cases this was not very difficult, as for instance the temper designating systems for aluminum that are used in many countries. In other cases, one could only make a rough comparison

since the condition specifications allowed considerable latitude in their application. In many cases, there was a one-to-many or many-to-one problem in which the foreign condition codes encompassed many more detailed domestic codes or vice versa.

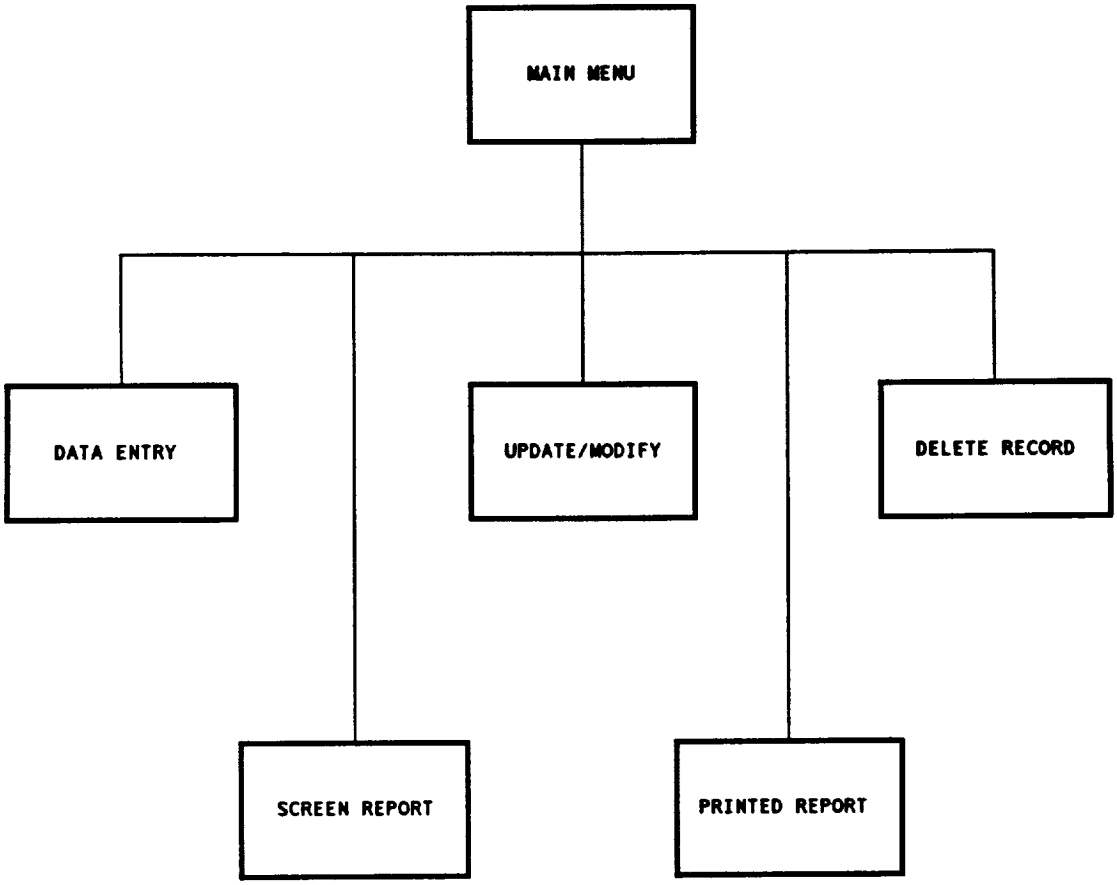
CHAPTER IV - SOFTWARE

The database program to be used with the data was originally specified to be written in Fortran using embedded Datatrieve statements. It was to have the usual database functions, including record searching, entry, deletion, updating, and report generation. With the transition of the materials database at MSFC to an ORACLE (TM) environment, development of the Fisk program was reduced to those activities needed to make it more responsive to the data collection and analysis effort. A simple block diagram of the main parts of the program is on the next page. The data entry and updating routines are the parts that have been most important to data collection, and have been written in a full screen forms-type environment for Digital Equipment VT series terminals. Program listings are given in Appendix E.

CHAPTER V - RESULTS

A. SUMMARY OF DATA: Approximately 10,000 aluminum data records, 10,000 steel records, and 2000 copper records were collected from

DATABASE PROGRAM



standards and specifications. Examples of the data are shown in the table on the next page. The first entry shows data for the US ferrous alloy 405 stainless steel, in the plate form and in the annealed condition. The next entry shows the printed record for the West German alloy X6CrAl13, which is equivalent to the US 405 steel. The third figure shows another alloy, this time from France, which is also nominally equivalent to the 405 designation. As all three records have the same form and condition, their mechanical properties as well as their compositions can be compared to further evaluate their equivalence for a particular application.

COMPARISON OF DATA FOR THREE SIMILAR ALLOYS

Rec#: 30215 Desig: 405
Altdesig:
Cond: Annealed Form: Plate

UNS: S40500 IRR#: NA Country: U.S.A. Type: Fe
Alcat: Stainless Steel
Org: ASTM US-Eqv: 405 Eqvtemp: Annealed

COMPOSITION:

	Al	Si	Fe	Cu	Mn	Mg	Zn	V	Ti	Zr	Cr	Ni	Pb	Sn
MIN:	0.1										11.5			
MAX:	0.3	1.0			1.0						14.5	0.6		

	C	Co	Mo	W	P	S	B	Be	Ga
MIN:									
MAX:	0.08				0.04	0.03			

Specs: ASTM A176

	MIN	MAX	TYP	UNITS
Yield Strength:	25			ksi
Tensile Strength:	60			ksi
Hardness :				

Elong: % Test Piece:
SCC Rating:

Notes:
Eqvref: S1 Ref1: S1 Ref2: S1 Status: VZ RC: S Updated: 1-Jul-1989

Rec#: 47262 Desig: X6CrAl13
Altdesig:
Cond: Annealed Form: Plate

UNS: S40500 IRR#: NA Country: West Germany Type: Fe
Alcat: Stainless Steel
Org: DIN US-Eqv: 405 Eqvtemp: Annealed

COMPOSITION:

	Al	Si	Fe	Cu	Mn	Mg	Zn	V	Ti	Zr	Cr	Ni	Pb	Sn
MIN:											12.0			
MAX:		1.0			1.0						14.0			

	C	Co	Mo	W	P	S	B	Be	Ga
MIN:									
MAX:	0.08				0.045	0.03			

Specs: DIN 17440

	MIN	MAX	TYP	UNITS
Yield Strength:	36			ksi
Tensile Strength:	58	87		ksi
Hardness :				

Elong: % Test Piece:
SCC Rating:

Notes: Mech Prop for <= 0.472" thickness
Eqvref: H34 Ref1: S1 Ref2: S1 Status: V RC: R Updated: 7-Jan-1991

Rec#: 34020 Desig: Z6CA13
Altdesig:
Cond: Annealed Form: Plate

UNS: S40500 IRR#: NA Country: France Type: Fe
Alcat: Stainless Steel
Org: NF US-Eqv: 405 Eqvtemp: Annealed

COMPOSITION:

	Al	Si	Fe	Cu	Mn	Mg	Zn	V	Ti	Zr	Cr	Ni	Pb	Sn
MIN:	0.1										11.5			
MAX:	0.3	1.0			1.0						13.5			

	C	Co	Mo	W	P	S	B	Be	Ga
MIN:									
MAX:	0.08				0.04	0.03			

Specs: NF A35-573 NF A35-602

	MIN	MAX	TYP	UNITS
Yield Strength:	33			ksi
Tensile Strength:	61	90		ksi
Hardness :				

Elong: % Test Piece:
SCC Rating:

Notes: Mech Prop for 0.197-0.394" thickness
Eqvref: S2 Ref1: S1 Ref2: S1 Status: V RC: S Updated: 1-Jul-1989

CHAPTER VI - CONCLUSIONS/LESSONS LEARNED

A. Size of Database: The number of distinct designations for the three alloy types we dealt with is very large, in the several tens of thousands if one counts commercial names. This meant that the problems of data control were very difficult to handle without constant attention. In the university environment, where the academic cycle makes a constant level of management virtually impossible, these data control problems are even harder to deal with.

B. Non-uniformity of Data Sources: We obtained much of our data on composition and properties from international standards written in various formats and languages. Each of these documents had to be at least partially translated and prepared for the data entry personnel, who were students for the most part. This made it necessary to first enter the data onto standard data sheets to assure a uniform format for data entry. This therefore greatly increased the time needed to get raw data on the database over the case where direct entry from the original documents would have been possible. It also created problems in trying to keep to a standard format when differing sources used slight variations in the format of designations. For example, the designation A199.9 in one standard might be written A199,9 , with the period changed to a comma.

C. Changing Standards: Standards are continually being created, revised and canceled. Canceled standards were difficult to obtain, and made a historically complete database extremely difficult, if not

impossible. On the other hand, revised and updated standards presented the problem of aiming at a moving target. As standards evolve, quite basic alloy properties such as composition may change. This meant that back-checking data records against recently revised standards often gave the impression that the original records were incorrect, while in actual fact the standard are simply been adjusted to better reflect modern metallurgical practice.

APPENDIX A
REFERENCE INDEX FOR US/FOREIGN CROSS-REFERENCE DATABASE

This list gives the meangings of the publication codes used in the three reference fields in each data record.

==========

H01
Handbook of Aluminum (Alcan)

H02
Handbook of International Alloys Composition and Designations, Vol 3
Harold J. Hucek, Editor
Metals and Ceramics Information Center: Columbus

H03
Handbook of Soviet Alloy Compositions
Douglas Joslyn, Jr and Marshall J. Wahll
Metals and Ceramics Information Center: Columbus
1980

H04
Metals Handbook, Ninth Edition, Vol 2
William H. Cubberly, Hugh Baker, et. al, editors
American Society For Metals: Metals Park
1979

H05
Worldwide Guide to Equivalent Nonferrous Metals and Alloys
Paul M. Unterweiser, Staff Editor
American Society For Metals: Metals Park
1980

H06
Key to Aluminum Alloys
W. Hufnagel
Aluminium-Zentrals: Dusseldorf
1982

H08
Sourcebook on Industrial Alloy and Engineering Data
American Society For Metals: Metals Park
1978

H09
The Properties of Aluminium and Its Alloys
The Aluminium Federation: Birmingham UK
1983

H10
Aluminium-Taschenbuch
Herausgeber und Bearbeiter

Aluminium-Zentrale: Dusseldorf
1983

H11
JIS Ferrous Materials and Metallurgy
Japanese Standards Association
1986

H12
JIS Non-Ferrous Materials and Metallurgy
Japanese Standards Association
1986

H13
Aluminum Properties and Physical Metallurgy
John E. Hatch
American Society For Metals: Metals Park
1984

H14
Metals and Alloys in the Unified Numbering System, Fourth Edition
Alvin G. Cook, Chairman UNS Advisory Board
Society of Automotive Engineers, Inc.: Warrendale
1986

H15
Handbook of Comparative World Steel Standards, Vol 6
International Tech Information Institute: Tokyo
1985

H16
Military Handbook 694A(MR)
1966

H17
Light Alloys Metallurgy of the Light Metals
I. J. Polmear
American Society For Metals: Metals Park

H18
Alcoa Aluminum Handbook
Aluminum Company of America: Pittsburgh
1967

H19
Metals Handbook, Ninth Edition, Vol 1
Bruce P. Bardes, Editor
American Society For Metals: Metals Park
1978

H20
Metals Handbook, Ninth Edition, Vol 3

David Benjamin, Senior Editor
American Society For Metals: Metals Park
1980

H22
Worldwide Guide to Equivalent Nonferrous Metals and Alloys, Second Edition
Harold M. Cobb, Consulting Editor
ASM International: Metals Park
1987

H23
Engineering Properties of Steel
Philip D. Harvey, Editor
American Society For Metals: Metals Park
1982

H24
Material Properties Handbook, Vol 1
The Royal Aeronautical Society: Hamilton Place
1959

H25
Material Properties Handbook, Vol 2
The Royal Aeronautical Society: Hamilton Place
1960

H26
Military Handbook 694A(MR)
1966

H27
Metals Data
Samuel L. Hoyt, Technical Advisor
Reinhold Publishing Corp: New York
1952

H28
ASM Metals Handbook
1964

H34
Heat Treaters's Guide
Paul M. Unterweiser, Senior Editor
American Society For Metals: Metals Park
1982

H35
Steel Products Manual: Stainless and Heat Resisting Steels
American Iron and Steel Institute: Washington
1974

H39
Metals Databook
Colin Robb
The Institute of Metals: London
1987

H42
Wolman's Engineering Alloys, Sixth Edition
Robert C. Gibbons, Editor
American Society for Metals: Metals Park
1979

H45
Metals and Alloys in the Unified Numbering System, Fifth Edition
Alvin G. Cook, Chairman UNS Advisory Board
Society of Automotive Engineers, Inc.: Warrendale
1989

M06
Aluminum Standards and Data, Seventh Edition
The Aluminum Association: Washington
1982

M07
Aluminum Standards and Data, First Edition
The Aluminum Association: Washington
1978

M08
Aluminum Standards and Data, Second Edition
The Aluminum Association: Washington
1986

M09
Standards for Aluminum Sand and Permanent Mold Castings,
Eleventh Edition
The Aluminum Association: Washington
1986

R01
International Metallic Materials Cross Reference, Second Edition
James V. Arcuri and Daniel L. Potts
Genium Publishing Corporation: Schenectady
1984

R02
Materials Selection List For Space Hardware Systems, Vol 6
Marshall Space Flight Center: Huntsville
1984

R08

Filler Metal Comparison Chart
American Welding Society: Miami
1986

R09
Stahlschlussel: Key to Steel
C. W. Wegst
Verlag Stahlschlussel Wegst GmbH
1986

R10
Registration Record of International Alloy Designations and Chemical Composition
The Aluminum Association: Washington
1987

R11
Registration Record of AA Designations and Chemical Composition Limits
The Aluminum Association: Washington
1987

R15
Nomenclature Internationale Des Alliages D'Aluminum De Fonderie, Fourth Edition
Patrick Bertrand
Centre Technique Des Industries De La Fonderie
1986

R16
Structural Aluminum Design
Karl Angermayer
Reynolds Metals Company
1962

R17
Design Criteria for Controlling Stress Corrosion Cracking
Marshall Space Flight Center: Huntsville
1977

R18
L'Aluminium, Tome 1
M. Pierre Barrand
Editions Eyrolles: Paris
1964

R19
The Aluminum Data Book
G. W. Birdsall, Editor
Reynolds Metals Company: Richmond
1965

R20
Metallurgy of Aluminum Alloys
Marc Van Lancker
John Wiley and Sons:
1967

R22
ASM Metals Reference Book
William Cubberly, Director Reference Publications
American Society For Metals: Metals Park
1983

R23
Multilingual Glossary of Heat Treatment Terminology
Prof. Dr. Ing. Habil. Eugeniusz Tyrkiel, Editor
The Institute of Metals: London
1986

R29
Registration Record of Aluminum Association Alloy Designations and Chemical Composition Limits for Aluminum Alloys in the Form of Castings and Ingot
The Aluminum Association: Washington
1987

S01
NF A35-604
Tool Steels - Comparison of French and Foreign Standard Grades
Association Francaise De Normalisation: Paris
1978

==========

Note: Each entry is preceded by the index symbol by which it is referred to in the database. The index symbols are not consecutive because some references on our publications list are not used in the database.

NATIONAL STANDARDS BY COUNTRY

COUNTRY ALUMINIUM SPECIFICATIONS

AUSTRALIA

AS 1734
AS 1865
AS 1866
AS 1867

CANADA

Alcan
CSA HA.1
CSA HA.2
CSA HA.3
CSA HA.4
CSA HA.5
CSA HA.6
CSA HA.7
CSA HA.7.1
CSA HA.8
CSA HA.9
CSA HA.10

DENMARK

DS 3002
DS 3012

FRANCE

AIR 3350
AIR 9050
NF A50-411
NF A50-451
NF A50-701
NF A50-901
NF A57-350
NF A57-702
NF A57-703
NF A02-002

JAPAN

JIS H2102
JIS H2103
JIS H2111
JIS H2117
JIS H2118
JIS H2206
JIS H2211
JIS H2212
JIS H4000
JIS H4040
JIS H4080
JIS H4090
JIS H4100
JIS H4120
JIS H4140
JIS H4160
JIS H4170
JIS H4180
JIS H5114
JIS H5202
JIS H5302
JIS H5402
JIS Z3232
JIS Z3263

NORWAY

NS 17005
NS 17010
NS 17011
NS 17105
NS 17205
NS 17210
NS 17215
NS 17220
NS 17305
NS 17310
NS 17405
NS 17410
NS 17510
NS 17512
NS 17520
NS 17525
NS 17532
NS 17535
NS 17550
NS 17552
NS 17570

SPAIN

UNE 28264
UNE 38030
UNE 38031
UNE 38032
UNE 38033
UNE 38034
UNE 38125
UNE 38201
UNE 38211
UNE 38213
UNE 38214
UNE 38215
UNE 38231
UNE 38233
UNE 38234
UNE 38235
UNE 38241
UNE 38252
UNE 38253
UNE 38256
UNE 38257
UNE 38261
UNE 38262
UNE 38263
UNE 38265
UNE 38266
UNE 38267
UNE 38268
UNE 38269
UNE 38271
UNE 38291
UNE 38319
UNE 38322
UNE 38332
UNE 38337
UNE 38344
UNE 38354
UNE 38372
UNE 38374
UNE 38383
UNE 38392

SWEDEN

MNC 14E
MNC 40E
MNC 41E
MNC 42E
SIS 144004
SIS 144005
SIS 144007

SIS 144008
SIS 144010
SIS 144015
SIS 144017
SIS 144020
SIS 144021
SIS 144022
SIS 144024
SIS 144054
SIS 144055
SIS 144067
SIS 144102
SIS 144103
SIS 144104
SIS 144106
SIS 144107
SIS 144120
SIS 144133
SIS 144134
SIS 144140
SIS 144146
SIS 144163
SIS 144212
SIS 144225
SIS 144244
SIS 144245
SIS 144250
SIS 144251
SIS 144252
SIS 144255
SIS 144260
SIS 144261
SIS 144262
SIS 144263
SIS 144282
SIS 144283
SIS 144337
SIS 144338
SIS 144355
SIS 144425
SIS 144438

SWITZERLAND

SN 210900
SN 210901
SN 210902/1
SN 210903/1
SN 210906/1
SN 210907/1
SN 210908/1

UNITED KINGDON

BS 1470
BS 1471
BS 1472
BS 1473
BS 1474
BS 1475
BS 1490
BS 2897
BS 2898
BS 2L55
BS 2L77
BS 2L80
BS 2L83
BS 2L84
BS 2L85
BS 2L87
BS 2L89
BS 2L90
BS 2L91
BS 2L92
BS 2L93
BS 2L95
BS 2L96
BS 2L97
BS 2L98
BS 2L99
BS 3L51
BS 3L52
BS 3L54
BS 3L58
BS 3L59
BS 3L60
BS 3L61
BS 3L63
BS 3L78
BS 3L80
BS 3L81
BS 3L86
BS 4300/4
BS 4300/6
BS 4300/7
BS 4300/8
BS 4300/14
BS 4L35
BS 4L36
BS 4L44
BS 4L53
BS 4L54
BS 4L56
BS 5L16
BS 5L17
BS 5L34
BS 6L37
BS L102
BS L103
BS L105
BS L106
BS L108
BS L109
BS L110
BS L111
BS L112
BS L113
BS L114
BS L115
BS L116
BS L117
BS L118
BS L119
BS L154
BS L155
BS L156
BS L157
BS L158
BS L159
BS L160
BS L161
BS L162
BS L163
BS L164
BS L165
BS L166
BS L167
BS L168
DTD 150A
DTD 246C
DTD 297A
DTD 324B
DTD 372B
DTD 5004A
DTD 5008B
DTD 5010A
DTD 5014A
DTD 5018A
DTD 5024
DTD 5030A
DTD 5040A
DTD 5044
DTD 5070B
DTD 5074A
DTD 5080
DTD 5084A

DTD 5094A
DTD 5100A
DTD 5104A
DTD 5110
DTD 5114
DTD 5120B
DTD 5124
DTD 5130A
DTD 716B
DTD 722B
DTD 727B
DTD 731B
DTD 735B
DTD 745A

USA

5049A-52242A(MR)
AA 86 Std and Data
AA-CS-M1-85
AA-CS-M11-85
AA-CS-M3-85
AA-CS-M4-84
AA Registry
AMS 4000
AMS 4001
AMS 4004
AMS 4005
AMS 4006
AMS 4006
AMS 4007
AMS 4009
AMS 4010
AMS 4011
AMS 4014
AMS 4015
AMS 4018
AMS 4020
AMS 4024
AMS 4025
AMS 4026
AMS 4027
AMS 4029
AMS 4033
AMS 4034
AMS 4038
AMS 4039
AMS 4041
AMS 4044
AMS 4046
AMS 4051
AMS 4056
AMS 4057
AMS 4058
AMS 4062
AMS 4063
AMS 4064
AMS 4065
AMS 4066
AMS 4067
AMS 4068
AMS 4069
AMS 4070
AMS 4071
AMS 4079
AMS 4080
AMS 4081
AMS 4083
AMS 4084
AMS 4085
AMS 4087
AMS 4090
AMS 4094
AMS 4095
AMS 4096
AMS 4100
AMS 4101
AMS 4102
AMS 4107
AMS 4108
AMS 4110
AMS 4111
AMS 4112
AMS 4113
AMS 4114
AMS 4115
AMS 4116
AMS 4117
AMS 4121
AMS 4125
AMS 4127
AMS 4130
AMS 4131
AMS 4134
AMS 4135
AMS 4136
AMS 4138
AMS 4139
AMS 4140
AMS 4142
AMS 4145
AMS 4146
AMS 4147
AMS 4148
AMS 4150
AMS 4152
AMS 4153
AMS 4156
AMS 4157
AMS 4158
AMS 4159
AMS 4162
AMS 4167
AMS 4170
AMS 4179
AMS 4182
AMS 4186
AMS 4190
AMS 4191
AMS 4198
AMS 4200
AMS 4207
AMS 4210
AMS 4212
AMS 4215
AMS 4217
AMS 4218
AMS 4220
AMS 4222
AMS 4223
AMS 4230
AMS 4231
AMS 4235
AMS 4236
AMS 4240
AMS 4241
AMS 4242
AMS 4275
AMS 4280
AMS 4281
AMS 4282
AMS 4283
AMS 4284
AMS 4290
AMS 4291
AMS 4310
AMS 4311
AMS 4313
AMS 4320
AMS 4321
AMS 4340
AMS 4341
AMS 4342
AMS 4344
AMS MAM 4208
AMS MAM 4209

AMSE SB209
AMSE SB209
AMSE SB211
AMSE SB221
ANSI/AWS A5.1-88
ANSI/AWS A5.3-80
ANSI/AWS A5.8-81
ANSI/AWS A5.10-80
ASME SB209
ASME SB210
ASME SB211
ASME SB221
ASME SB234
ASME SB241
ASME SB247
ASTM B85
ASTM B85-60
ASTM B108
ASTM B108-59T
ASTM B178-49T
ASTM B179
ASTM B179-60
ASTM B209
ASTM B210
ASTM B210-49T
ASTM B211
ASTM B211-49T
ASTM B221
ASTM B221-49T
ASTM B230
ASTM B233
ASTM B234
ASTM B234-48T
ASTM B236
ASTM B241
ASTM B247
ASTM B26
ASTM B26-60T
ASTM B275
ASTM B275-63
ASTM B313
ASTM B314
ASTM B316
ASTM B317
ASTM B324
ASTM B345
ASTM B373
ASTM B396
ASTM B398
ASTM B404
ASTM B429
ASTM B483
ASTM B491
ASTM B531
ASTM B547
ASTM B609
ASTM B632
MIL-C-211808
MIL-4-81596
MIL-A-12545
MIL-A-12545C(MR)
MIL-A-15153A
MIL-A-21180C
MIL-A-22771D
MIL-A-22774
MIL-A-25995
MIL-A-45225
MIL-A-46027
MIL-A-46083
MIL-A-46104
MIL-A-52242A(MR)
MIL-A-63547 (AR)
MIL-A-81596
MIL-A-8882
MIL-B-20148C
MIL-C-211808
MIL-E-16053
MIL-F-17132
MIL-P-22771
MIL-P-25995
MIL-T-50777
MIL-T-7081
MIL-W-23351
MIL-W-85
QQ-A-1876
QQ-A-200/1
QQ-A-200/2
QQ-A-200/3F
QQ-A-200/4
QQ-A-200/5
QQ-A-200/6
QQ-A-200/7
QQ-A-200/8
QQ-A-200/9
QQ-A-200/10
QQ-A-200/11
QQ-A-200/12
QQ-A-200/13
QQ-A-200/14
QQ-A-200/16
QQ-A-200/19
QQ-A-20D/1C
QQ-A-224/6
QQ-A-225/1
QQ-A-225/2
QQ-A-225/2
QQ-A-225/3
QQ-A-225/4
QQ-A-225/5
QQ-A-225/6
QQ-A-225/7
QQ-A-225/8
QQ-A-225/10
QQ-A-225/14
QQ-A-250/1
QQ-A-250/2
QQ-A-250/10
QQ-A-250/11
QQ-A-250/12
QQ-A-250/13
QQ-A-250/14
QQ-A-250/15
QQ-A-250/16
QQ-A-250/17
QQ-A-250/18
QQ-A-250/20
QQ-A-250/21
QQ-A-250/22
QQ-A-250/28
QQ-A-250/29A
QQ-A-250/3
QQ-A-250/30
QQ-A-250/4
QQ-A-250/5
QQ-A-250/6
QQ-A-250/7
QQ-A-250/8
QQ-A-250/9
QQ-A-367
QQ-A-371D
QQ-A-371F
QQ-A-430
QQ-A-566
QQ-A-591B
QQ-A-591E
QQ-A-591F
QQ-A-596B
QQ-A-596D
QQ-A-596E
QQ-A-601
QQ-A-601B-1
QQ-A-601E
QQ-A-825
QQ-A-900/5F
QQ-B-655
QQ-B-825

QQ-Q-200/17
QQ-R-566
SAE 201
SAE B547
SAE J452
SAE J454
SAE J4540
SAE J459C
SAE J460E
WW-T-700/1
WW-T-700/2
WW-T-700/3
WW-T-700/4
WW-T-700/5
WW-T-700/6
WW-T-70012D

USSR

GOST 2685
GOST 4784

WEST GERMANY

DIN 1712/1
DIN 1712/3
DIN 1714 (Sup. 1)
DIN 1725/1
DIN 1725/2
DIN 1725/3
DIN 1725/5
DIN 1732/1
DIN 1745/1
DIN 1746/1
DIN 1747/1
DIN 1747/3
DIN 1748/1
DIN 1749/1
DIN 1788
DIN 1788/3
DIN 5513
DIN 8512
DIN 8513
DIN 8566/1
DIN SS13
VDS Liste
WHDL Teil 1, Band 2
WL 3.1124/1
WL 3.1124/100
WL 3.1254/1
WL 3.1254/100
WL 3.1254/2
WL 3.1254 Beiblatt 1
WL 3.1324/1
WL 3.1324/100
WL 3.1354/1
WL 3.1354/100
WL 3.1354/2
WL 3.1354/3
WL 3.1354/4
WL 3.1364/1
WL 3.1364/100
WL 3.1754
WL 3.1854
WL 3.1854 beiblatt 1
WL 3.2374/1
WL 3.2374/100
WL 3.2374/2
WL 3.2374/3
WL 3.2374 Beiblatt 1
WL 3.2384/1
WL 3.2384/100
WL 3.2384/3
WL 3.3214/1
WL 3.3214/100
WL 3.3214/2
WL 3.3214/3
WL 3.3214/4
WL 3.3214/5
WL 3.3214 Beiblatt 1
WL 3.3354
WL 3.3354 Beiblatt 1
WL 3.3524/1
WL 3.3524/2
WL 3.3524 Beiblatt 1
WL 3.4144/1
WL 3.4144/100
WL 3.4144 Beiblatt 1
WL 3.4334/1
WL 3.4334/100
WL 3.4364/1
WL 3.4364/100
WL 3.4374/1
WL 3.4374/100
WL 3.4377/1
WL 3.4377/100
WL 3.4384/1
WL 3.4384/100
WL 3.4394/1
WL 3.4394/100
WL 3.4394/2
WL 3.4394/3

ITALY

UNI 3041
UNI 3043
UNI 3044
UNI 3045
UNI 3046
UNI 3048
UNI 3049
UNI 3050
UNI 3051
UNI 3052
UNI 3054
UNI 3055
UNI 3058
UNI 3059
UNI 3567-66
UNI 3568
UNI 3569-66
UNI 3570
UNI 3571
UNI 3572
UNI 3573
UNI 3574
UNI 3575
UNI 3576
UNI 3577
UNI 3579
UNI 3581
UNI 3582
UNI 3583
UNI 3584
UNI 3735
UNI 3736
UNI 4507
UNI 4508
UNI 4509
UNI 4513
UNI 4514
UNI 5074-74
UNI 5076-74
UNI 5077-74
UNI 5079-74
UNI 5080-74
UNI 5452-64
UNI 6170-68
UNI 6250-68
UNI 6251-68
UNI 6252-68
UNI 6253-68
UNI 6263-68
UNI 6359-68
UNI 6360-68
UNI 6362-68

UNI 7257-73
UNI 7369-74/1
UNI 7369-74/2
UNI 7369-74/4
UNI 7369-74/5
UNI 7788
UNI 7789
UNI 7790
UNI 7791
UNI 7963
UNI 8024
UNI 9001/1
UNI 9001/2
UNI 9001/3
UNI 9001/4
UNI 9003/2
UNI 9005/1
UNI 9006/4

BELGIUM

NBN 437.01
NBN P21-101

NETHERLANDS

NEN 6021
NEN 6022
NEN 6026

PORTUGAL

NPI-1230

ISO

ISO 2779
ISO 3335
ISO 3522
ISO R209
ISO R827
ISO R829
ISO TR2136
ISO TR2778

COPPER SPECIFICATIONS

JAPAN

JIS H3100
JIS H3110
JIS H3130

UNITED KINGDOM

BS 2870
BS 2875

USA

AMS 4500
AMS 4501
AMS 4505
AMS 4507
AMS 4510
AMS 4520
AMS 4530
AMS 4532
AMS 4544
ASTM B21
ASTM B36
ASTM B96
ASTM B103
ASTM B121
ASTM B122
ASTM B133
ASTM B135
ASTM B152
ASTM B169
ASTM B187
ASTM B194
ASTM B249
ASTM B291
ASTM B422
ASTM B466
ASTM B534
ASTM B591
ASTM B592
ASTM B747

WEST GERMANY

DIN 17675

STEEL SPECIFICATIONS

AUSTRALIA

AS 1204
AS 1443
AS 1444
AS 1446
AS 1449
AS 1450
AS 1548
AS 1594
AS 1595
AS 1835
AS 1836

CANADA

CAN/CSA-G40.20
CAN/CSA-G40.21

FRANCE

NF A33-101
NF A35-051
NF A35-501
NF A35-502
NF A35-551
NF A35-552
NF A35-553
NF A35-554
NF A35-561
NF A35-562
NF A35-564
NF A35-571
NF A35-573
NF A35-574
NF A35-575
NF A35-576
NF A35-577
NF A35-578
NF A35-580
NF A35-581
NF A35-582
NF A35-584
NF A35-590
NF A35-600
NF A35-601
NF A35-602
NF A35-604
NF A35-610
NF A36-204
NF A36-205
NF A36-206
NF A36-207
NF A36-208
NF A36-209
NF A36-210
NF A36-610
NF A37-401

JAPAN

JIS G3103
JIS G3108
JIS G3115
JIS G3118
JIS G3119
JIS G3120
JIS G3123
JIS G3124
JIS G3126
JIS G3127
JIS G3201
JIS G3203
JIS G3444
JIS G3458
JIS G3459
JIS G3463
JIS G4051
JIS G4052
JIS G4102
JIS G4103
JIS G4104
JIS G4105
JIS G4106
JIS G4107
JIS G4109
JIS G4303
JIS G4304
JIS G4305
JIS G4306
JIS G4307
JIS G4309
JIS G4311
JIS G4312
JIS G4318
JIS G4401
JIS G4403
JIS G4404
JIS G4804
JIS G5101
JIS G5111
JIS G5121
JIS G5122
JIS G5131
JIS G5151
JIS G5152
JIS G5201
JIS G5202

NORWAY

NS 10010
NS 12101
NS 12111
NS 12120
NS 12122
NS 12123
NS 12124
NS 12142
NS 12143
NS 12153
NS 12501
NS 12502
NS 12536
NS 12542
NS 12546
NS 12555
NS 12569
NS 13205
NS 13234
NS 13244
NS 13254
NS 13840
NS 13860
NS 13882
NS 14001
NS 14002
NS 14115
NS 14210
NS 14230
NS 14240
NS 14310
NS 14325
NS 14330
NS 14350
NS 14355
NS 14360
NS 14450
NS 14455
NS 14460

NS 14480
NS 14485

SPAIN

UNE 36011 75
UNE 36011 79
UNE 36012 75
UNE 36012 78
UNE 36013
UNE 36014
UNE 36021
UNE 36032
UNE 36034
UNE 36070
UNE 36071
UNE 36080
UNE 36081
UNE 36082
UNE 36087/1 74
UNE 36087/1 79
UNE 36087/2 76
UNE 36087/4 89

SWEDEN

MNC 720E
MNC 810E
MNC 811E
MNC 830E
MNC 831E
MNC 842E
MNC 845E
MNC 850E
MNC 851E
MNC 852E
MNC 854E
MNC 870E
MNC 880E
MNC 900E
MNC 916E
MNC 920E
SIS 141111
SIS 141160
SIS 141211
SIS 141232
SIS 141265
SIS 141270
SIS 141300
SIS 141305
SIS 141306
SIS 141311
SIS 141312
SIS 141316
SIS 141330
SIS 141331
SIS 141360
SIS 141365
SIS 141370
SIS 141412
SIS 141414
SIS 141426
SIS 141430
SIS 141431
SIS 141432
SIS 141434
SIS 141435
SIS 141450
SIS 141505
SIS 141550
SIS 141565
SIS 141572
SIS 141606
SIS 141650
SIS 141655
SIS 141672
SIS 141674
SIS 141770
SIS 141774
SIS 141870
SIS 141880
SIS 141912
SIS 141914
SIS 141922
SIS 141926
SIS 141957
SIS 142090
SIS 142101
SIS 142102
SIS 142103
SIS 142106
SIS 142107
SIS 142116
SIS 142117
SIS 142120
SIS 142120
SIS 142121
SIS 142122
SIS 142127
SIS 142130
SIS 142131
SIS 142132
SIS 142134
SIS 142135
SIS 142136
SIS 142140
SIS 142142
SIS 142144
SIS 142145
SIS 142146
SIS 142172
SIS 142174
SIS 142183
SIS 142216
SIS 142218
SIS 142223
SIS 142224
SIS 142225
SIS 142230
SIS 142234
SIS 142242
SIS 142244
SIS 142245
SIS 142253
SIS 142254
SIS 142258
SIS 142260
SIS 142302
SIS 142303
SIS 142304
SIS 142310
SIS 142312
SIS 142314
SIS 142317
SIS 142321
SIS 142324
SIS 142331
SIS 142333
SIS 142343
SIS 142346
SIS 142347
SIS 142366
SIS 142377
SIS 142380
SIS 142382
SIS 142383
SIS 142384
SIS 142385
SIS 142387
SIS 142388
SIS 142506
SIS 142511
SIS 142512
SIS 142523
SIS 142541
SIS 142550

SIS 142564
SIS 142570
SIS 142614
SIS 142615
SIS 142624
SIS 142625
SIS 142632
SIS 142634
SIS 142642
SIS 142644
SIS 142652
SIS 142654
SIS 142662
SIS 142664
SIS 142710
SIS 142715
SIS 142722
SIS 142723
SIS 142736
SIS 142912
SIS 142940
SIS 219701
SIS 219703

UNITED KINGDOM

2S111
2S112
2S113
2S114
2S116
2S117
2S119
2S120
2S124
2S129
2S130
2S131
2S135
2S136
2S137
2S140
2S142
2S143
2S144
2S145
2S146
2S147
2S149
2S510
2S511
2S513
2S514
2S515
2S516
2S517
2S92
2S93
2S97
2S98
2T57
2T66
2T68
3S102
3S105
3S132
3S61
3S62
3S70
3S79
3S91
3S95
3T53
3T60
4S106
4S14
4S28
4S99
4T45
5S15
5S21
5S82
5T2
7S1
BS 1449/1
BS 1449/2
BS 1501/1
BS 1501/2
BS 1501/3
BS 1503
BS 1504
BS 1506
BS 1506
BS 1554
BS 2803
BS 2901/1
BS 3100
BS 3111/1
BS 3601
BS 3603
BS 3604
BS 3605
BS 4360
BS 4534
BS 4659
BS 4670
BS 6323
BS 970
BS 970/1
BS 970/1 1955
BS 970/1 1983
BS 970/5
HC1
HC10
HC101
HC102
HC103
HC104
HC105
HC106
HC2
HC3
HC4
HC5
HC6
HC7
HC8
HC9
S125
S126
S127
S128
S133
S134
S138
S139
S141
S148
S150
S151
S152
S153
S154
S155
S156
S157
S158
S159
S201
S202
S203
S204
S205
S524
S525
S527

S528
S529
S530
S531
S532
S533
S534
S535
S536
S537
S538
T64
T67
T69
T72
T73
T74
T75
T76
T77

USA

AASHTO M223
AISI 303
AISI 303Se
AISI 304
AISI 304N
AISI 316N
AISI 329
AISI 330
AISI 4130
AISI 416
AISI 416Se
AISI 420
AISI 420F
AISI 434
AISI 436
AISI 440A
AISI E52100
AISI S17400
AISI S17700
AISI S30430
AISI W4
AMS 5010
AMS 5024
AMS 5029
AMS 5032
AMS 5040
AMS 5042
AMS 5044
AMS 5045
AMS 5047
AMS 5053
AMS 5060
AMS 5069
AMS 5070
AMS 5075
AMS 5077
AMS 5080
AMS 5082
AMS 5085
AMS 5110
AMS 5112
AMS 5115
AMS 5121
AMS 5122
AMS 5331
AMS 5333
AMS 5335
AMS 5336
AMS 5339
AMS 5342
AMS 5343
AMS 5344
AMS 5355
AMS 5376
AMS 5501
AMS 5503
AMS 5504
AMS 5505
AMS 5506
AMS 5507
AMS 5510
AMS 5511
AMS 5512
AMS 5513
AMS 5514
AMS 5515
AMS 5516
AMS 5517
AMS 5518
AMS 5519
AMS 5520
AMS 5521
AMS 5522
AMS 5523
AMS 5524
AMS 5525
AMS 5528
AMS 5529
AMS 5532
AMS 5546
AMS 5547
AMS 5548
AMS 5549
AMS 5554
AMS 5556
AMS 5557
AMS 5558
AMS 5559
AMS 5560
AMS 5561
AMS 5562
AMS 5563
AMS 5564
AMS 5565
AMS 5566
AMS 5567
AMS 5568
AMS 5570
AMS 5571
AMS 5572
AMS 5573
AMS 5574
AMS 5575
AMS 5576
AMS 5577
AMS 5591
AMS 5592
AMS 5595
AMS 5604
AMS 5610
AMS 5611
AMS 5612
AMS 5613
AMS 5615
AMS 5618
AMS 5620
AMS 5621
AMS 5622
AMS 5624
AMS 5626
AMS 5627
AMS 5628
AMS 5629
AMS 5630
AMS 5631
AMS 5632
AMS 5635
AMS 5636
AMS 5637
AMS 5638
AMS 5639
AMS 5640

AMS 5641
AMS 5642
AMS 5643
AMS 5644
AMS 5645
AMS 5646
AMS 5647
AMS 5648
AMS 5649
AMS 5650
AMS 5651
AMS 5652
AMS 5653
AMS 5654
AMS 5655
AMS 5656
AMS 5657
AMS 5659
AMS 5673
AMS 5674
AMS 5678
AMS 5680
AMS 5685
AMS 5686
AMS 5688
AMS 5689
AMS 5690
AMS 5691
AMS 5692
AMS 5693
AMS 5694
AMS 5695
AMS 5696
AMS 5697
AMS 5700
AMS 5713
AMS 5716
AMS 5719
AMS 5731
AMS 5732
AMS 5734
AMS 5735
AMS 5736
AMS 5737
AMS 5738
AMS 5743
AMS 5744
AMS 5745
AMS 5762
AMS 5764
AMS 5768
AMS 5769
AMS 5774
AMS 5776
AMS 5777
AMS 5780
AMS 5790
AMS 5794
AMS 5804
AMS 5805
AMS 5812
AMS 5813
AMS 5821
AMS 5824
AMS 5825
AMS 5861
AMS 6250
AMS 6260
AMS 6263
AMS 6264
AMS 6272
AMS 6274
AMS 6275
AMS 6276
AMS 6277
AMS 6280
AMS 6281
AMS 6290
AMS 6294
AMS 6317
AMS 6322
AMS 6325
AMS 6327
AMS 6350
AMS 6351
AMS 6355
AMS 6356
AMS 6358
AMS 6359
AMS 6360
AMS 6361
AMS 6362
AMS 6365
AMS 6370
AMS 6371
AMS 6372
AMS 6373
AMS 6381
AMS 6382
AMS 6390
AMS 6395
AMS 6412
AMS 6414
AMS 6415
AMS 6417
AMS 6429
AMS 6435
AMS 6437
AMS 6440
AMS 6443
AMS 6444
AMS 6446
AMS 6447
AMS 6448
AMS 6449
AMS 6450
AMS 6455
AMS 6470
AMS 6485
AMS 6487
AMS 6488
AMS 6490
AMS 6491
AMS 6512
AMS 6514
AMS 6520
AMS 6526
AMS 6530
AMS 7240
AMS 7301
AMS 7304
AMS 7477
AMS 7478
AMS 7482
ASM 5731
ASTM A108
ASTM A128
ASTM A135
ASTM A148
ASTM A167
ASTM A176
ASTM A178
ASTM A182
ASTM A192
ASTM A20
ASTM A202
ASTM A203
ASTM A204
ASTM A211
ASTM A213
ASTM A217
ASTM A220
ASTM A225
ASTM A240
ASTM A249
ASTM A250

ASTM A268
ASTM A27
ASTM A276
ASTM A283
ASTM A284
ASTM A285
ASTM A29
ASTM A297
ASTM A299
ASTM A302
ASTM A304
ASTM A312
ASTM A313
ASTM A320
ASTM A322
ASTM A324
ASTM A331
ASTM A333
ASTM A335
ASTM A336
ASTM A351
ASTM A352
ASTM A353
ASTM A355
ASTM A356
ASTM A36
ASTM A366
ASTM A368
ASTM A370
ASTM A387
ASTM A389
ASTM A412
ASTM A414
ASTM A423
ASTM A426
ASTM A436
ASTM A439
ASTM A441
ASTM A442
ASTM A455
ASTM A473
ASTM A479
ASTM A480
ASTM A493
ASTM A500
ASTM A501
ASTM A505
ASTM A510
ASTM A512
ASTM A514
ASTM A515
ASTM A516
ASTM A517
ASTM A519
ASTM A53
ASTM A533
ASTM A534
ASTM A537
ASTM A538
ASTM A553
ASTM A569
ASTM A570
ASTM A571
ASTM A572
ASTM A573
ASTM A580
ASTM A581
ASTM A582
ASTM A6
ASTM A600
ASTM A605
ASTM A611
ASTM A612
ASTM A619
ASTM A620
ASTM A621
ASTM A622
ASTM A633
ASTM A645
ASTM A662
ASTM A666
ASTM A668
ASTM A681
ASTM A682
ASTM A686
ASTM A693
ASTM A709
ASTM A715
ASTM A732
ASTM A734
ASTM A735
ASTM A736
ASTM A737
ASTM A738
ASTM A743
ASTM A747
ASTM A757
ASTM A769
ASTM A781
ASTM A782
ASTM A788
ASTM A832
ASTM A837
ASTM A842
ASTM A844
MIL-S-22141
MIL-S-25043
MIL-S-46850
MIL-S-5059
MIL-S-7393
MIL-S-862
MIL-S-8955
QQ-S-766
SAE J118
SAE J1249
SAE J1397
SAE J217
SAE J403
SAE J404
SAE J405
SAE J4125
SAE J413
SAE J467b
SAE J778
Steel Product Manual

USSR

MCIC-HB-05

WEST GERMANY

DIN 1623
DIN 1624
DIN 1651
DIN 1652
DIN 1654
DIN 1654/1
DIN 1654/3
DIN 1654/4
DIN 1681
DIN 1692
DIN 1693
DIN 1694
DIN 17100
DIN 17102
DIN 17102/10
DIN 17111
DIN 17120/10
DIN 17140
DIN 17155/2
DIN 17200
DIN 17210
DIN 17211
DIN 17212
DIN 17224

DIN 17240
DIN 17243
DIN 17245
DIN 17280
DIN 17350
DIN 17440
DIN 17441
DIN 17445
DIN 17465
DIN 2391/2
DIN 2393/2
DIN 488/1
WL 1.1144
WL 1.1144 Beiblatt 1
WL 1.1174
WL 1.1174 Beiblatt 1
WL 1.1194
WL 1.1194 Beiblatt 1
WL 1.1274/1
WL 1.1274/100
WL 1.1274 Beiblatt 1
WL 1.3504/1
WL 1.3504/2
WL 1.3505/1
WL 1.3505/2
WL 1.3514/1
WL 1.3514/2
WL 1.3544/1
WL 1.3544/100
WL 1.3550/1
WL 1.3550/100
WL 1.3551
WL 1.3551 Beiblatt 1
WL 1.3552/1
WL 1.3554
WL 1.3934
WL 1.3934 Beiblatt 1
WL 1.4014/1
WL 1.4014/100
WL 1.4014/2
WL 1.4014 Beiblatt 1
WL 1.4044
WL 1.4044 Beiblatt 1
WL 1.4314/1
WL 1.4314/100
WL 1.4314/2
WL 1.4314/3
WL 1.4314/4
WL 1.4324/1
WL 1.4324/2
WL 1.4324/3
WL 1.4454
WL 1.4454/100
WL 1.4504/1
WL 1.4504/100
WL 1.4514
WL 1.4534
WL 1.4534 Beiblatt 1
WL 1.4544/1
WL 1.4544/2
WL 1.4544/3
WL 1.4545/1
WL 1.4545/100
WL 1.4546/1
WL 1.4546/3
WL 1.4546 Beiblatt 2
WL 1.4548/1
WL 1.4548/100
WL 1.4548/2
WL 1.4548/3
WL 1.4548 Beiblatt 1
WL 1.4549
WL 1.4549 Beiblatt 1
WL 1.4554
WL 1.4554/100
WL 1.4554 Beiblatt 1
WL 1.4564/1
WL 1.4564/100
WL 1.4574/1
WL 1.4574/100
WL 1.4574/2
WL 1.4574/3
WL 1.4574 Beiblatt 1
WL 1.4581
WL 1.4581 Beiblatt 1
WL 1.4911
WL 1.4911 Beiblatt 1
WL 1.4914
WL 1.4914 Beiblatt 1
WL 1.4924
WL 1.4928
WL 1.4928/100
WL 1.4928 Beiblatt 1
WL 1.4930/1
WL 1.4930/100
WL 1.4930 Beiblatt 1
WL 1.4933/1
WL 1.4933/2
WL 1.4933 Beiblatt 1
WL 1.4934
WL 1.4939/1
WL 1.4939/2
WL 1.4939 Beiblatt 1
WL 1.4943/1
WL 1.4943/100
WL 1.4943/2
WL 1.4943/3
WL 1.4943 Beiblatt 1
WL 1.4944/1
WL 1.4944/2
WL 1.4944/3
WL 1.4944 Beiblatt 1
WL 1.4954
WL 1.4954 Beiblatt 1
WL 1.4956
WL 1.4956
WL 1.4957
WL 1.4957 Beiblatt 1
WL 1.4974/1
WL 1.4974/2
WL 1.44974 Beiblatt 1
WL 1.4984/1
WL 1.4984/2
WL 1.4984 Beiblatt 1
WL 1.5924
WL 1.5924 Beiblatt 1
WL 1.5934
WL 1.5934 Beiblatt 1
WL 1.6351
WL 1.6351 Beiblatt 1
WL 1.6354
WL 1.6354 Beiblatt 1
WL 1.6359/1
WL 1.6359/1 Beiblatt 1
WL 1.6359/2
WL 1.6359/2 Beiblatt 1
WL 1.6604/1
WL 1.6604/2
WL 1.6604 Beiblatt 1
WL 1.6607
WL 1.6607 Beiblatt 1
WL 1.6657/1
WL 1.6657/100
WL 1.6658/1
WL 1.6658/100
WL 1.6722
WL 1.6722/1
WL 1.6722/100
WL 1.6723
WL 1.6723/1
WL 1.6723/100
WL 1.6723 Beiblatt 1
WL 1.6944/1

WL 1.6944/2
WL 1.6944/3
WL 1.6944/4
WL 1.6944 Beiblatt 1
WL 1.6964
WL 1.6964 Beiblatt 1
WL 1.6974
WL 1.6974/1
WL 1.6974/100
WL 1.6974 Beiblatt 1
WL 1.7214/1
WL 1.7214/100
WL 1.7214/2
WL 1.7220
WL 1.7220 Beiblatt 1
WL 1.7224/1
WL 1.7224/100
WL 1.7254/1
WL 1.7254/100
WL 1.7324
WL 1.7324/100
WL 1.7324 Beiblatt 1
WL 1.7334 Beiblatt 1
WL 1.7734/1
WL 1.7734/100
WL 1.7734 Beiblatt 1
WL 1.7736/1
WL 1.7744/1
WL 1.7744/100
WL 1.7744 Beiblatt 1
WL 1.7784/1
WL 1.7784/100
WL 1.7784/2
WL 1.7784/3
WL 1.7784/4
WL 1.7784/5
WL 1.7784 Beiblatt 1
WL 1.8154
WL 1.8154 Beiblatt 1
WL 1.8514
WL 1.8514 Beiblatt 1
WL 1.8544
WL 1.8544 Beiblatt 1
WL 1.8564
WL 1.8564 Beiblatt 1

UNI 5867
UNI 6407
UNI 7070
UNI 7382
UNI 7500
UNI 7845
UNI 8317
UNI EU28

ISO

ISO 630
ISO 2604/2
ISO 2604/3
ISO 3573
ISO 3755
ISO 4950/2
ISO 4950/3
ISO 4951
ISO 4960
ISO 4995
ISO R683/1
ISO R683/2
ISO R683/3
ISO R683/4
ISO R683/5
ISO R683/7
ISO R683/8
ISO R683/9
ISO R683/10
ISO R683/11
ISO R683/13
ISO R683/14

ITALY

UNI 3608
UNI 4838
UNI 5462-64

APPENDIX B
AVAILABLE CROSS-REFERENCE PUBLICATIONS

International Metallic Materials Cross Reference, 3rd ed
D. L. Potts, J. G. Gensure, editors
Genium Publishing Corporation
PO BOx 1436 Schenectady, NY 12301

Worldwide Guide to Equivalent Nonferrous Metals and Alloys
ASM International
Materials Park, Ohio 44073

Worldwide Guide to Equivalent Irons and Steels
ASM International
Materials Park, Ohio 44073

APPENDIX C
EXTRACT FROM MANUAL FOR DATA ENTRY

INTRODUCTION

Data Accuracy and Validity

The accuracy and validity of the database is extremely important. As the file is built, the record will be matched for consistency with the original data sheets and appropriate corrections will be made. However, getting the data entered correctly the first time will save much time in the long run.

1. DATA COLLECTION

Alloy data are to be extracted from the standards currently on hand. The ALLOY DATA SHEETS are used to record all information about an alloy to be stored in the data file. Each sheet should have a pre- assigned record number which shall be keyed when the record containing that data is entered. These sheets may be obtained from Data Control.

The following pages show sample data sheets. Extreme caution should be taken when writing information onto the data sheets as these are the primary building blocks for the data file.

New records may be created using either of two methods.

1. The ALLOY DATA SHEET should be used when a designation is to be entered for the first time. All information for the designation should be written on the data sheet.

2. The MULTIPLE ENTRY ALLOY DATA SHEET may be used when a record which has characteristics similar to those of an existing record is to be written and only a few changes are required. Only the changes should be written on the data sheet. "Record Number" refers to the "new" record number. "Master Record Number" refers to the record from which the duplicates to be made.

Prior to completing these data sheets it is necessary to have an adequate understanding of the format for writing such information as alloy designation, condition, etc. as the formats vary from one country to another. Given this, a FORMAT GUIDE PER COUNTRY form and STANDARDS ASSESSMENT FORM must be completed by Data Control before work is begun on any group of alloys. (See Figure 1.1 and Figure 1.2)

The STANDARDS ASSESSMENT FORM is especially crucial when working on steel alloys. Because of the more complex nature of steels, there is a lesser degree of uniformity in the writing of standards. Each foreign standard must be paired with its potentially equivalent U.S. standard and analyzed on the basis of equivalence in chemical and mechanical properties.

Listed below are general guidelines. Refer to Appendix A (DATA SHEET GLOSSARY) for detailed explanations of these terms. Designations should be written according to the format given in the source Standard (See FORMAT GUIDE PER COUNTRY). Chemical Name designations should be written in upper/ lower case letters with the first letter of each element being capitalized. (E.g., AlMg) Form should be written in upper/lower case letters. To provide consistency throughout the database, a FORM MODIFICATION notebook which contains all forms to be used in the database has been developed. You must refer to "Forms Summary" in the notebook to ascertain that all forms which you write are valid. Condition should be that terminology given in the source Standard. (See Figure 1.1 FORMAT GUIDE PER COUNTRY.)
Typical conditions for Aluminum are:

F - as fabricated.
O - annealed/recrystallized; may be followed by an integer indicating how it was annealed.
H# - strain hardened, where # represents an integer designation of the type of treatment applied to the alloy.
T - Thermally treated to produce stable tempers other than F, O or H, followed by one or more digits indicating specific treatment applied.
W - Solution heat-treated; an unstable temper applicable to alloys which spontaneously age at room temperature after solution heat treatment. May be followed by a value indicating the period of natural aging (eg, W 1/2 hr).

Eqv-Cond indicates the U.S. condition code equivalent to the code for similar treatments by which foreign aluminum alloy producers achieve a desired result. The EQVTEMP program (in the TEMPER directory) is used to determine condition equivalence for aluminum alloys. Standards organizations handle steel conditions differently from aluminum conditions. Condition codes
are not systematically written so as to represent a given set of treatments which a steel alloy undergoes. Instead, more general terms (e.g., "quenched & tempered") are given and supplemental tables of temperature and time ranges required to achieve specific conditions must be used.

EQV-COND for steel should be determined as follows:

a) Data Control shall perform a comparative analysis of similar foreign/U.S. standards and assess the literal meanings discussed therein. As a result of this analysis, conditions which have different phrasing, under proper circumstances, may be regarded as equivalent. E.g., Solution Heat Treated might be equivalent to Annealing if temperature and time ranges, yield strength, tensile strength, etc. are the same.

b) Otherwise, the EQV-COND should be worded the same as the condition. I.e., "Quenched and Tempered" equals "Quenched and Tempered". Data Control has compiled a list of steel condition abbreviations to be used. (See "Steel Conditions" in Appendix A)

ALLOY CATEGORIES

Aluminum Alloys included in the database fall into one of two categories: 1) Wrought Alloy or 2) Casting Alloy. The diagram below is a pictorial representation of these two groups. This illustration is not inclusive of all forms in which alloys are available; however, these are some of the most common. It is necessary to distinguish between these two categories as the designation formats oftentimes differ for wrought and casting alloys.

UNS Number applies to the U.S. designations. In foreign records the UNS-NO is that number for the US-Equivalent listed in the record (UNS=Unified Numbering System).

Country Name indicates the name of the country in which the alloy originated. (ISO-International Standards Organizations is treated with the same regard as country.)

US-Equiv. should be obtained from the specification, Matching Program or indexed reference materials. All US-Equivalents given in foreign records must also exist in the U.S. data file. If no such U.S. alloy exists, consult Data Control.

Equiv. Ref. refers to the source index representing the publication listing the US-Equivalent for a given alloy.

Alloy Type refers to the primary material in an alloy's composition.

Country Code indicates the unique two-digit numerical code assigned to the country by Data Control.

Orig-Org refers to the (abbreviated) standard organization under which the alloy was developed or the company whose Trademark is used.

Status - Records in their final stage of the verification process shall contain the letter "V". When the record is entered for the first time it is given a status code "VECMT" where

C indicates the Chemical Composition has not been verified.
D appears in the Status field when the duplicate feature of SCRNENTR is used to create a new record (VECMDT).
E indicates the US-Equivelent has not been verified.
M indicates the Mechanical Properties have not been verified.
T indicates the equivalent U.S. temper has not been added to the US-Equivalent. An asterisk (*) in the "T" position indicates no equivalent US temper exists for the foreign temper shown in the record.

As the related fields are verified, the corresponding status code is removed the status field. (E.g., when the chemical composition has been verified, the "C" should be removed from the status field.)

Specifications refer to Standards which contain information about the alloy. See FORMAT GUIDE PER COUNTRY for the proper format. (Figure 1.1)

Composition Unit = 1 (% of element contained in alloy by weight).

Composition Values should be written as follows:

Add leading zero for values less than 1.0, eg., 0.25, 0.1
At least one digit should follow the decimal, eg., 1.0, 1.25
Do not add trailing zero when there is at least one digit already following the decimal.
Correct: 1.5
Incorrect: 1.50
The Min-Al value equals REM whenever composition table lists "Remainder" for that field.
The OTHER1, OTHER2 and OTHER3 fields are used to record chemical composition information for miscellaneous elements. "Each Othr" indicates the limits for non-specified impurities on an individual basis. "Total Othr" indicates total limits for non-specified impurities. Sometimes a standard will give limits for impurities not defined in the FADB record structure. The chemical symbol should be written in the OTHER1 field and the min/max values should be written in the min-o1/max-o1 fields. When recording additional chemical composition data record the miscellaneous elements first, then "Each Othr", followed by "Total Othr". If more than the three fields are needed, enter the remaining data in the NOTES field.

Yield/Tensile Strength should always contain units equal to "ksi" in the final verification stage of the record. Record the information on the data sheet as it appears in the standard. That is, if the standard has Yield/Tensile values in MPa, kp or kgf units, key them into the record that way. There are two exceptions:

1. N/mm^2 is the same as MPa. Replace N/mm^2 with MPa on the data sheet.
2. Some standards give Yield/Tensile values in psi units. These numbers are 1000 times the ksi values.

Example: 85,000 psi equals 85 ksi

When writing the data sheets, divide by 1000 (i.e., drop the three zeros) and change the psi units to ksi on the data sheets as in the example above. Convert all "MPa", "kp", "kgf", etc. units using the appropriate conversion procedure stored in Datatrieve.
Whenever multiple thickness ranges are given and the Yield/Tensile values change with thickness, the values corresponding to the smallest thickness range should be used. As shown in the example below the following entry would be made in the record.

Examples:

Thickness (mm)	Min. Yield (MPa)	Min. Tensile (MPa)
1.00 through 12.50	185	230
12.50 through 25.00	215	245
over 25.00	250	300

MIN-YLD field should contain "185 MPa". MIN-TNS field should contain "230 MPa".

NOTES fields should contain: Mech Prop for 1.00-12.50mm thickness

Notice the spacing and punctuation in the NOTES field. All information stored in the database in its final form should be in English units. The CONV-MPA-KSI procedure should be used to convert the values in the Yield and Tensile field to "ksi" units. This procedure is executed in Datatrieve (at the "DTR>" prompt). The NOTES program should be used to convert milimeters to inches in the Notes field (at the "$" prompt).

IRR Designation is the International Registry Record number. These designations are registered with the Aluminum Association (international). The number stored in the IRR field in the database is a compositional equivalent to the alloy designation in that record. This field provides information on similar alloys in the absence of or in addition to the US Equivalent data.

ALCAT indicates Alloy Category. This field is used for Ferrous designations to further distinguish product types. Examples of Alloy Categories include "Stainless Steel", "Carbon Steel", etc.

RC (Ready-Code) indicates the "tape" status of the record. The following codes are currently being used.

R = record is ready to be placed on tape to NASA
S = record has been sent to NASA on previous tape

A blank entry or one which contains a code other than the above indicates the record either has not been completed and reviewed.

3. DATA VERIFICATION

This phase of the project begins when all data have been entered and the full report printed. The Verification Record Form should be used to record all necessary changes.

The Verification Record Form has seven items which should be completed when recording a change.

PG# -- The computer-generated page number on which the record appears in the report.

RECORD NUMBER -- The number which was assigned during the Data Entry phase.

COUNTRY -- The numeric code assigned to the country of origin.

Country Codes

01	Australia	02	Canada	03	China
04	Denmark	05	East Germany	06	Finland
07	France	08	Japan	09	Mexico
10	New Zealand	11	Norway	12	South Africa
13	Spain	14	Sweden	15	Switzerland
16	United Kingdom	17	U.S.A.	18	U.S.S.R.
19	West Germany	20	Italy	21	Belgium
22	Netherlands	23	Portugal	30	ISO

RECORD STATUS CODE -- The two-digit code listed at the bottom of the Verification Record Form corresponding to the data category. (E.g., "01" indicates a change of some kind in the Designation field.)

RECORD STATUS CORRECTION -- The correct data for that field. (See the Record Layout for list of field names.)

RECORD CORRECTED -- Upon making the correction as specified on the Verification Record Form, the person keying the change should write his/her initials and the date the change was keyed.

TOTAL RECORDS -- When all spaces of the Verification Record Form have been used, the total number of records (not the number of changes) should be written in the space in the upper right corner of the form.

SAMPLE VERIFICATION RECORD FORM ENTRIES

PG#	RECORD NUMBER	COUNTRY	VERIFYING DOCUMENT	RECORD STATUS CODE	RECORD STATUS CORRECTION	RECORD CORRECTED
15	1345	17	ASTM B209	10	Min-Si=0.1	XX mm/dd/yy
39	7782	17	AMS 3099	11	Min-Yld=315	XX mm/dd/yy
				01	218.5	XX mm/dd/77

Occasionally, the situation arises when the same change must be made to a several records. In such cases, this problem should be recorded on an ALLOY DATABASE PROJECT PROBLEM REPORT and will be resolved by Data Control.

VERIFICATION RECORD TOTAL RECORDS:_____

PG#	RECORD NUMBER	COUNTRY	VERIFYING DOCUMENT	RECORD STATUS CODE	RECORD STATUS CORRECTION	RECORD CORRECTED

(RECORD STATUS CODES)

00 NO CHANGE	04 COUNTRY NAME	08 STATUS	12 TENSILE STRG.
01 DESIGNATION	05 COUNTRY CODE	09 SPECIFICATION	13 IRR DESIG.
02 FORM	06 US EQUIVALENT	10 COMPOSITION	14 DATA REF.
T2 CONDITION	07 ORIGINATING ORG.	11 YIELD STRG.	15 NOTES
03 UNS NO.	EC EQV CONDITION	RC READY-CODE	D DUPLICATE

VERIFICATION FORM GUIDELINES

"PG#" refers to the computer-generated page number on which the alloy data appear in the report.

"Record Number" refers to the four-digit number assigned to the record in the data file.

"Country" refers to the numeric code assigned to the country from which the alloy originated. (U.S. code is "17".)

"Verifying Document" indicates the publication used to verify information for the alloy.

"Record Status" indicates whether the record is in a correct or an incorrect status. The record status codes are listed at the bottom of each verification record form. The two-digit number corresponding to the field in question should be listed on the verification form under the "code" heading and the correct information should be written under the "correction" heading.

E.g., a record with an incorrect designation code of 1111 which should have read 2222 would appear as follows on the verification form:

RECORD NUMBER	COUNTRY	VERIFYING DOCUMENT	RECORD CODE	STATUS CORRECTION	RECORD CORRECTED
####	co	document	01	2222	xx date

In such cases when there are errors in a field containing multiple items of information such as Composition, Yield Strength, Tensile Strength, etc. the code should be listed and all necessary corrections listed individually on the following lines.

E.g., the above record also contained a minimum value of .40 for zinc which should have been .04, a maximum value of .55 for copper which should have been .50 and a minimum value of .19 for iron which should have been .10. The verification form would then show:

RECORD NUMBER	COUNTRY	VERIFYING DOCUMENT	RECORD CODE	STATUS CORRECTION	RECORD CORRECTED
####	co	document	01	2222	xx date
			10	min-ZI = .04	xx date
				max-CU = .50	xx date
				min-FE = .10	xx date

(NOTE: Aluminum composition percentage should contain a minimum value "REM" [for Remainder] unless otherwise specified in the composition table.)

Verification Form Guidelines

Record Status Code "D" indicates duplicate record. Such records may

be of three types:

Type I — Record is identical to another record.
Action: Delete the duplicate.

Type II — Alloy has entries for multiple Tempers.
Action: Enter one record from each Temper class.
Delete any subsequent records for the alloy.

Type III* Temper class has multiple thicknesses. Action: Note [notes] the "Mechanical Property Limits are based on thicknesses having minimum value 'x'" ('x' = smallest thickness range). Enter the corresponding Yield [11] and Tensile [12] strengths for that thickness. Delete subsequent records for that alloy and temper.

The Specification code (09) should be used to add or correct data in the spec field.

1. Specs should be listed with the most fundamental ones first.
2. Standard naming conventions should be used from one record to the next.

a. Use spaces (which should be notated by " ") instead of a period.

b. Check to see that every spec consists of the issuing standards organization's abbreviation followed by the specification number. (E.g., ASTM B209)

"Record Corrected" should contain the initials of the person making the correction and should have the date on which the correction was made.

*Items in [] should be written under the "Code" heading on the Verification Record Form.

DATA SHEET GLOSSARY

FADB - Fisk Alloy Database Reference Number (or RECNO, record number).

This is an arbitrary sequential number which is used for bookkeeping purposes. Each sheet should have a unique number assigned to it before it is filled out. The database program will not allow duplicated FADB numbers.

UNS - Unified Numbering System

This stands for Unified Numbering System, and is a general numbering system designed to provide one common unique designation for an alloy that may be known by several different designations under various systems of nomenclature.

DESIG - Designation

An alloy designation is the name by which it is identified. Since the same alloy may be described by several standards organizations which have different systems of nomenclature, it may have several identifying designations. It is important that the designations for a given alloy be included under the specifications heading. In general, for U.S. alloys the Aluminum Association of America designation will be used.

FORM - Form

This indicates the shape or type of product into which the alloy is made. The form is usually specified by one of the specifications. One designation may have several forms, requiring separate records.

CONDITION - Condition

The physical properties of an alloy are in part determined by heat treatments and work or strain hardening after the alloy is formed. The history of work hardening and treatments that a particular alloy has undergone in reaching its final useable state is called its condition.

Typical conditions for Aluminum are :

- F - As fabricated
- O - Annealed/recrystallized - May be followed by an integer indicating how it was annealed.
- H# - Strain hardened, where # represents an integer designation of the type of treatment applied to the alloy.
- T - thermally treated to produce stable tempers other than F, O, or H, followed by a digit or digits indicating specific treatments applied.
- W - Solution heat treated - An unstable temper applicable to alloys which spontaneously age at room temperature after solution heat treatment. May be followed by a value indicating the period of natural aging, eg. W 1/2 hr.

Tempering affects certain physical properties, so different conditions of the same alloy will require separate data sheets. Although the same principles apply to steel alloys as for aluminum, the coding for steel conditions is not simply achieved. This is due mostly to the more complex nature of steel alloys. Refer to the table - "Steel Conditions" for a list of condition and condition codes used in the database.

EQVCOND - Equivalent Condition

This field is used to cross reference similar foreign and domestic treatments.

Originating Organization -

This is the company or governmental organization that has introduced the alloy.

Country of Origin - The country in which the originating organization is located.

Both the country and its two digit code should be entered on the datasheet. These codes include:

C Data definition for Country Codes Table

1 = Australia	2 = Canada	3 = China	4 = Denmark
5 = East Germany	6 = Finland	7 = France	8 = Japan
9 = Mexico	10 = New Zealand	11 = Norway	12 = South Africa
13 = Spain	14 = Sweden	15 = Switzerland	16 = United Kingdom
17 = USA	18 = USSR	19 = West Germany	20 = Italy
21 = Belgium	22 = Netherlands	23 = Portugal	30 = ISO

U. S. Equivalent -

For foreign alloys, the designation of the closest United States equivalent. If this is given in the reference, include it. If a U.S. equivalent is unknown, this should be left blank. The designation of the equivalent alloy should be written using the same rules for entering alloy designations so that a search on this field will find a match if it exists.

Specifications -

The specifications for an alloy are the set of procedures and tests that completely define it. The specifications may refer to the title of a specifying document, such as an ASTM number, or an alternate designation by which the alloy is also known.

Composition -

The composition is the proportion of chemical elements that make up the alloy. This will usually be specified as a range of minimum and maximum percentages of the elements the alloy. These percentages may be specified as either a weight percent or an atomic number percent, ie., relative numbers of atoms of each element in the mixture.

Yield Strength - When forces are applied to a bar which tend to stretch it, the bar undergoes deformations or strains. These strains are proportional to the applied forces when they are small, and the bar will return to its original length when the forces are removed. As the tensile forces grow, however, a point will be reached where the bar undergoes a disproportionate increase in length and suffers permanent distortion. The force at which this inelastic deformation occurs is called the metal's yield strength. It will have units of force / area, usually in Mega-Pascals (MPa) or thousands of pounds per square inch (ksi). Yield strengths will in general be specified within a maximum-minimum range or be given as typical values.

Tensile strength - This quantity refers to the tensile or longitudinal stress at which the cohesive forces within the metal decrease suddenly, but before the metal actually fractures. It will have the same units as the yield strength.

SCC Rating - Stress Corrosion Cracking Rating - This is a letter code which indicates the susceptibility of an alloy to surface crack formation in a corrosive environment.

FORMS USED FOR DATABASE PREPARTION

ALLOY DATASHEET:

This form is used as the basic record of data taken from standards and specifcations. Once completed, the datasheet will be used as the source for a single record on the database.

MULTIPLE ENTRY ALLOY DATASHEET:

This form is used when a new record which has similar characteristics as an existing record is to be created. A new record number is assigned as with the Alloy Data Sheet. The master record number (record being copied) must also be written in the appropriate area. However, unlike with the Alloy Data Sheet, only those fields to be changed need be re-keyed. (E.g., it would not be necessary to repeatedly enter composition values for new records created in this manner.)

PROBLEM REPORT:

Occasionally, records contain errors of the same type. Example, the form shown in a record should correspond to those listed on the FORM LIST. Required changes may be done globally using Datatrieve as opposed to making changes one by one using JSMAINDUP. The records in error are recorded on the Problem Report form and recommended actions are determined on a group basis.

STANDARDS ASSESSMENT:

This form is used to record special relavant information about specific standards.

VERIFICATION RECORD FORM

After all data have been entered, a report is generated. It is necessary to verify the accuracy of the stored data. This is achieved through the use of the Verification Record Form. All required changes are recorded here and keyed at a later date.

ALLOY DATA SHEET

FADB-NO._____

Designation _____ UNS No._____

Alt Desig _____ IRR No._____

Form _____

Condition _____

AlType _____ AlCat _____

Country Name _____ Code_____ Orig. Org._____

U.S. Equiv _____ US Eqv. Condition _____

Specifications: 1) _____ 2) _____

3) _____ 4) _____ 5) _____

COMPOSITION Wt% 1

Element	Min	Max	Element	Min	Max
Al			Pb		
Si			Sn		
Fe			C		
Cu			Co		
Mn			Mo		
Mg			W		
Zn			P		
V			S		
Ti			B		
Zr					
Cr					
Ni					

MECH PROPERTY	MIN	MAX	TYPICAL	UNITS
Yield Strength				
Tensile Strength				
Hardness				

% Elongation Long _____ Trans _____ Test piece: _____

Mech Prop Notes:_____

SCC RATING:_____

Data References: Eqvref_____ Cref _____ MPref _____

Notes: _____

OPERATOR:_____

MULTIPLE ENTRY ALLOY DATASHEET

Master Designation: _____

New Record Number: _____ Master Record Number: _____

CHANGES

Designation: _____

Form: _____

Condition: _____

UNS No. : _____ Alt Desig: _____

US Equiv Desig: _____

US Equiv Cond: _____

Orig. Org : _____

Spec1: _____ Spec2: _____

Composition:	Element	Min	Max
	_____	_____	_____
	_____	_____	_____
	_____	_____	_____

Min-Yld: _____ Max-Yld: _____ Typ-Yld: _____

Min-Tns: _____ Max-Tns: _____ Typ-Tns: _____

Min-Hrd: _____ Max-Hrd: _____ H-Units: _____

%Elongation: _____ Test Piece: _____

Mech Prop Notes: _____

Eqvref: _____ CRef: _____ MRef: _____

Notes: _____

OPERATOR:

VERIFICATION RECORD TOTAL RECORDS:_____

PG#	RECORD NUMBER	COUNTRY	VERIFYING DOCUMENT	RECORD STATUS CODE	RECORD STATUS CORRECTION	RECORD CORRECTED

(RECORD STATUS CODES)

00 NO CHANGE	04 COUNTRY NAME	08 STATUS	12 TENSILE STRG.
01 DESIGNATION	05 COUNTRY CODE	09 SPECIFICATION	13 IRR DESIG.
02 FORM	06 US EQUIVALENT	10 COMPOSITION	14 DATA REF.
T2 CONDITION	07 ORIGINATING ORG.	11 YIELD STRG.	15 NOTES
03 UNS NO.	EC EQV CONDITION	RC READY-CODE	D DUPLICATE

STANDARDS ASSESSMENT FORM
Fisk Alloy Database

COUNTRY:_____ ALLOY CATEGORY:_____

STANDARD NUMBER:_____ ISSUE DATE:_____

TITLE:_____

_____

FORMS:_____

COMPARABLE STANDARDS:

U.S. STANDARDS

_____

_____

FOREIGN STANDARDS

_____

_____

_____

_____

_____

REMARKS:

APPENDIX D - ABBREVIATIONS

ORGANIZATIONAL ABBREVIATIONS

AA	The Aluminum Association
AFC	A.F.C., Societe des
AIA	Apex International Alloys Inc.
AISI	American Iron and Steel Institute
ALCAN	Aluminum Co. of Canada, Ltd.
ALPAX	Alais, F
AMS	Aerospace Materials Specifications
AP	Aluminum Pechiney
AS	Standards Association of Australia
ASM	American Society for Metals
ASTM	American Society for Testing Materials
AWS	American Welding Society
BSI	British Standards Institute
CSA	Canadian Standards Association
CTC	Carpenter Technology Corp.
DIN	Deutsches Institut fur Normung
DOD	Department of Defense
DS	Dansk Standardiseringsrad
DTD	Department of Trade and Industry
GE	Gillett & Eaton Inc.
GI	Gould Inc., Engine Parts Division
GOST	Staatliches Komitee fur Standardisierung
HDA	High Duty Alloys Ltd.
HW	Honsel-Werke AG., Leichtmetallwerke

IRR	International Registration Record
ISO	International Standards Organization
JIS	Japanese Industrial Standards
KLAL	Koch Light Alloys Ltd.
MCIC	Metals and Ceramics Information Center
MNC	Metallnormcentralen
MSA	Montecatini Settore Alluminio
NBN	Belgium Standards
NEN	Nederlands Normalisatie-instituut
NF	Association Francaise de Normalisation
NS	Norwegian Standards Association
SAE	Society of Automotive Engineers Inc.
SAL	Swiss Aluminum Ltd.
SIS	Svensk Standard
SML	Sterling Metals Ltd.
SN	Schweizer Norm
STONE	Stone Manganese Marine
UNE	Instituto Espanol De Normalization
UNI	Ente Nazionale Italiano Di Unificazione
USA	United States of America
VASA	Veneto per Azioni, Soc. Alluminio
VDS	Vereinigung Deutscher Schmelzhutten
VL	Vereinigte Leichtmetallwerke
WL	Werkstoff-Leistungsblatt

STEEL CONDITION ABBREVIATIONS

AC = AUSTENITE CONDITIONED

AUSTN = AUSTENITIZED

BF = BRIGHT FINISHED

BRI = BRIGHT

CD = COLD DRAWN

CF = COLD FINISHED

CR = COLD ROLLED

CW = COLD WORKED

DECAR = DECARBURIZED

GRND = GROUND

HDN = HARDENED

HDTMP = HARD TEMPERED

HF = HOT FINISHED

HOMOGEN = HOMOGENIZED

HR = HOT ROLLED

HTR = HEAT TREATED

INTMP = INTERMEDIATE TEMPERED

NA = NOT APPLICABLE

NF = NOT FOUND

NORM = NORMALIZED

NS = NOT SPECIFIED

NWH = NO WORK HARDENING

PHDN = PRECIPITATION HARDENED

PHTR = PRECIPITATION HEAT TREATED

Q = QUENCHED

RVA = REVERSION ANNEALED

SFT = SOFTENED

SHTR = SOLUTION HEAT TREATED

SPH = SPHEROIDIZED

STR = STRESS RELIEVED

STRAINHDN = STRAIN HARDENED

TMP = TEMPERED

APPENDIX E - PROGRAM LISTINGS

```
C*************************************************************************
C                                                                        *
C                    CONTROL PROGRAM (MAIN PROGRAM)                      *
C                                                                        *
C Program Name:         NASAMAIN.FOR                                     *
C Date Written:         August, 1985                                     *
C Designer/Programmer: Joseph K. Amanfu, Fisk University, Nashville      *
C Revised by:  J. M. Springer                                            *
C              9/1/88 - Changed default domain to STEEL                  *
C This Program Displays the Main Menu: After a Processing Option has     *
C been selected by the user, the program Calls the corresponding         *
C Subroutine                                                             *
C                                                                        *
C*************************************************************************
C                                                                        *
C The following subroutines are available:                               *
C                                                                        *
C 1. SUBROUTINE CHOOSE: - Shows available domains and prompts the        *
C                         User to select a domain name to be Readied     *
C                                                                        *
C 2. SUBROUTINE DELETE: - Enables the user to PERMANENTLY remove         *
C                         records from the database                      *
C                                                                        *
C 3. SUBROUTINE INSERT: - Enables the user to insert new records into    *
C                         the database                                   *
C                                                                        *
C 4. SUBROUTINE MODIFY: - Enables the user to modify one or more         *
C                         fields within one or more records              *
C                                                                        *
C 5. SUBROUTINE REPORT: - Enables the user to generate an unlimited      *
C                         number of reports from the data base;          *
C                                                                        *
C                         Such reports may be displayed on the video     *
C                         screen or printed on a local printer           *
C                                                                        *
C NOTES:                                                                 *
C                                                                        *
C    Subroutine features 2, 3, and 4 are restricted operations;          *
C    only users with specific access rights may use them                 *
C                                                                        *
C*************************************************************************

  100 INCLUDE 'DTR$LIBRARY:DAB'

C Declare Variables
      CHARACTER*31 DOMAIN
      CHARACTER*1  CHOICE
      CHARACTER*2  ANSWER
      INTEGER      STATUS

C Initialize the DATATRIEVE Call Interface
      CALL LIB$ERASE_PAGE (1,1)
      type *,' '
      type *,'Please stand by'
      type *,' '
      INIT_OPTS =
      1  +DTR$K_SEMI_COLON_OPT
      2  +DTR$K_UNQUOTED_LIT
```

```
     3  +DTR$K_SYNTAX_PROMPT
   200 CALL DTR$INIT (DAB, 100, MSG_BUFF, AUX_BUFF,
      1               INIT_OPTS)
C PORT #1
C Declare a PORT PT1 for STOREing the number records in any
C collection to be established by user with an Rse,
C e.g. in SBREPORT

      CALL DTR$COMMAND (DAB, 'DECLARE PORT PT1 USING ')
      CALL DTR$COMMAND (DAB, '01 NUM PIC 9(4) COMP.;')
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Choose and Ready the Domain

   300 CALL LIB$ERASE_PAGE (1,1)
      type *,' '
      type *,'Please stand by'
      DOMAIN = 'COPPER'       ! New file with fadb-no primary key
                              !  and all character record
      CALL DTR$COMMAND (DAB, 'READY !CMD SHARED;', DOMAIN)
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
C Clear the Screen and Build the Main Menu
   600 CALL LIB$ERASE_PAGE (1,1)
   700 TYPE 750,DOMAIN
   750 FORMAT (T29,' WELCOME TO THE'
      1       /T18,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
      1       /T23,'Default domain = ',A
      2      //' Main Features:'
      3       /' -------------'
      4      //T15,'I = Insert New Records Into The Database'
      6      //T15,'D = Delete One or More Records From The Database'
      5      //T15,'M = Modify Existing Record[s] '
      7      //T15,'S = Search Database To Display or Print Reports'
      8      //T15,'C = Select or Change Domain'
      9      //T15,'H = Help - I Need Guidance'
      A      //T15,'E = Exit - if Finished Using the System'
      B      //T15,'Please Enter The Letter Corresponding To Your Choice'
      C       /T15,'Then Hit the RETURN Key')

   800 ACCEPT 850,CHOICE
   850 FORMAT (A)

      IF ((CHOICE .EQ. 'I') .OR. (CHOICE .EQ. 'i')) THEN
           CALL JSINMOD(DOMAIN,1)
      ELSE IF ((CHOICE .EQ. 'D') .OR. (CHOICE .EQ. 'd')) THEN
           CALL JSDELETE (DOMAIN)
      ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
           CALL JSINMOD(DOMAIN,2)
      ELSE IF ((CHOICE .EQ. 'S') .OR. (CHOICE .EQ. 's')) THEN
           CALL jsREPORT (DOMAIN)
      ELSE IF ((CHOICE .EQ. 'C') .OR. (CHOICE .EQ. 'c')) THEN
           CALL jsCHOOSE (DOMAIN)
           GO TO 600
      ELSE IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
           CALL LIB$ERASE_PAGE (1,1)
           CALL LIB$SPAWN ('fullhelp')
      ELSE IF ((CHOICE .EQ. 'E') .OR. (CHOICE .EQ. 'e')) THEN
           CALL DTR$FINISH (DAB)
           CALL LIB$ERASE_PAGE (1,1)
           TYPE 1000
  1000     FORMAT (' Goodbye ... exiting to the operating'
      1            ' system')
```

```
            GO TO 9999

       ELSE
            type *,'Wrong selection, Please hit RETURN to try again'
            accept 2000,answer
 2000       format (A)
       END IF
       CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
       CALL DTR$COMMAND (DAB, 'FINISH !CMD;',DOMAIN)
       GO TO 200
 9999  END


C*************************************************************************
C                       SUBROUTINE CHOOSE                                *
C                                                                        *
C Module Name:          SBCHOOSE.FOR                                     *
C Date Written:         August XX, 1985                                  *
C Designer/Programmer: Joseph K. Amanfu, Fisk University, Nashville      *
C Modified by J. Springer - 1987,88                                      *
C The module shows the domains available in the current dictionary       *
C and prompts the user to ready a domain.                                *
C If the domain name is invalid or the domain cannot be readied, the     *
C program reprompts for another domain name.                             *
C                                                                        *
C Revised: 5/17/88 - js (Finish any readied domains)                     *
C          6/22/88 - js (Finish only previous data domain)
C*************************************************************************

       SUBROUTINE JSCHOOSE (DOMAIN)

       INCLUDE 'DTR$LIBRARY:DAB'
       CHARACTER*31 DOMAIN
       CHARACTER*2  ANSWER
       LOGICAL      NO_DOMAIN/.TRUE./
       CALL DTR$COMMAND(DAB,'FINISH !CMD;',DOMAIN)
       CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

  100  DO WHILE (NO_DOMAIN)

  150       TYPE 200
  200       FORMAT (' Do you wish to see Domain Names?'
     1             /' Please respond with Y or N '/)
  300       ACCEPT 400,ANSWER
  400       FORMAT (A)

C Input Error-Trap

            IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 450,answer
  450            format (A)
                 GO TO 150
            END IF
            IF ((ANSWER .EQ. 'Y') .OR. (ANSWER .EQ. 'y')) THEN
  500            CALL DTR$COMMAND (DAB, 'SHOW DOMAINS;')
            END IF

C Select DTR Options
 1020 CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
```

```
C Ask the user for the domain and ready it.

 1030       TYPE 1040
 1040     FORMAT (' Enter the name of the domain you'
     1             ' want to use,'
     2            /' or just enter R to return to'
     3             ' the Main Menu:'
     4           //' Domain Name = ',$)

 1050     ACCEPT 1060,DOMAIN
 1060     FORMAT (A)

            IF ((DOMAIN .EQ. 'R') .OR. (DOMAIN .EQ. 'r')) THEN
               go to 9999
            END IF

            type *,' '
            type *,'Searching for Domain, Please stand by'

 1070       CALL DTR$COMMAND (DAB, 'READY !CMD SHARED WRITE;', DOMAIN)
 1080       CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for an error in readying the domain; reprompt if any errors.

 1090       IF (DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) THEN
               TYPE 1100
 1100          FORMAT (' Error: Try domain name again....')
            ELSE
               NO_DOMAIN = .FALSE.
          END IF
      END DO
        NO_DOMAIN = .TRUE.
        CALL LIB$ERASE_PAGE(1,1)
      CALL DTR$COMMAND(DAB,'SHOW READY;')  !Let's see ready domains
      CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      WRITE (6,1095)
 1095 FORMAT (//,5x,'This is your readied domain, type RETURN to
      1 proceed.')
      READ (5,2090)ANSWER
 2090 FORMAT (A)
      RETURN
 9999   END


C*************************************************************************
C                       SUBROUTINE COUNTRY                               *
C                                                                        *
C Module Name:          SBCNTRY.FOR                                      *
C Date Written:         August XX, 1985                                  *
C Designer/Programmer: Joseph K. Amanfu, Fisk University, Nashville      *
C                                                                        *
C                                                                        *
C*************************************************************************

C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C        1         2         3         4         5         6         7

      SUBROUTINE JSCNTRY (cntry,icntry,Xcntry)

      character*2  cntry
      character*15 Xcntry
      CHARACTER*15 CNTRIES (31)
```

```
      integer      Icntry

C Data definition for Country Codes Table

      DATA CNTRIES/'Australia','Canada','China','Denmark',
     1 'East Germany','Finland','France','Japan','Mexico',
     2 'New Zealand','Norway','South Africa','Spain','Sweden',
     3 'Switzerland','United Kingdom','U.S.A.','U.S.S.R.',
     4 'West Germany','Italy','Belgium','Netherlands',
     5 'Portugal',' ',' ',' ',' ',' ',' ',' ','ISO','Cntry Not Known'/

C Prepare subscript for country literals table

100   IF       ((CNTRY .EQ. '01') .OR. (CNTRY .EQ. '1')) THEN
                ICNTRY    =  1
      ELSE IF ((CNTRY .EQ. '02') .OR. (CNTRY .EQ. '2')) THEN
                ICNTRY    =  2
      ELSE IF ((CNTRY .EQ. '03') .OR. (CNTRY .EQ. '3')) THEN
                ICNTRY    =  3
      ELSE IF ((CNTRY .EQ. '04') .OR. (CNTRY .EQ. '4')) THEN
                ICNTRY    =  4
      ELSE IF ((CNTRY .EQ. '05') .OR. (CNTRY .EQ. '5')) THEN
                ICNTRY    =  5
      ELSE IF ((CNTRY .EQ. '06') .OR. (CNTRY .EQ. '6')) THEN
                ICNTRY    =  6
      ELSE IF ((CNTRY .EQ. '07') .OR. (CNTRY .EQ. '7')) THEN
                ICNTRY    =  7
      ELSE IF ((CNTRY .EQ. '08') .OR. (CNTRY .EQ. '8')) THEN
                ICNTRY    =  8
      ELSE IF ((CNTRY .EQ. '09') .OR. (CNTRY .EQ. '9')) THEN
                ICNTRY    =  9
      ELSE IF  (CNTRY .EQ. '10') THEN
                ICNTRY    =  10
      ELSE IF  (CNTRY .EQ. '11') THEN
                ICNTRY    =  11
      ELSE IF  (CNTRY .EQ. '12') THEN
                ICNTRY    =  12
      ELSE IF  (CNTRY .EQ. '13') THEN
                ICNTRY    =  13
      ELSE IF  (CNTRY .EQ. '14') THEN
                ICNTRY    =  14
      ELSE IF  (CNTRY .EQ. '15') THEN
                ICNTRY    =  15
      ELSE IF  (CNTRY .EQ. '16') THEN
                ICNTRY    =  16
      ELSE IF  (CNTRY .EQ. '17') THEN
                ICNTRY    =  17
      ELSE IF  (CNTRY .EQ. '18') THEN
                ICNTRY    =  18
      ELSE IF  (CNTRY .EQ. '19') THEN
                ICNTRY    =  19
      ELSE IF  (CNTRY .EQ. '20') THEN
                ICNTRY    =  20
      ELSE IF  (CNTRY .EQ. '21') THEN
                ICNTRY    =  21
      ELSE IF  (CNTRY .EQ. '22') THEN
                ICNTRY    =  22
      ELSE IF  (CNTRY .EQ. '23') THEN
                ICNTRY    =  23
      ELSE IF  (CNTRY .EQ. '24') THEN
                ICNTRY    =  24
      ELSE IF  (CNTRY .EQ. '25') THEN
```

```
              ICNTRY    =  25
       ELSE IF  (CNTRY .EQ. '26') THEN
              ICNTRY    =  26
       ELSE IF  (CNTRY .EQ. '27') THEN
              ICNTRY    =  27
       ELSE IF  (CNTRY .EQ. '28') THEN
              ICNTRY    =  28
       ELSE IF  (CNTRY .EQ. '29') THEN
              ICNTRY    =  29
       ELSE IF  (CNTRY .EQ. '30') THEN
              ICNTRY    =  30
       ELSE
              ICNTRY    = 31
       END IF
       Xcntry = cntries (icntry)
       RETURN
       END


C*************************************************************************
C                                                                        *
C                       SUBROUTINE JSINMODUP                             *
C                                                                        *
C Program Name:         JSINMODUP.FOR                                    *
C Date Written:         September, 1986                                  *
C Designer/Programmeer, J. Springer, Fisk University, Nashville, TN      *
C Created:    11/26/86                                                   *
C Revised:    3/27/87                                                    *
C             6/21/88 - Added preset status code                         *
C             8/2/88 - Put READY WRITE command at beginning,            *
C                      SET CONTROL=Y at end of routine                   *
C                                                *
C             8/9/89  - Modified for 833 byte record structure           *
C This subroutine accesses the full screen display subroutine SCRNTR     *
C to all either insertion or modification of records in the database.    *
C and also for changing record numbers and duplicating rest of record.  *
C*************************************************************************

       SUBROUTINE JSINMOD (DOMAIN,IM)  ! IM = 1 for insert, 2 for modify
                                       ! 3 for duplicate with new record no.
C Include the DATATRIEVE Access Block

       INCLUDE 'DTR$LIBRARY:DAB'
       INCLUDE 'IODRVCOM'
       INCLUDE 'DATABUFF'
       INCLUDE 'CONTROLY'
C Declarations, etc.

       EXTERNAL       DTR$_SHOWTEXT
       EXTERNAL       DTR$_ERROR
       EXTERNAL       SS$_NORMAL

       CHARACTER*836 DATAREC     !Holds data record
       CHARACTER*31  DOMAIN
       CHARACTER*1   XC          !DEBUG TEST CHARACTER
       INTEGER*2     I
       INTEGER*4     DTR_OPTIONS,RECNO,RECNODUP
       INTEGER*4     NUM_RECS
       INTEGER*4     PGLEN, PGWIDE
       INTEGER       RET_STATUS
       CHARACTER*9   OPS
       CHARACTER*8   FILE
```

```
      CHARACTER*7   INDEX,INDEXDUP
      EQUIVALENCE (FULLREC(1:1),DATAREC(1:1))

C     Initialize arrays, etc.
      DATA File /'DATAFILE'/
      CALL INIT_ARRAYS (file)
      CALL INIT_IODRIVER
      MODIFY = 'Modify'
      INSERT = 'Insert'
      NORMAL = 'Normal'
      FAILED = 'Failed'
      COMPLETE = 'Complt'

C Select DTR$DTR Options:

      DTR_OPTIONS =
     1    DTR$M_OPT_CMD          ! Return on DTR$K_STL_CMD
     2  + DTR$M_OPT_CONTROL_C    ! Enable Control C Handling
     3  + DTR$K_UNQUOTED_LIT     ! Assumes a string is a literal

  400 CALL LIB$ERASE_PAGE (1,1)

      CALL DTR$COMMAND (DAB,'READY !CMD SHARED WRITE;',DOMAIN)
      CALL DTR$DTR(DAB,DTR$M_OPT_CMD)
C   Include file to declare port2

      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
      NUM_RECS = 0      !Initialize to zero
      IF (IM .EQ. 1) THEN
        OPS = 'Insert'
      ELSE IF (IM .EQ. 2) THEN
        OPS = 'Modify'
      ELSE
        OPS = 'Duplicate'
      END IF
200   WRITE (6,201),OPS
201   FORMAT (1X,'Enter record number to ',(A),' [use negative
      1 value to exit]')
      READ (5,202) INDEX
202   FORMAT (A)
      READ (INDEX,'(I)') RECNO
      IF (RECNO .LE. 0)  GOTO 9999
      type *,' '
      type *,'Searching for record, please stand by'
C   Clean up data buffers
      DO 10500 I = 1,836
10500 DATAREC(I:I) = ' '

10310 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH FADB_NO = !CMD;',
     1 DOMAIN,INDEX)

      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          GO TO 90100
      END IF

C Investigate the number of records found,
C if no records were found then return to try another Rse
```

```
10320 CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')

      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF
      IF ((NUM_RECS .EQ. 0 ) .AND. (IM .EQ. 1)) IDIR = 1
      IF ((NUM_RECS .NE. 0 ) .AND. (IM .EQ. 1)) IDIR = 2
      IF ((NUM_RECS .EQ. 0 ) .AND. (IM .EQ. 2)) IDIR = 3
      IF ((NUM_RECS .NE. 0 ) .AND. (IM .EQ. 2)) IDIR = 4
      IF ((NUM_RECS .EQ. 0 ) .AND. (IM .EQ. 3)) IDIR = 5
      IF ((NUM_RECS .NE. 0 ) .AND. (IM .EQ. 3)) IDIR = 6
       GOTO (310, 320, 330, 340, 330, 340), IDIR

310         DATAREC(31:37) = INDEX  !Enter index in data buffer
        DATAREC(114:123) = 'VECM     T'   !Initial Status code
      DATAREC(826:836) = 'TODAY'
      GOTO 1000          !Now goto screen entry routine
320   WRITE (6,321)
321   FORMAT (1X,'This record already exists.  Duplicate record
     2            numbers are not allowed.  Reenter if there was
     3            a typing error.')
      GOTO 200

330   WRITE (6,331)
331   FORMAT (1X, 'Record with this index number does not exist')
      GOTO 200

340   CONTINUE           !Now pick up record to modify
      NUM_RECS = 0       !  Reinitialize

10400 CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'DTR ERROR'
          type *,'Just hit RTN to continue'
          accept 10405,answer
10405     format (A)
          GOTO 200
      END IF

11200 CALL DTR$GET_PORT (DAB, %REF(DATAREC))

C   At this point we have a record to modify or duplicate
      IF (IM .EQ. 3) THEN
403     WRITE(6,490)
490     FORMAT(1X,'Enter new record number for duplicate.'/,
      1           '(Negative value returns to menu)',/)
        READ (5,491)INDEXDUP
491       FORMAT (A)
          READ (INDEXDUP,'(I)') RECNODUP
          IF (RECNODUP .LE. 0)  GOTO 9999
          type *,' '
          type *,'Searching for record, please stand by'

10312 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH FADB_NO = !CMD;',
     1 DOMAIN,INDEXDUP)
```

```
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           GO TO 90100
      END IF

C Investigate the number of records found,
C if any records were found, then cannot use given number
      call dtr$dtr(dab,dtr$m_opt_cmd)
10322 CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')

      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
C          type *,'Numrecs',num_recs
      END IF

      IF (NUM_RECS .NE. 0 ) THEN
        WRITE(6,493)
493     FORMAT (1X,'The record number you want to use exists; ',
      1       /' duplicate record numbers are not allowed.')
        GOTO 403
      END IF
        DATAREC (31:37) = INDEXDUP        !Insert new record number
        DATAREC (114:123) = 'VECMD    T'  !Status code
        DATAREC (825:825) = ' '           !Blank ready code
      END IF
        DATAREC (826:836) = 'TODAY'
1000    CONTINUE
580   CALL  Q_AST_CTRLY
      CALL  SCRNENTR
      CALL LIB$ERASE_PAGE (1,1)
585   IF (.NOT. CTRLY) THEN
        CALL DQ_AST_CTRLY
          IF (Succ .EQ. NORMAL) THEN
          Succ = Flag
C   If modify, delete old record and save new version
C   If insert, just save new record
      IF ((IM .EQ. 1) .OR. (IM .EQ. 3)) GOTO 587
        CALL DTR$COMMAND(DAB,'SHOW ALL')
        CALL DTR$DTR(DAB,DTR$M_OPT_CMD)
        CALL DTR$COMMAND(DAB,'FIND !CMD WITH RECNO = !CMD',
      1  DOMAIN,INDEX)
        CALL DTR$DTR(DAB,DTR$M_OPT_CMD)
        CALL DTR$COMMAND(DAB,'ERASE ALL')
        CALL DTR$DTR(DAB,DTR$M_OPT_CMD)

587     CALL DTR$COMMAND(DAB,'FOR PORT2 STORE !CMD USING
      1 NASAFILE_REC = TEMPREC',DOMAIN)
          CALL DTR$PUT_PORT (DAB,%REF(DATAREC))
          IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
              CALL DTR$DTR(DAB,DTR$M_OPT_CMD)
          ELSE
              CALL DTR$PORT_EOF(DAB)
              CALL DTR$DTR(DAB,DTR$M_OPT_CMD)
          END IF
      GOTO 200
          ELSE
          Succ = 'No Update'
```

```
        END IF
      ELSE
590               CALL HEAD_SET
                  WRITE(6,1001)
                  READ(5,1002)I
                  IF ( I.EQ.1 ) THEN    ! Yes--Save Data
                  CTRLY=.FALSE.
                  GOTO 585
                  ELSE IF (I.EQ.2) THEN  !  Drop This Record
                  GOTO 200
                  ELSE IF (I.EQ.3 ) THEN  !  Reedit Same Record
                  GOTO 580
                  ELSE
                  GOTO 590     ! Invalid Entry
                  END IF
                END IF

1001  FORMAT(1H ,///,T30,'Control/Y Detected.',/,
      2            T20,'Do You Wish To  (1) Save The Data As Is,',/,
      2            T36,'(2)  Discard The Data,',/,
      2            T32,'Or  (3)  Return To The Entry Screen.')
1002  FORMAT(I1)

C Below is the general error message handling routine
C Call the Terminal Server to handle messages at the end of the report

90000 CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C If there was any arror then prompt user to retry again

      IF ((DAB$L_CONDITION .EQ. %LOC(DTR$_SUCCESS)) .AND.
     1    (DAB$L_CONDITION .NE. %LOC(DTR$_ERROR)))  THEN
           GOTO 200
      END IF

90100 TYPE 90105
90105 FORMAT (' There was a Datatrieve error,'
     1        ' Do you wish to try again?'
     2      //' Please respond with Y or N'/)
      accept 90205,answer
90205 format (A)

C Input Error-Trap

            IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 90305,answer
90305            format (A)
                 GO TO 200
            END IF
9999  ISTAT = LIB$SPAWN('SET CONTROL=Y')
      RETURN
      END          !JSINMOD


C*************************************************************************
C                                                                        *
C                      SUBROUTINE DELETE                                 *
C                                                                        *
C Module Name:         SBDELETE.FOR                                      *
```

```
C Date Written:        August XX, 1985                                  *
C Designer/Programmer: Joseph K. Amanfu                                 *
C                                                                       *
C This subroutine enables only NASA users with special access           *
C privileges to delete records from the database                        *
C                                                                       *
C***********************************************************************
      SUBROUTINE JSDELETE (DOMAIN)

C Include the DATATRIEVE Access Block
  100 INCLUDE 'DTR$LIBRARY:DAB'
      INTEGER*4    DTR_OPTIONS
      INTEGER*4    NUM_RECS
      INTEGER      NUMBER
      CHARACTER*31 DOMAIN
      CHARACTER*3  PASSWD
      CHARACTER*30 DSGKEY
      CHARACTER*7  FADB
      CHARACTER*2  ANSWER
      CHARACTER*80 EXPRLINE

C Select DTR$DTR Options:

      DTR_OPTIONS =
     1    DTR$M_OPT_CMD         ! Return on DTR$K_STL_CMD
     2  + DTR$M_OPT_CONTROL_C   ! Enable Control C Handling
     3  + DTR$K_UNQUOTED_LIT    ! Assumes a string is a literal
  200 CALL LIB$ERASE_PAGE (1,1)
  300 TYPE 400
  400 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Deleting Records:'
     2        /' ----------------'
     3       //' Please Enter your Password'/)


C The next library routine will supress the display of the
C Password input on the screen

      CALL LIB$SPAWN ('SET TERM/NOECHO')

  500 ACCEPT 600,passwd
  600 FORMAT (A3)

C Restore the Echo
      CALL LIB$SPAWN ('SET TERM/ECHO')

  700 IF ((passwd .EQ. 'del') .OR. (passwd .EQ. 'DEL')) THEN
          GO TO 800
      ELSE
          type *,' '
          TYPE *,'Sorry, Access Privilege Violation'
          TYPE *,'Hit RETURN to continue'
          ACCEPT 750,ANSWER
  750     format (a)
          RETURN
      END IF

  800 CALL DTR$COMMAND(DAB,'READY !CMD SHARED WRITE;',DOMAIN)
        CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Prompt user to select the record to be deleted
```

```
CCC  800 CALL LIB$ERASE_PAGE (1,1)

      TYPE 1000
 1000 FORMAT (//' Enter the FADB_NO of the record you'
     1          ' wish to delete,'
     3         /' Then hit the RETURN key'/
     4         /' [Enter M to return to main menu]')
      ACCEPT 1020,FADB
 1020 FORMAT (A7)
      IF (FADB .EQ. 'M') GOTO 2300

            type *,' '
            type *,'Searching for record, Please stand by'

C Pass this number to datatrive via DTR$COMMAND

 1050 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH FADB-NO = !CMD;',
     1   DOMAIN,FADB)

C Check for Datatrieve errors
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           type *,'There was a datatrieve error'
           GO TO 2500
      END IF

C Make sure the record was found
C Investigate the number of records found,
C if no records were found then return to try another Rse

 1100 CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')

      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF

      IF (NUM_RECS .EQ. 0) THEN
          type *,' '
          type *,'That record was not found'
          type *,'Please hit RETURN to try again'
          accept 1115,answer
 1115     format (A)
          RETURN
      END IF

C We will come here only if record was found
      CALL DTR$DTR (DAB, 'SELECT;')

C List some fields for user confirmation

      CALL DTR$COMMAND (DAB, 'FOR CURRENT PRINT FADB_NO,
     1 DESIG, US_EQV;')
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           type *,'There was a datatrieve error'
           GO TO 2500
      END IF

C Ask for user confirmation before erasing !!! DANGER !!!
```

```
 2130 TYPE 2135
 2135 FORMAT (/' Are you sure you want to delete the above record(s)?'
     1        /' Please respond with Y or N'/)

      ACCEPT 2140,ANSWER
 2140 FORMAT (A)

C Input Error-Trap
            IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 2145,answer
 2145            format (A)
                 GO TO 2130
            END IF

      IF ((ANSWER .EQ. 'N') .OR. (ANSWER .EQ. 'n')) THEN
           GO TO 2300
      END IF

 2290 CALL DTR$COMMAND (DAB, 'ERASE ALL;')
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      type *,' '
      TYPE *,' The Record has been deleted'
      GOTO 800
 2300 CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
 2400 CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
 2500 TYPE 2600
 2600 FORMAT (/' Hit the RETURN key to continue')
      accept 2700,ANSWER
 2700 format (A)
      CALL DTR$COMMAND (DAB,'READY !CMD SHARED;',DOMAIN)
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
 8888 RETURN
 9999 END


C*************************************************************************C
C                                                                         *
C                       SUBROUTINE REPORT                                 *
C                                                                         *
C Program Name:         SUBREPORT.FOR                                     *
C Date Written:         August XX, 1985                                   *
C Designer/Programmer: Joseph K. Amanfu, Fisk University, Nashville       *
C                                                                         *
C                                                                         *
C This subroutine enables the user to perform various searches on the     *
C database and generate an unlimited number and types of reports.         *
C The reports may be displayed on the video screen or output on a         *
C printer.                                                                *
C                                                                         *
C Revised:  J. Springer, 1986-87                                          *
C Last revision:  10/1/87 to use 762 character record (NASAFILE_REC4)     *
C Please see the On-Line Help notes for available searches                *
C Revised:  J. Springer, July 1989
C           Changed to use NASAFILE_REC5, with 830 characters
C
C*************************************************************************

      SUBROUTINE JSREPORT (DOMAIN)

C Include the DATATRIEVE Access Block
```

```
      INCLUDE 'DTR$LIBRARY:DAB'
C Declarations, etc.
      EXTERNAL      DTR$_SHOWTEXT
      EXTERNAL      DTR$_ERROR
      EXTERNAL      SS$_NORMAL
      LOGICAL*1     FINISH /.FALSE./
      LOGICAL*1     OPENPORT /.FALSE./  !Tells if port is open
      CHARACTER*80  REPHEADER
      CHARACTER*255 dtr_command
      CHARACTER*80  EXPRLINE
      CHARACTER*80  SHOWFLDS(62)
      CHARACTER*80  PRTFLDS
      CHARACTER*30  DSGKEY
      CHARACTER*30  DSGKEY1
      CHARACTER*30  DSGKEY2
      CHARACTER*2   ANSWER
      CHARACTER*2   CHOICE
      CHARACTER*4   UNITY
      CHARACTER*20  FILENAME
      CHARACTER*7   RECKEY
      CHARACTER*7   Xfadb
      CHARACTER*30  Xdesig
      CHARACTER*25  Xequiv
      CHARACTER*27  Xequiv2
      CHARACTER*2   Xcountry
      CHARACTER*4   XcountryX,CNTRYX
      CHARACTER*15  Xcntry
      CHARACTER*1   dummy
      character*9   reckeyx
      character*9   minalx
      character*9   maxalx
      character*9   minsix
      character*9   maxsix
      character*9   minfex
      character*9   maxfex
      character*9   mincux
      character*9   maxcux
      character*9   minmnx
      character*9   maxmnx
      character*9   minmgx
      character*9   maxmgx
      character*9   minznx
      character*9   maxznx
      character*9   MINVx
      character*9   MAXVx
      character*9   mintix
      character*9   maxtix
      character*9   minzrx
      character*9   maxzrx
      character*9   mincrx
      character*9   maxcrx
      character*9   minnix
      character*9   maxnix
      character*9   minpbx
      character*9   maxpbx
      character*9   minsnx
      character*9   maxsnx
      CHARACTER*2   CNTRY
      CHARACTER*1   CCNTRL
      INTEGER*2     ICNTRY
      INTEGER*4     DTR_OPTIONS
      INTEGER*4     NUM_RECS
```

```
      INTEGER*4     PGLEN, PGWIDE
      INTEGER       RET_STATUS
      INTEGER       RECPRT
      INTEGER       IPAGE
      INTEGER       PRT,COUNT


C FULLREC is the space defined to receive the record from
C the Datatrieve buffer

      INCLUDE '[NASA3.JSEXREC]FULLREC5.INC'
C test common
      COMMON/DATAREC/FULLREC  !Holds full datarecord
      COMMON/COUNTRY/XCNTRY
      COMMON/KOUNTS/IPAGE,RECPRT,CCNTRL,COUNT

C Select DTR$DTR Options:

      DTR_OPTIONS =
     1    DTR$M_OPT_CMD          ! Return on DTR$K_STL_CMD
     2  + DTR$M_OPT_CONTROL_C    ! Enable Control C Handling
     3  + DTR$K_UNQUOTED_LIT     ! Assumes a string is a literal

C Select report options

  100 CALL LIB$ERASE_PAGE (1,1)

  200 TYPE 250
  250 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Report Generation Features:'
     2        /' -------------------------'
     3    //T8,' 1 = Produce Standard Reports'
     4    //T8,' 2 = Build your own reports'
     5    //T8,' H = Display help information'
     6    //T8,' M = Return to the Main Menu'
     7   ///T8,' Select option 1 or 2, then hit the RETURN key'/)

 300  ACCEPT 350,CHOICE
 350  FORMAT (A)
C Input Error-Trap
            IF ((((CHOICE .NE. '1') .AND. (CHOICE .NE. '2'))  .AND.
     1          (((CHOICE .NE. 'H') .AND. (CHOICE .NE. 'h')))  .AND.
     1           ((CHOICE .NE. 'M') .AND. (CHOICE .NE. 'm'))))  THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 360,answer
  360            format (A)
                 GO TO 100
            END IF

      IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
           CALL LIB$ERASE_PAGE (1,1)
           CALL LIB$SPAWN ('rephelp')
           GO TO 100
      ELSE IF (CHOICE .EQ. '2') THEN
           GO TO 80000
      ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
           RETURN
      END IF


C ************************ * * * * * ********************************
C                                                                     *
```

```
C This section for standard reports
C Note: To any one attempting to modify the programs;
C
C       Please leave the continuation characters as is on
C       format number 550; they represent the search numbers
C       on the NASA job specification
C                                                                *
C ************************ * * * * * ******************************

  400 CALL LIB$ERASE_PAGE (1,1)
      CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
      CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      IF (OPENPORT) THEN
        CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
        CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      OPENPORT = .FALSE.
      END IF
C  Include port2 definition commands

      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
      OPENPORT = .TRUE.

  500 TYPE 550
  550 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports:'
     2        /' -------------------------'
     e       //' 1 = For one country, find all designations'
     e              ' and U.S. equivalents'
     b        /' 2 = For one foreign alloy, find all similar'
     b              ' foreign alloys'
     i        /' 3 = Print the whole database'
     a        /' 4 = For one foreign alloy, find all similar'
     a              ' U.S. alloys'
     f        /' 5 = For one U.S. alloy, find similar foreign alloys'
     f              ' from one country'
     g        /' 6 = For one U.S. alloy, find similar foreign alloys'
     g              ' from all countries'
     c        /' 7 = For a Range of foreign alloys, find all similar'
     c              ' U.S. alloys'
     h        /' 8 = For a Range of U.S. alloys, find all similar'
     h              ' foreign alloys'

     K       //' H = Display Help Information'
     L        /' P = Return to the previous Menu'
     M        /' M = Return to the Main Menu'
     N       //' Type the number corresponding to your choice,'
     O        /' then hit the RETURN key'/)

  600 ACCEPT 650,CHOICE
  650 FORMAT (A)
      IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
           CALL LIB$ERASE_PAGE (1,1)
           CALL LIB$SPAWN ('stdrephelp')
           GO TO 400
      ELSE IF (CHOICE .EQ. '1') THEN
           GO TO 10000
      ELSE IF (CHOICE .EQ. '2') THEN
           GO TO 20000
      ELSE IF (CHOICE .EQ. '3') THEN
           GO TO 30000
      ELSE IF (CHOICE .EQ. '4') THEN
           GO TO 40000
```

```
      ELSE IF (CHOICE .EQ. '5') THEN
           GO TO 50000
      ELSE IF (CHOICE .EQ. '6') THEN
           GO TO 60000
      ELSE IF (CHOICE .EQ. '7') THEN
           GO TO 70000
c     ELSE IF (CHOICE .EQ. '8') THEN
c          GO TO 78000
      ELSE IF (CHOICE .EQ. '8') THEN
           GO TO 79000
      ELSE IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           GO TO 100
      ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
           RETURN
      ELSE
           type *,'Wrong entry, hit RETURN to try again'
           accept 700,answer
  700      format (A)
           go to 400
      END IF


C ************************ * * * * * ******************************
C                                                                   *
C Standard Reports: Option 1                                        *
C For one country, find all designations and U.S. equivalents       *
C                                                                   *
C Search 1.e                                                        *
C ************************ * * * * * ******************************

10000 CALL LIB$ERASE_PAGE (1,1)

10100 TYPE 10105
10105 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1      //' Producing Standard Reports: [1]'
     2       /' -------------------------'
     3      //' For one country, find all designations'
     4        ' and U.S. equivalents'

     5      //' 01=Australia     06=Finland      11=Norway'
     6        '         16=United Kingdom'
     7       /' 02=Canada        07=France       12=South Africa'
     8        '  17=U.S.A.'
     9       /' 03=China         08=Japan        13=Spain'
     A        '           18=U.S.S.R.'
     B       /' 04=Denmark       09=Mexico       14=Sweden'
     C        '         19=West Germany'
     D       /' 05=East Germany  10=New Zealand  15=Switzerland'
     E        '   20=Italy'
     F           /' 21=Belgium       22=Netherlands  23=Portugal'
     F        '      30=ISO'
     F      //' Please enter Country code from the table,'
     G        ' Then hit the RETURN key'
     H       /' Or, To return to the previous Menu, enter P'
     I       /' Then hit the RETURN key'/)

10200 ACCEPT 10205,CNTRY
10205 FORMAT (A2)
      IF ((CNTRY(1:1) .EQ. 'P') .OR. (CNTRY(1:1) .EQ. 'p')) THEN
           GO TO 400
      END IF
C Prepare subscript for country literals table
```

```
      CALL jsCNTRY (cntry,icntry,Xcntry)
      IF (ICNTRY .EQ. 31) THEN
          type *,' '
          type *,'Country Code out of range, hit RETURN to try again'
          type *,'Or type M, then hit RETURN to return to Main Menu'
          accept 10305,answer
10305     format (A)
          IF ((ANSWER .EQ. 'M') .OR. (ANSWER .EQ. 'm')) THEN
               RETURN
          ELSE
               GO TO 10000
          END IF
      END IF
      type *,' '
      type *,'Searching for records, Please stand by'

      CNTRYX = ''''//CNTRY//''''
10310 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH COUNTRY = !CMD;',
     1 DOMAIN,CNTRYX)
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          GO TO 90100
      END IF

C Investigate the number of records found,
C if no records were found then return to try another Rse

10320 CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')

      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF

      IF (NUM_RECS .EQ. 0) THEN
          type *,'No records found from that country'
          type *,'Hit RETURN to try another country'
          accept 10325,answer
10325     format (A)
          GO TO 10000
      END IF

C Program will branch here only if RSE has been successful,
10330 TYPE 10335
10335 FORMAT (/' Select one of the following options: then hit RETURN'
     1        /' 1 = Print only standard fields'
     2       //' 2 = Print all fields'
     3       //' P = Do not print, just return to the previous menu'/)

      ACCEPT 10337,CHOICE
10337 FORMAT (A)
      IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           GO TO 10000
      END IF
      IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
           type *,'Wrong entry, hit RETURN to try again'
           accept 10339,answer
10339      format (A)
```

```
          go to 10330
      END IF

10400 CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'DTR ERROR'
          type *,'Just hit RTN to continue'
          accept 10405,answer
10405     format (A)
          RETURN
      END IF

        CALL JSSETUP (PRT,CHOICE)

11000 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,' '
          type *,'The Report has been printed'
          type *,'Hit the RETURN key to continue'
          accept 11005,answer
11005     format (A)
          CALL LIB$SPAWN   ('SET TERM/WIDTH=80')
          GOTO 400
      END IF

11200 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
C Extract the country code from the fortran buffer to be used to
C   pull out the country literal from the literal pool
      CNTRY = CONTRY
      CALL jsCNTRY (cntry,icntry,Xcntry)

C     Print the detail line from the record buffer
      IF (CHOICE .EQ. '1') THEN
          CALL JSPRTFEW(prt)
        GOTO 11000
      END IF
      IF (CHOICE .EQ. '2') THEN
          CALL JSPRTALL(PRT,FINISH)
        IF (FINISH) THEN
            call lib$spawn ('set term/width=80')
          GOTO 400
        ELSE
          GOTO 11000
        END IF
      END IF


C ************************ * * * * * ******************************
C                                                                   *
C Standard Reports: Option 2
C For one foreign alloy, find all similar foreign alloys
C                                                                   *
C This is Search 1.b in the specifications
C                                                                   *
C ************************ * * * * * ******************************


20000 CALL LIB$ERASE_PAGE (1,1)
```

```
      TYPE 20005
20005 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports: [2]'
     2        /' -------------------------'
     3       //' For one foreign alloy, find all similar foreign alloys'
     4       //' Choose one of the following criteria:'
     5       //' 1 = Select by matching U.S. Equivalents'
     6        /' 2 = Select by matching chemical composition'
     7        /' P = Return to the previous menu'
     8        /' M = Return to the Main Menu'
     9        /' H = Display Help Information'
     A       //' Type the number corresponding to your choice'
     B        /' then hit the RETURN key'/)

20010 ACCEPT 20015,CHOICE
20015 FORMAT (A)

      IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
           CALL LIB$ERASE_PAGE (1,1)
           CALL LIB$SPAWN ('stdrephelp')
           GO TO 20000
      ELSE IF (CHOICE .EQ. '1') THEN
           GO TO 20100
      ELSE IF (CHOICE .EQ. '2') THEN
           GO TO 22000
      ELSE IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           GO TO 400
      ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
           RETURN
      ELSE
           type *,'Wrong entry, hit RETURN to try again'
           accept 20017,answer
20017      format (A)
           go to 20000
      END IF


C ****************************  * * * * * ******************************
C                                                                      *
C Selecting by matching U.S. Equivalents
C                                                                      *
C This is still Search 1.b in the specifications
C                                                                      *
C The following procedure/logic is used:
C 1.  Accept the foreign designation value
C 2.  Search the database for all foreign alloys with that designation
C 3.  From the above established collection, find the first record
C     with an US_Equivalent value.
C     Extract the first US_Equivalent value into a fortran field
C       If all of the records within the collection have blank
C       US_Equivalent fields, then print a warning message and
C       return to the previous menu
C 4.  Release all the records
C 5.  Search the whole data base for all foreign alloys that have
C       the same US_Equivalent
C 6.  Print/display the following information:
C                                                                      *
C                                                                      *
C ****************************  * * * * * ******************************
```

```
20100 CALL LIB$ERASE_PAGE (1,1)

      TYPE 20105
20105 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1      //' Producing Standard Reports: [2]'
     2       /' -------------------------'
     3      //' For one foreign alloy, find all similar foreign alloys'
     4       /' [Selecting by matching U.S. Equivalents]'

     5      //' Please enter the foreign designation number'
     6        ' within double quotation marks'
     7       /' Then Hit the RETURN Key'/)

C Step 1:

20110 ACCEPT 20115,DSGKEY
20115 FORMAT (A)

           type *,' '
           type *,'Searching for records, Please stand by'
           type *,' '
           type *,'Search start time is shown below'
           call lib$spawn ('ti')
           type *,' '

C Step 2: Establish a collection of all foreign alloys with the
C         the given designation

20300 CALL DTR$COMMAND (DAB, 'PORT2 = !CMD WITH DESIG = !CMD AND
     1 COUNTRY NOT = "17";',DOMAIN,DSGKEY)

20310 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'No foreign records found with that designation'
          type *,'Hit RETURN to try another designation'
          accept 20315,answer
20315     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      IF (OPENPORT) THEN
        CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
        CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      OPENPORT = .FALSE.
      END IF
          RETURN
      END IF

           type *,' '
           type *,'Search finish time is shown below'
           call lib$spawn ('ti')
           type *,' '

C Step 3:
C Retrieve (GET) 1 record with an US_Equivalent value
C        into the Buffer (FULLREC)

20600 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
C Test for a non-blank US_Equivalent field
       IF (EQUIV .EQ. ' ') THEN
          IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
              type *,'- All selected records have blank US_Equiv'
              type *,'Hit RETURN to try another designation'
              accept 20605,answer
```

```
20605          format (A)
               CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
             CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
             CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
           OPENPORT = .FALSE.
           END IF
               RETURN
           else
               go to 20600
           end if
       END IF

C At this point we know that we have a non-blank US_Equivalent
C We shall store the following values for later use

       Xfadb    = fadb
       Xdesig   = desg
       Xcountry = contry
       Xequiv   = equiv

C Step 4: Release all records from the current collection
C         NOTE:
C              This next routine extracts all the records from the
C              collection to force the DAB$W_STATE to change from
C              DTR$K_STL_PGET


20640 IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
           CALL DTR$GET_PORT (DAB, %REF(FULLREC))
           go to 20640
       else
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
             CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
             CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
           OPENPORT = .FALSE.
           END IF
       end if

C Declare a PORT PORT2 again for Storing all records
       INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
         OPENPORT = .TRUE.

C Step 5:  Search the whole data base for all foreign alloys that have
C          the same US_Equivalent
C The next statement converts the character data Xequiv into
C a literal within double quotes by concatenation
C Datatrieve would not just accept the Xequiv as stored above

         XEQUIV2 = '"'//XEQUIV//'"'

       CALL DTR$COMMAND (DAB, 'find !CMD WITH US_EQV = !CMD and
      1 country not = "17";',DOMAIN,Xequiv2)

           CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

20700 TYPE 20705
20705 FORMAT (/' Select one of the following options: then hit RETURN'
     1       //' 1 = Print only standard fields'
     2        /' 2 = Print all fields'
     3       //' P = Do not print, just return to the previous menu'/)
```

```
      ACCEPT 20715,CHOICE
20715 FORMAT (A)

      IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
           GO TO 20000
      END IF
      IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
           type *,'Wrong entry, hit RETURN to try again'
           accept 20725,answer
20725      format (A)
           go to 20700
      END IF

20730 CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'There was a fatal Datatrieve ERROR'
          type *,'Hit RETURN to go back to the Main Menu'
          accept 20735,answer
20735     format (A)
          RETURN
      END IF

C Choose between screen display and printed report
C NOTE:
C      We need to do this little routine B4 entering into
C      the record retrieval loop (based on DAB$W_STATE)

21010 TYPE 21015
21015 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)


      accept 21025,answer
21025 format (A)
          IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1        ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
          type *,'Wrong entry, please hit RETURN and try again'
               accept 21035,answer
21035          format (A)
               GO TO 21010
          END IF


      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          WRITE (*,21045)
21045     FORMAT (/' Your report will be stored in a temporary'
     1             ' data file e.g. PRTTEMP.DAT which'
     2            /' you may browse with EDT or print on your'
     3             ' local printer i.e. PRINT PRTTEMP.DAT'
     4            //' Choose and enter a name for your'
     5             ' temporary print file, e.g. PRTTEMP'/)
```

```
            ACCEPT 21055,FILENAME
21055       FORMAT (A)
            PRT = 3
            OPEN (3,FILE=FILENAME,STATUS='NEW')
      ELSE
            PRT = 5
      END IF
21060 CONTINUE
      IF ((ANSWER .EQ. 'S') .OR. (ANSWER .EQ. 's')) THEN
          CALL LIB$SPAWN ('SET TERM/WIDTH=132')
      END IF
      IPAGE  = 0
      RECPRT = 50

C Step 6: Retrieve (GET) 1 record at a time via PORT2 into
C         the Buffer (FULLREC) and print with fortran

21100 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'No foreign alloys match that US_Equivalent'
          type *,'Hit RETURN to try another designation'
          accept 21205,answer
21205     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          RETURN
      END IF

21300 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
C Extract the country code from the fortran buffer to be used to
C   pull out the country literal from the literal pool
      CNTRY = CONTRY
      CALL jsCNTRY (cntry,icntry,Xcntry)

C Step 7: Print from the FORTRAN Buffer FULLREC

      IF (CHOICE .EQ. '2') THEN
          go to 21400
      END IF

C This section for printing the Abbreviated report

      RECPRT = RECPRT+1
      IF (RECPRT .GT. 50) THEN
          RECPRT = 1
          IPAGE  = IPAGE+1
          CALL LIB$ERASE_PAGE (1,1)
          WRITE (PRT,21305) IPAGE,Xdesig,Xequiv
21305 FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
     1         '          Page ',I4,
     2       //' List of foreign Alloys similar to foreign alloy'
     3         ' with Designation: = ',A30,
     4        /' and US Equivalent = ',A30,
     5       //' [Similarity by matching US_Equivalents]'
     6       //' Rec. No. Designation',20X,'US_Equivalent',12X,
     7         ' Temper',10X,'Country',9X,'Form'/)
      END IF
```

```
      WRITE (PRT,21315) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
21315 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)

      IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,' '
          type *,'The report has been printed'
          type *,'Hit RETURN to go back to the Main Menu'
          accept 21325,answer
21325     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          CALL LIB$SPAWN ('SET TERM/WIDTH=80')
          return
      END IF
      go to 21300

C This section for full report

21400 CALL LIB$ERASE_PAGE (1,1)

      IPAGE = IPAGE+1
      WRITE (PRT,21405) IPAGE,Xdesig,Xequiv
21405 FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
     1        '                    Page ',I4,
     2       //' List of foreign Alloys similar to foreign alloy'
     3         ' with Designation: = ',A30,
     4        /' and US Equivalent = ',A30,
     5       //' [Similarity by matching US_Equivalents]'/)


      WRITE (PRT,21415) FADB,DESG,EQUIV,XCNTRY
21415 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1        'Country: ',A15/)

      WRITE (PRT,21425) ALTYP,TEMPR,FORMNUM,ORIGIN
21425 FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
     1        ' Orig. Org: ',A10/)

      WRITE (PRT,21435)
21435 FORMAT (' COMPOSITION:')

      WRITE (PRT,21445)
21445 FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1      'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2      'Pb',6X,'Sn')

      WRITE (PRT,21455) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1      MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
21455 FORMAT (8X,'MIN: ',14(A7,1X))

      WRITE (PRT,21465) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1      MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
21465 FORMAT (8X,'MAX: ',14(A7,1X)/)

      WRITE (PRT,21475) OTHER1,OTHER2,SPECS1
21475 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
```

```
      WRITE (PRT,21485) MINO1,MINO2,SPECS2
21485 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,21495) MAXO1,MAXO2,SPECS2
21495 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,21505) SPECS4
21505 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,21515) SPECS5
21515 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,21525) SCCRTG
21525 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,21535) MINYLD,MAXYLD,TYPYLD,YLUNIT
21535 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,21545) MINTNS,MAXTNS,TYPTNS,TNUNIT
21545 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 21900
      END IF

      type *,' '
      write (prt,21547)
21547 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)

      accept 21550,answer
21550 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if
21900 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'The report has been printed'
          type *,'Hit RETURN to go back to the Main Menu'
          accept 21905,answer
21905     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          call lib$spawn ('set term/width=80')
          return
      END IF

      go to 21300


C ******************************  * * * * * ******************************
C                                                                       *
C Selecting by matching chemical composition
C                                                                       *
C This is still Search 1.b in the specifications
C                                                                       *
C The following procedure/logic will be used:
C                                                                       *
C 1. Accept the record number whose composition is to be matched
C 2  Establish a one record collection and
C 3. Retrieve (GET) that 1 record via PORT2 into the Buffer (FULLREC)
C 4. Release the current collection
C 5. Redeclare PORT2
C 6. Use the composition data of the record in the buffer
```

```
C    to establish another collection
C 7. Retrieve (GET) 1 record at a time from the new collection via
C    PORT2 into the Buffer (FULLREC) and
C 8. Print the information from the FORTRAN Buffer FULLREC
C                                                        *
C ******************************  * * * * * ******************************



22000 CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      IF (OPENPORT) THEN
        CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
        CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      OPENPORT = .FALSE.
      END IF
      CALL DTR$COMMAND (DAB,'SHOW ALL')
      CALL DTR$DTR (DAB,DTR$M_OPT_CMD)

C   Include file of commands to declare port2

      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
        OPENPORT = .TRUE.

22002 CALL LIB$ERASE_PAGE (1,1)

      TYPE 22005
22005 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1      //' Producing Standard Reports: [2]'
     2       /' -------------------------'
     3      //' For one foreign alloy, find all similar foreign alloys'
     4       /' [Selecting by matching chemical composition]'
     5      //' Enter the record number whose chemical composition',
     6        ' is to be matched'
     7      //' Or, To return to the Main Menu, enter M'
     8       /' Then hit the RETURN key'/)


C Step 1:

22040 ACCEPT 22045,RECKEY
22045 FORMAT (A)

      IF ((RECKEY .EQ. 'M') .OR. (RECKEY .EQ. 'm')) THEN
           RETURN
      END IF
      reckeyx = '"'//reckey//'"'
            type *,' '
            type *,'Searching for the foreign record, Please stand by'
            type *,' '

C Step 2:  Search the database for that foreign record
        CALL DTR$PRINT_DAB(DAB)
22100 CALL DTR$COMMAND (DAB, 'PORT2 = !CMD WITH FADB_NO = !CMD',
     1DOMAIN,RECKEYX)

      type *,' '
      type *,'DAB Dump after 22100; search for foreign record'
            type *,' '
      CALL DTR$PRINT_DAB (DAB)
            type *,' '
```

```
C Step 3:
C Retrieve (GET) that 1 record from PORT2 into the Buffer (FULLREC)

22200 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'That foreign record was not found'
          type *,'Hit RETURN to try another record'
          accept 22205,answer
22205     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          GO TO 22000
      END IF

22300 CALL DTR$GET_PORT (DAB, %REF(FULLREC))

C Save the fadb, designation, country_of_origin, and U.S._Equivalent
C   for the heading

       Xfadb    = fadb
       Xdesig   = desg
       Xcountry = contry
       Xequiv   = equiv
       XcountryX = '"'//Xcountry//'"'
       XEQUIV2 = '"'//XEQUIV//'"'

C The following print statement is only a checkpoint

      write (5,22301) fadb,desg,XcountryX,minal,maxal,minsi,maxsi,
     1       minfe,maxfe,mincu,maxcu,minmn,maxmn,minmg,maxmg,sccrtg,
     2       tempr,xequiv2,units,minyld,mintns,ref1

22301 format (/' The following are checkpoint values',
     1        /' Fadb = ',A7,' Desig = ',A30,' Country code = ',A4,
     2        /' Min_Al = ',A7,' Max_Al = ',A7,' Min_Si = ',A7,
     3         ' Max_Si = ',A7,
     2        /' Min_Fe = ',A7,' Max_Fe = ',A7,' Min_Cu = ',A7,
     3         ' Max_Cu = ',A7,
     2        /' Min_Mn = ',A7,' Max_Mn = ',A7,' Min_Mg = ',A7,
     3         ' Max_Mg = ',A7,
     7        /' SCC Rating = ',A4,' Temper = ',A15,' Equiv = ',A27,
     8        /' Units = ',A,' Min_Yld = ',A3,' Min_Tns = ',A3,
     9         ' Reference #1 = ',A3,
     4       //' Just hit RETURN to continue')

      accept 22302,dummy
22302 format (A)


C Step 4:
C Release all records from the current collection

      CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      IF (OPENPORT) THEN
        CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
        CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
```

```
      OPENPORT = .FALSE.
      END IF

C   Include file for commands declaring port2

      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
        OPENPORT = .TRUE.

      type *,' '
      type *,'DAB Dump after declare PORT2 B4 composition search'
      type *,' '
      CALL DTR$PRINT_DAB (DAB)
      type *,' '


C Step 5:
C Use the composition data of the record in the buffer
C   to search the database and find all U.S. alloys
C   with the same composition

      type *,' '
      type *,'Searching for records with equal composition - Stand by'


22323 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH MIN-AL = !cmd AND
     1        MAX-AL = !cmd  AND MIN-SI = !cmd AND MAX-SI = !cmd
     2        AND MIN-FE = !CMD  AND MAX-FE = !CMD ',
     3        DOMAIN,MINAL,MAXAL,MINSI,MAXSI,MINFE,MAXFE)
C     3         AND MIN-FE = !cmd AND MAX-FE = !cmd
C     4         AND MIN-CU = !cmd AND MAX-CU = !cmd
C     4         AND MIN-MN = !cmd AND MAX-MN = !cmd
C     5         AND MIN-MG = !cmd AND MAX-MG = !cmd
C     6         AND MIN-ZN = !cmd AND MAX-ZN = !cmd
C     7         AND MIN-VD = !cmd AND MAX-VD = !cmd
C     8         AND MIN-TI = !cmd AND MAX-TI = !cmd
C     9         AND MIN-ZR = !cmd AND MAX-ZR = !cmd
C     A         AND MIN-CR = !cmd AND MAX-CR = !cmd
C     B         AND MIN-NI = !cmd AND MAX-NI = !cmd
C     C         AND MIN-PB = !cmd AND MAX-PB = !cmd
C     D         AND MIN-SN = !cmd AND MAX-SN = !cmd;',DOMAIN,
C     e             MINAL,MAXAL,MINSI,MAXSI,MINFE,MAXFE,
C     f             MINCU,MAXCU,MINMN,MAXMN,MINMG,MAXMG,
C     g             MINZN,MAXZN,MINV,MAXV,MINTI,MAXTI,
C     h             MINZR,MAXZR,MINCR,MAXCR,MINNI,MAXNI,
C     i             MINPB,MAXPB,MINSN,MAXSN)
C

       CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

      type *,' '
      type *,'DAB Dump after the search for equal composition'
      type *,' '
      type *,' '


C Investigate the number of records found,
C if no records were found then return to try another Rse

      CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')

      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
```

```
         CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF

      IF (NUM_RECS .EQ. 0) THEN
          type *,'No records found with the same composition'
          type *,'Hit RETURN to select another record'
          accept 22325,answer
22325     format (A)
          GO TO 22000
      END IF

      CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')
      type *,' '
      type *,'DAB Dump after PORT2 = CURRENT'
      type *,' '
      CALL DTR$PRINT_DAB (DAB)
      type *,' '
22330 TYPE 22335
22335 FORMAT (/' Select one of the following options: then hit RETURN'
     1       //' 1 = Print only standard fields'
     2        /' 2 = Print all fields'
     3       //' M = Do not print, just return to the Main Menu'/)

      ACCEPT 22345,CHOICE
22345 FORMAT (A)
      IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
           CALL DTR$COMMAND (DAB, 'finish ALL;')
           call lib$spawn ('set term/width=80')
           return
      END IF
      IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
           type *,'Wrong entry, hit RETURN to try again'
           accept 22355,answer
22355      format (A)
           go to 22330
      END IF

C Choose between screen display and printed report

22360 TYPE 22365
22365 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)
       accept 22375,answer
22375 format (A)
C Input Error-Trap
           IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1         ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
           type *,'Wrong entry, please hit RETURN and try again'
                accept 22385,answer
22385           format (A)
                GO TO 22360
           END IF
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
           WRITE (*,22395)
22395      FORMAT (/' Your report will be stored in a temporary'
     1              ' data file e.g. PRTTEMP.DAT which'
     2             /' you may browse with EDT or print on your'
     3              ' local printer i.e. PRINT PRTTEMP.DAT'
     4             //' Choose and enter a name for your'
     5              ' temporary print file, e.g. PRTTEMP'/)
```

```
            ACCEPT 22405,FILENAME
22405       FORMAT (A)
            PRT = 3
            OPEN (3,FILE=FILENAME,STATUS='NEW')
      ELSE
            PRT = 5
      END IF

C Step 6:
C Retrieve (GET) 1 record at a time from PORT2 into
C         the Buffer (FULLREC), and print from the buffer

22500 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
            type *,'The record has been printed'
            type *,'Hit the RETURN key to continue'
            accept 22505,answer
22505       format (A)
            CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
            IF (OPENPORT) THEN
              CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
              CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
              OPENPORT = .FALSE.
            END IF
            go to 22000
      END IF
22600 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
C Print the detail line from the fortran buffer
      IF (CHOICE .EQ. '2') THEN
            go to 22800
      END IF
C This section for printing the Abbreviated report
      RECPRT = RECPRT+1
      IF (RECPRT .GT. 50) THEN
            RECPRT = 1
            IPAGE  = IPAGE+1
            CALL LIB$ERASE_PAGE (1,1)
            WRITE (PRT,22605) IPAGE,Xfadb,Xdesig,Xcntry,Xequiv
22605 FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
     1         '           Page ',I4,
     2       //' List of foreign Alloys with exactly the same',
     3         ' composition as the foreign alloy'
     3         ' of record number = ',A7,' designation = ',A30,
     4         ' from country = ',A15,' and of U.S. Equivalent = ',A25,
     6       //' Rec. No. Designation',20X,'US_Equivalent',12X,
     7         ' Temper',10X,'Country',9X,'Form'/)
      END IF
      WRITE (PRT,22715) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
22715 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)

      GO TO 22500

C This section for full report
22800 CALL LIB$ERASE_PAGE (1,1)
      IPAGE = IPAGE+1
      WRITE (PRT,22605) IPAGE,Xfadb,Xdesig,Xcntry,Xequiv
22805 FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
     1         '                      Page ',I4,
     2       //' List of foreign Alloys with exactly the same',
     3         ' composition as the foreign alloy'
     3         ' of record number = ',A7,' designation = ',A30,
     4         ' from country = ',A15,' and of U.S. Equivalent = ',A25/)
```

```
      WRITE (PRT,22815) FADB,DESG,EQUIV,XCNTRY
22815 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1        'Country: ',A15/)

      WRITE (PRT,22825) ALTYP,TEMPR,FORMNUM,ORIGIN
22825 FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
     1        ' Orig. Org: ',A10/)

      WRITE (PRT,22835)
22835 FORMAT (' COMPOSITION:')

      WRITE (PRT,22845)
22845 FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1      'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2      'Pb',6X,'Sn')
      WRITE (PRT,22855) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1       MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
22855 FORMAT (8X,'MIN: ',14(A7,1X))
      WRITE (PRT,22865) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1       MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
22865 FORMAT (8X,'MAX: ',14(A7,1X)/)
      WRITE (PRT,22875) OTHER1,OTHER2,SPECS1
22875 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
      WRITE (PRT,22885) MINO1,MINO2,SPECS2
22885 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,22895) MAXO1,MAXO2,SPECS2
22895 FORMAT (8X,'MAX: ',A8,6X,A8,61X,'[3] ',A30)
      WRITE (PRT,22905) SPECS4
22905 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,22915) SPECS5
22915 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,22925) SCCRTG
22925 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,22935) MINYLD,MAXYLD,TYPYLD,YLUNIT
22935 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,22945) MINTNS,MAXTNS,TYPTNS,TNUNIT
22945 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 22970
      END IF
      type *,' '
      write (prt,22955)
22955 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)
      accept 22965,answer
22965 format (A)

      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if
22970 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'The report has been printed'
          type *,'Hit RETURN to go back to the Main Menu'
          accept 22975,answer
22975     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
```

```
          call lib$spawn ('set term/width=80')
          return
      END IF
      GO TO 22500

C ************************ * * * * * ******************************
C                                                                   *
C Option 3
C Print the whole database
C                                                                   *
C ************************ * * * * * ******************************

30000 Continue          !UNITY = 'Wt %'

C Step 1:

C PORT2 is declared in SBREPORT to hold any DTR collection

30100 CALL DTR$COMMAND (DAB, 'PORT2 = !CMD WITH
     1 DESIG NOT = "XXXXX";',DOMAIN)

C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'DTR ERROR'
          type *,'Just hit RTN to continue'
          accept 30205,answer
30205     format (A)
          RETURN
      END IF

C Step 2:

C The above command causes the DTR$K_STL_PGET stall point
C While at this DTR stall point, we will continue to use
C DTR$GET_PORT to copy one record at a time from the port
C into our Fortran record buffer FULLREC

      CALL JSSETUP (PRT,2)    !Set up to print all fields

C IF NEXT CONDITION IS TRUE THEN RESET DOMAIN B4 RETURN
30300 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,' '
          type *,'- no more records to print'
          type *,'Just hit RTN to continue'
          accept 30405,answer
30405     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          RETURN
      END IF

30500 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
      CNTRY = CONTRY
      CALL jsCNTRY (cntry,icntry,Xcntry)

C   Print initial heading
```

```
c32100 WRITE (PRT,32105)
c32105 FORMAT (T20,' NASA ALLOY DATABASE FULL REPORT'
c     1       /T20,' ------------------------------'
c     2      //T21,' Listing of the Whole Database'
c     2       /T21,' ---------------------------')

       CALL JSPRTALL (PRT,FINISH)
       IF (FINISH) THEN
           CALL LIB$SPAWN ('SET TERM/WIDTH=80')
           GOTO 400
       ELSE
           GOTO 30300
       END IF

C         ******** END OF PRINT ALL RECORDS ********

C ************************ * * * * * ******************************
C                                                                  *
C Option 4
C For one foreign alloy, find all similar U.S. alloys
C                                                                  *
C This is Search 1a in the specifications
C                                                                  *
C ************************ * * * * * ******************************

40000 CALL LIB$ERASE_PAGE (1,1)
40002 TYPE 40005
40005 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports:'
     2        /' ------------------------'
     3       //' For one foreign alloy, find all similar U.S. alloys'
     4        /' Select one of the following:'
     5       //' 1 = Selecting by predetermined U.S. Equivalents'
     6        /' 2 = Select by matching chemical composition'
     7        /' P = Return to the previous menu'
     8        /' M = Return to the Main Menu'
     9        /' H = Display Help Information'
     A       //' Type the number corresponding to your choice'
     B        /' then hit the RETURN key'/)

40010 ACCEPT 40015,CHOICE
40015 FORMAT (A)

      IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
           CALL LIB$ERASE_PAGE (1,1)
           CALL LIB$SPAWN ('stdrephelp')
           GO TO 40000
      ELSE IF (CHOICE .EQ. '1') THEN
           GO TO 40100
      ELSE IF (CHOICE .EQ. '2') THEN
           GO TO 42000
      ELSE IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           GO TO 400
      ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
           RETURN
      ELSE
           type *,'Wrong entry, hit RETURN to try again'
           accept 40017,answer
40017      format (A)
           go to 40000
      END IF
```

```
C ********************** * * * * * ******************************
C                                                                *
C This is still Search 1a in the specifications
C                                                                *
C Listing predetermined US Equivalents,
C The following logic shall be used:
C 1.  Accept the foreign designation value
C 2.  Search the database for any (and all) non-U.S.  records
C                with that designation
C 3.  Print/display the records
C                                                                *
C ********************** * * * * * ******************************


40100 CALL LIB$ERASE_PAGE (1,1)
40102 TYPE 40105
40105 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports: [4]'
     2        /' -------------------------'
     3       //' For one foreign alloy, find all similar U.S. alloys'
     4        /' [Selecting predetermined U.S. Equivalents]'
     5       //' Enter Alloy Designation within double quotation marks'
     6        /' Then hit the RETURN key'/)

C Step 1:
40110 ACCEPT 40115,DSGKEY
40115 FORMAT (A)
            type *,' '
            type *,'Searching for records, Please stand by'

C   Include file for declaring PORT2
      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
        OPENPORT = .TRUE.
c  Search for foreign alloys woth that designation
40120 CALL DTR$COMMAND (DAB, 'PORT2 = !CMD WITH DESIG = !CMD AND
     1 COUNTRY NOT = "17";', DOMAIN,DSGKEY)

C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'DTR ERROR'
          type *,'Just hit RTN to continue'
          accept 40125,answer
40125     format (A)
          RETURN
      END IF

C Choose type of report
40200 TYPE 40205
40205 FORMAT (/' Select one of the following options: then hit RETURN'
     1       //' 1 = Print only standard fields'
     2        /' 2 = Print all fields'
     3       //' P = Do not print, just return to the previous menu'/)

      ACCEPT 40215,CHOICE
40215 FORMAT (A)

      IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
```

```
            IF (OPENPORT) THEN
              CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
              CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
            OPENPORT = .FALSE.
            END IF
             GO TO 40000
       END IF
       IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
            type *,'Wrong entry, hit RETURN to try again'
            accept 40225,answer
40225       format (A)
            go to 40200
       END IF

C Choose between screen display and printed report
40300 TYPE 40305
40305 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)

      accept 40315,answer
40315 format (A)

C Input Error-Trap

            IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1          ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 40325,answer
40325            format (A)
                 GO TO 40300
            END IF

       IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
            WRITE (*,40335)
40335       FORMAT (/' Your report will be stored in a temporary'
     1              ' data file e.g. PRTTEMP.DAT which'
     2             /' you may brouse with EDT or print on your'
     3              ' local printer i.e. PRINT PRTTEMP.DAT'
     4             //' Choose and enter a name for your'
     5              ' temporary print file, e.g. PRTTEMP'/)

            ACCEPT 40345,FILENAME
40345       FORMAT (A)
            PRT = 3
            OPEN (3,FILE=FILENAME,STATUS='NEW')
       ELSE
            PRT = 5
       END IF

40400  CALL LIB$ERASE_PAGE (1,1)
       IF ((ANSWER .EQ. 'S') .OR. (ANSWER .EQ. 's')) THEN
            CALL LIB$SPAWN ('SET TERM/WIDTH=132')
       END IF
       IPAGE  = 0
       RECPRT = 50

40500  IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
            type *,' '
            type *,'no records found - Just hit RTN to try again'
            accept 40505,answer
40505       format (A)
```

```
            CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
            IF (OPENPORT) THEN
              CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
              CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
            OPENPORT = .FALSE.
            END IF
            go to 40000
         END IF

 40510 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
 C Step 7: Print from the FORTRAN Buffer FULLREC
       IF (CHOICE .EQ. '2') THEN
           go to 40700
       END IF
 C This section for printing the Abbreviated report
       RECPRT = RECPRT+1
       IF (RECPRT .GT. 50) THEN
           RECPRT = 1
           IPAGE  = IPAGE+1
           CALL LIB$ERASE_PAGE (1,1)
           WRITE (PRT,40605) IPAGE,desg
 40605     FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
      1             '           Page ',I4,
      2           //' Foreign Alloys with Designation: = ',A30,
      3            /' Listed with their corresponding pre-determined'
      4             ' US_Equivalents'
      5           //' Rec. No. Designation',20X,'US_Equivalent',12X,
      6             ' Temper',10X,'Country',9X,'Form'/)
       END IF

       WRITE (PRT,40615) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
 40615 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
       IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
           type *,' '
           type *,'The report has been printed'
           type *,'Hit RETURN to go back to the Main Menu'
           accept 40625,answer
 40625     format (A)
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
             CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
             CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
             OPENPORT = .FALSE.
           END IF
           CALL LIB$SPAWN ('SET TERM/WIDTH=80')
           return
       END IF

       go to 40510

 C This section for full report
 40700 CALL LIB$ERASE_PAGE (1,1)
           IPAGE = IPAGE+1
           WRITE (PRT,40705) IPAGE,desg
 40705     FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
      1             '           Page ',I4,
      2           //' Foreign Alloys with Designation: = ',A30,
      3            /' Listed with their corresponding pre-determined'
      4             ' US_Equivalents'
      5           //' Rec. No. Designation',20X,'US_Equivalent',12X,
      6             ' Temper',10X,'Country',9X,'Form'/)
```

```
      WRITE (PRT,40715) FADB,DESG,EQUIV,XCNTRY
40715 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1        'Country: ',A15/)
      WRITE (PRT,40725) ALTYP,TEMPR,FORMNUM,ORIGIN
40725 FORMAT (' Type: ',A4,'    Temper: ',A15,' Form: ',A30,
     1        ' Orig. Org: ',A10/)
      WRITE (PRT,40735)
40735 FORMAT (' COMPOSITION:')
      WRITE (PRT,40745)
40745 FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1     'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2     'Pb',6X,'Sn')
      WRITE (PRT,40755) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1     MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
40755 FORMAT (8X,'MIN: ',14(A7,1X))
      WRITE (PRT,40765) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1     MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
40765 FORMAT (8X,'MAX: ',14(A7,1X)/)
      WRITE (PRT,40775) OTHER1,OTHER2,SPECS1
40775 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
      WRITE (PRT,40785) MINO1,MINO2,SPECS2
40785 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,40795) MAXO1,MAXO2,SPECS2
40795 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,40805) SPECS4
40805 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,40815) SPECS5
40815 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,40825) SCCRTG
40825 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,40835) MINYLD,MAXYLD,TYPYLD,YLUNIT
40835 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,40845) MINTNS,MAXTNS,TYPTNS,TNUNIT
40845 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 40900
      END IF

      type *,' '
      write (prt,40855)
40855 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)
      accept 40865,answer
40865 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if

40900 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'The report has been printed'
          type *,'Hit RETURN to go back to the Main Menu'
          accept 40905,answer
40905     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          call lib$spawn ('set term/width=80')
          return
```

```
      END IF
      go to 40510

C ***************************** * * * * * ****************************

C This is still Search 1a in the specifications

C Selecting by matching chemical composition,
C The following logic shall be used:

C 1.  Accept the record number of the foreign alloy
C 2.  Search the database for that foreign record
C        and store this record in a fortran buffer
C 3.  Release all collections
C 4.  Search the whole database to find all U.S. alloys with
C        exactly the same composition as that foreign alloy
C        which is sitting in the fortran buffer


42000 CALL LIB$ERASE_PAGE (1,1)

      CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
      IF (OPENPORT) THEN
        CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
        CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      OPENPORT = .FALSE.
      END IF

C   Include file for declaring PORT2
      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
        OPENPORT = .TRUE.
      TYPE 42005
42005 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports: [4]'
     2        /' -------------------------'
     3       //' For one foreign alloy, find all similar U.S. alloys'
     4        /' [Selecting by matching chemical composition]'
     5       //' Enter the record number whose chemical composition',
     6         ' is to be matched'
     7       //' Or, To return to the Main Menu, enter M'
     8        /' Then hit the RETURN key'/)

C Step 1:
      ACCEPT 42015,RECKEY
42015 FORMAT (A)
      IF ((RECKEY .EQ. 'M') .OR. (RECKEY .EQ. 'm')) THEN
           RETURN
      END IF
           type *,' '
           type *,'Searching for the record, Please stand by'
C Step 2:  Search the database for that foreign record

42025 CALL DTR$COMMAND (DAB, 'PORT2 = !CMD WITH FADB_NO = !CMD AND
     1 COUNTRY NOT = "17";',DOMAIN,RECKEY)

c            type *,' '
c      type *,'DAB Dump after searching for the foreign recrd'
c            type *,' '
c      CALL DTR$PRINT_DAB (DAB)
c            type *,' '
```

```
C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'Fatal Datatrieve ERROR'
          type *,'Hit RETURN to restart'
          accept 42035,answer
42035     format (A)
          RETURN
      END IF

C Step 3:
C Retrieve (GET) that 1 record from PORT2 into the Buffer (FULLREC)

42100 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'That foreign record was not found'
          type *,'Hit RETURN to try another record'
          accept 42105,answer
42105     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          GO TO 42000
      END IF


42200 CALL DTR$GET_PORT (DAB, %REF(FULLREC))


C Save the fadb, designation, country_of_origin, and U.S._Equivalent
C   for the heading
       Xfadb    = fadb
       Xdesig   = desg
       Xcountry = contry
       Xequiv   = equiv
C Convert the character composition data into
C literal data within double quotes by concatenation

        minalx = '"'//minal//'"'
        maxalx = '"'//maxal//'"'
        minsix = '"'//minsi//'"'
        maxsix = '"'//maxsi//'"'
        minfex = '"'//minfe//'"'
        maxfex = '"'//maxfe//'"'
        mincux = '"'//mincu//'"'
        maxcux = '"'//maxcu//'"'
        minmnx = '"'//minmn//'"'
        maxmnx = '"'//maxmn//'"'
        minmgx = '"'//minmg//'"'
        maxmgx = '"'//maxmg//'"'
        minznx = '"'//minzn//'"'
        maxznx = '"'//maxzn//'"'
        MINVx = '"'//MINV//'"'
        MAXVx = '"'//MAXV//'"'
        mintix = '"'//minti//'"'
        maxtix = '"'//maxti//'"'
        minzrx = '"'//minzr//'"'
        maxzrx = '"'//maxzr//'"'
```

```
          mincrx = '"'//mincr//'"'
          maxcrx = '"'//maxcr//'"'
          minnix = '"'//minni//'"'
          maxnix = '"'//maxni//'"'
          minpbx = '"'//minpb//'"'
          maxpbx = '"'//maxpb//'"'
          minsnx = '"'//minsn//'"'
          maxsnx = '"'//maxsn//'"'


C Step 4:
C Release the current collection
C Note: Only one record was selected earlier and retrieved

      CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
      IF (OPENPORT) THEN
        CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
        CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
      OPENPORT = .FALSE.
      END IF

C   Include file for declaring PORT2

      INCLUDE '[NASA3.JSEXREC]JSPORT25.INC'
        OPENPORT = .TRUE.

C Step 5:
C Use the composition data of the record in the buffer
C   to search the database and find all U.S. alloys
C   with the same composition
      type *,' '
      type *,'Searching for U.S. records with equal composition'
42210 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH COUNTRY NOT = "17"
     1         AND MIN-AL = !cmd AND MAX-AL = !cmd
     2         AND MIN-SI = !cmd AND MAX-SI = !cmd
     3         AND MIN-FE = !cmd AND MAX-FE = !cmd
     4         AND MIN-CU = !cmd AND MAX-CU = !cmd
     5         AND MIN-MN = !cmd AND MAX-MN = !cmd
     6         AND MIN-MG = !cmd AND MAX-MG = !cmd
     7         AND MIN-ZN = !cmd AND MAX-ZN = !cmd
     8         AND MIN-V  = !cmd AND MAX-V  = !cmd
     9         AND MIN-TI = !cmd AND MAX-TI = !cmd
     A         AND MIN-ZR = !cmd AND MAX-ZR = !cmd
     B         AND MIN-CR = !cmd AND MAX-CR = !cmd
     C         AND MIN-NI = !cmd AND MAX-NI = !cmd
     D         AND MIN-PB = !cmd AND MAX-PB = !cmd
     E         AND MIN-SN = !cmd AND MAX-SN = !cmd;',DOMAIN,
     f             MINALX,MAXALX,MINSIX,MAXSIX,MINFEX,MAXFEX,
     g             MINCUX,MAXCUX,MINMNX,MAXMNX,MINMGX,MAXMGX,
     h             MINZNX,MAXZNX,MINVX,MAXV,MINTIX,MAXTIX,
     i             MINZRX,MAXZRX,MINCRX,MAXCRX,MINNIX,MAXNIX,
     j             MINPBX,MAXPBX,MINSNX,MAXSNX)

      type *,' '
      type *,'DAB Dump after the search for equal composition'
      type *,' '
      CALL DTR$PRINT_DAB (DAB)
      type *,' '

C Investigate the number of records found,
C if no records were found then return to try another Rse
      CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')
```

```
        IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
            CALL DTR$GET_PORT (DAB, NUM_RECS)
            CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
        END IF
        IF (NUM_RECS .EQ. 0) THEN
            type *,'No records found with the same composition'
            type *,'Hit RETURN to select another record'
            accept 42225,answer
 42225      format (A)
            GO TO 42000
        END IF
        CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')
        type *,' '
        type *,'DAB Dump after record was found, and PORT2 = CURRENT'
        type *,' '
        CALL DTR$PRINT_DAB (DAB)
        type *,' '

 42230 TYPE 42235
 42235 FORMAT (/' Select one of the following options: then hit RETURN'
      1       //' 1 = Print only standard fields'
      2        /' 2 = Print all fields'
      3       //' M = Do not print, just return to the Main Menu'/)

       ACCEPT 42245,CHOICE
 42245 FORMAT (A)
       IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
            CALL DTR$COMMAND (DAB, 'finish ALL;')
            call lib$spawn ('set term/width=80')
            return
       END IF
       IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
            type *,'Wrong entry, hit RETURN to try again'
            accept 42247,answer
 42247      format (A)
            go to 42230
       END IF

C Choose between screen display and printed report
 42250 TYPE 42255
 42255 FORMAT (/' Do you want to display the report on the screen'
      1        /' or print it to a temporary file for later use?'
      2       //' Please respond with S or F:'/)

       accept 42257,answer
 42257 format (A)
C Input Error-Trap

            IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
      1         ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 42265,answer
 42265           format (A)
                 GO TO 42250
            END IF

       IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
            WRITE (*,42275)
 42275      FORMAT (/' Your report will be stored in a temporary'
      1              ' data file e.g. PRTTEMP.DAT which'
      2             /' you may browse with EDT or print on your'
      3              ' local printer i.e. PRINT PRTTEMP.DAT'
```

```
     4            //' Choose and enter a name for your'
     5              ' temporary print file, e.g. PRTTEMP'/)

           ACCEPT 42285,FILENAME
42285      FORMAT (A)
           PRT = 3
           OPEN (3,FILE=FILENAME,STATUS='NEW')
      ELSE
           PRT = 5
      END IF

C Step 6:
C Retrieve (GET) 1 record at a time from PORT2 into
C         the Buffer (FULLREC), and print from the buffer

42300 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
           type *,'The record has been printed'
           type *,'Hit the RETURN key to continue'
           accept 42305,answer
42305      format (A)
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
             CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
             CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
           OPENPORT = .FALSE.
           END IF
           go to 42000
      END IF

42310 CALL DTR$GET_PORT (DAB, %REF(FULLREC))
C Print the detail line from the fortran buffer
      IF (CHOICE .EQ. '2') THEN
          go to 42400
      END IF

C This section for printing the Abbreviated report
      RECPRT = RECPRT+1
      IF (RECPRT .GT. 50) THEN
          RECPRT = 1
          IPAGE  = IPAGE+1
          CALL LIB$ERASE_PAGE (1,1)
          WRITE (PRT,42315) IPAGE,Xfadb,Xdesig,Xcntry,Xequiv
42315 FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
     1         '            Page ',I4,
     2       //' List of U.S. Alloys with exactly the same',
     3         ' composition as the foreign alloy'
     4         ' of record number = ',A7,' designation = ',A30,
     5         ' from country = ',A15,' and of U.S. Equivalent = ',A25,
     6       //' Rec. No. Designation',20X,'US Equivalent',12X,
     7         ' Temper',10X,'Country',9X,'Form'/)
      END IF

      WRITE (PRT,42325) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
42325 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
      GO TO 42300


C This section for full report

42400 CALL LIB$ERASE_PAGE (1,1)

      IPAGE = IPAGE+1
```

```
      WRITE (PRT,42405) IPAGE,Xfadb,Xdesig,Xcntry,Xequiv
42405 FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
     1        '                        Page ',I4,
     2       //' List of U.S. Alloys with exactly the same',
     3         ' composition as the foreign alloy'
     4         ' of record number = ',A7,' designation = ',A30,
     5         ' from country = ',A15,' and of U.S. Equivalent = ',A25/)


      WRITE (PRT,42415) FADB,DESG,EQUIV,XCNTRY
42415 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1        'Country: ',A15/)

      WRITE (PRT,42425) ALTYP,TEMPR,FORMNUM,ORIGIN
42425 FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
     1        ' Orig. Org: ',A10/)

      WRITE (PRT,42435)
42435 FORMAT (' COMPOSITION:')

      WRITE (PRT,42445)
42445 FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1     'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2     'Pb',6X,'Sn')

      WRITE (PRT,42455) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1      MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
42455 FORMAT (8X,'MIN: ',14(A7,1X))
      WRITE (PRT,42465) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1      MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
42465 FORMAT (8X,'MAX: ',14(A7,1X)/)
      WRITE (PRT,42475) OTHER1,OTHER2,SPECS1
42475 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
      WRITE (PRT,42485) MINO1,MINO2,SPECS2
42485 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,42495) MAXO1,MAXO2,SPECS2
42495 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,42505) SPECS4
42505 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,42515) SPECS5
42515 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,42525) SCCRTG
42525 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,42535) MINYLD,MAXYLD,TYPYLD,YLUNIT
42535 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,42545) MINTNS,MAXTNS,TYPTNS,TNUNIT
42545 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)

      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 42570
      END IF
      type *,' '
      write (prt,42555)
42555 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)
      accept 42565,answer
42565 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if
```

```
42570 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,'The report has been printed'
          type *,'Hit RETURN to go back to the Main Menu'
          accept 42575,answer
42575     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          IF (OPENPORT) THEN
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          call lib$spawn ('set term/width=80')
          return
      END IF

      GO TO 42300


C ************************ * * * * * ******************************
C                                                                   *
C Standard Reoports: Option 5                                       *
C                                                                   *
C For one U.S. alloy, find all similar foreign alloys from          *
C one country:   This is Search 1.f in the specifications           *
C                                                                   *
C The following logic is used:                                      *
C 1. Accept the U.S. Designation as input                           *
C 2. Accept the foreign country code as input [to search from]      *
C                                                                   *
C Just search for all foreign alloys from the given country         *
C with U.S. Equivalents equal to that U.S. alloy Designation        *
C                                                                   *
C ************************ * * * * * ******************************

50000 CALL LIB$ERASE_PAGE (1,1)
50010 TYPE 50015
50015 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports: [5]'
     2        /' -------------------------'
     3       //' For one U.S. alloy, find all similar foreign'
     4         ' alloys from one country'
     5       //' Enter the U.S. designation and foreign contry code'

     6       //' 01=Australia      06=Finland       11=Norway'
     7         '          16=United Kingdom'
     8        /' 02=Canada         07=France        12=South Africa'
     9         '  17=U.S.A.'
     A        /' 03=China          08=Japan         13=Spain'
     B         '           18=U.S.S.R.'
     C        /' 04=Denmark        09=Mexico        14=Sweden'
     D         '         19=West Germany'
     E        /' 05=East Germany   10=New Zealand   15=Switzerland'
     F         '   20=Italy'
     F        /' 21=Belgium        22=Netherlands   23=Portugal'
     F         '       30=ISO'
     G       //' First, enter U.S. designation within double'
     H         ' quotation marks',
     I        /' and hit the RETURN key'/)

50020 ACCEPT 50025,DSGKEY
50025 FORMAT (A)
```

```
50030 TYPE 50035
50035 FORMAT (/' Now, enter Country code from the table,'
     1          ' Then hit the RETURN key'
     2         /' Or, To return to the previous Menu, enter P'
     3         /' Then hit the RETURN key'/)

      ACCEPT 50045,CNTRY
50045 FORMAT (A2)

      IF ((CNTRY(1:1) .EQ. 'P') .OR. (CNTRY(1:1) .EQ. 'p')) THEN
           GO TO 400
      END IF

C Extract country text from the country literals table

      CALL jsCNTRY (cntry,icntry,Xcntry)
      IF (ICNTRY .EQ. 31) THEN
          type *,' '
          type *,'Country Code out of range, hit RETURN to try again'
          type *,'Or type M, then hit RETURN to return to Main Menu'
          accept 50055,answer
50055     format (A)
          IF ((ANSWER .EQ. 'M') .OR. (ANSWER .EQ. 'm')) THEN
               RETURN
          ELSE
               GO TO 50030
          END IF
      END IF

      type *,' '
      type *,'Searching for records, Please stand by'
      CALL DTR$COMMAND (DAB, 'FIND !CMD WITH COUNTRY = !CMD AND
     1 DESIG = !CMD;',DOMAIN,CNTRY,DSGKEY)
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
           type *,'There was a Fatal Datatrieve ERROR'
           type *,'Hit RETURN to restart'
           accept 50065,answer
50065      format (A)
           RETURN
      END IF


C Investigate the number of records found,
C if no records were found then return to try another Rse

      CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')
      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF
      IF (NUM_RECS .EQ. 0) THEN
          type *,'No similar alloys found from that country'
          type *,'Hit RETURN to try another country'
          accept 50075,answer
50075     format (A)
          GO TO 50000
      END IF
```

```
C Program will branch here only if RSE has been successful,

50100 TYPE 50105
50105 FORMAT (/' Select one of the following options: then hit RETURN'
     1        /' 1 = Print only standard fields'
     2       //' 2 = Print all fields'
     3       //' P = Do not print, just return to the previous menu'/)

      ACCEPT 50115,CHOICE
50115 FORMAT (A)
      IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           GO TO 400
      END IF
      IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
           type *,'Wrong entry, hit RETURN to try again'
           accept 50125,answer
50125      format (A)
           go to 50100
      END IF
      CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')
C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
           type *,'There was a Fatal Datatrieve ERROR'
           type *,'Hit RETURN to restart'
           accept 50135,answer
50135      format (A)
           RETURN
      END IF

C Choose between screen display and printed report
50140 TYPE 50145
50145 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)
      accept 50155,answer
50155 format (A)
C Input Error-Trap

           IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1         ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
           type *,'Wrong entry, please hit RETURN and try again'
                accept 50165,answer
50165           format (A)
                GO TO 50140
           END IF

      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
           WRITE (*,10715)
10715      FORMAT('DUMMY FORMAT')
           ACCEPT 50175,FILENAME
50175      FORMAT (A)
           PRT = 3
           OPEN (3,FILE=FILENAME,STATUS='NEW')
      ELSE
           PRT = 5
      END IF
      CALL LIB$ERASE_PAGE (1,1)
      IF ((ANSWER .EQ. 'S') .OR. (ANSWER .EQ. 's')) THEN
           CALL LIB$SPAWN ('SET TERM/WIDTH=132')
```

```
       END IF
       IPAGE  = 0
       RECPRT = 50
 50200 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
           type *,' '
           type *,'The Report has been printed'
           type *,'Hit the RETURN key to continue'
           accept 50205,answer
 50205     format (A)
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
             CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
             CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
           OPENPORT = .FALSE.
           END IF
           CALL LIB$SPAWN    ('SET TERM/WIDTH=80')
           RETURN
       END IF


 50210 CALL DTR$GET_PORT (DAB, %REF(FULLREC))

      IF (CHOICE .EQ. '2') THEN
          go to 50300
       end if

 C  This section for abbreviated report
       RECPRT = RECPRT+1
       IF (RECPRT .GT. 50) THEN
           RECPRT = 1
           IPAGE  = IPAGE+1
           CALL LIB$ERASE_PAGE (1,1)
           WRITE (PRT,50215)  IPAGE, Xcntry,desg
 50215 FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
      1         '           Page ',I4,
      2        /' List of Alloys from ',A15,' that are similar'
      3         ' to the U.S. alloy designation ',A30,
      4       //' Rec. No. Designation',20X,'US_Equivalent',12X,
      5         ' Temper',10X,'Country',9X,'Form'/)
       END IF
       WRITE (PRT,50225) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
 50225 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
       GO TO 50200

 C This section for full report

 50300 CALL LIB$ERASE_PAGE (1,1)
       IPAGE = IPAGE+1
       WRITE (PRT,50305)  IPAGE,Xcntry,desg
 50305 FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
      1         '                      Page ',I4,
      2        /' List of Alloys from ',A15,' that are similar'
      3         ' to the U.S. alloy designation ',A30//)

       WRITE (PRT,50315) FADB,DESG,EQUIV,XCNTRY
 50315 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
      1         'Country: ',A15/)
       WRITE (PRT,50325) ALTYP,TEMPR,FORMNUM,ORIGIN
 50325 FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
      1         ' Orig. Org: ',A10/)
       WRITE (PRT,50335)
 50335 FORMAT (' COMPOSITION:')
```

```
      WRITE (PRT,50345)
50345 FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1      'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2      'Pb',6X,'Sn')
      WRITE (PRT,50355) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1      MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
50355 FORMAT (8X,'MIN: ',14(A7,1X))
      WRITE (PRT,50365) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1      MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
50365 FORMAT (8X,'MAX: ',14(A7,1X)/)
      WRITE (PRT,50375) OTHER1,OTHER2,SPECS1
50375 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
      WRITE (PRT,50385) MINO1,MINO2,SPECS2
50385 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,50395) MAXO1,MAXO2,SPECS2
50395 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,50405) SPECS4
50405 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,50415) SPECS5
50415 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,50425) SCCRTG
50425 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,50435) MINYLD,MAXYLD,TYPYLD,YLUNIT
50435 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,50445) MINTNS,MAXTNS,TYPTNS,TNUNIT
50445 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 50460
      END IF
      type *,' '
      write (prt,50447)
50447 format (' Hit the RETURN key to continue printing, To stop',
     1       ' printing type S, then Hit the RETURN key: ',$)
      accept 50450,answer
50450 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if

50460 GO TO 50200


C ************************ * * * * * ******************************
C                                                                   *
C Option 6                                                          *
C For one U.S. alloy, find all similar foreign alloys               *
C     from all countries                                            *
C                                                                   *
C This is Search 1.g in the specifications                          *
C                                                                   *
C The following logic is used:                                      *
C 1. Accept the U.S. Designation as input                           *
C                                                                   *
C Just search for all foreign alloys with U.S. Equivalents          *
C equal to that U.S. alloy Designation                              *
C                                                                   *
C                                                                   *
C ************************ * * * * * ******************************


60000 CALL LIB$ERASE_PAGE (1,1)
```

```
      TYPE 60015
60015 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports: [6]'
     2        /' -------------------------'
     3       //' For one U.S. alloy, find all similar foreign'
     4         ' alloys from all countries'
     5       //' Enter U.S. designation within double quotation marks'
     6        /' and hit the RETURN key'
     7       //' Or, To return to the previous Menu, enter P'
     8        /' Then hit the RETURN key'/)


      ACCEPT 60025,DSGKEY
60025 FORMAT (A)

      IF ((DSGKEY(1:1) .EQ. 'P') .OR. (DSGKEY(1:1) .EQ. 'p')) THEN
           GO TO 400
      END IF
      type *,' '
      type *,'Searching for records, Please stand by'
      CALL DTR$COMMAND (DAB, 'FIND !CMD WITH US_EQV = !CMD AND
     1 COUNTRY NOT = "17";',DOMAIN,DSGKEY)
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
           type *,'There was a Fatal Datatrieve ERROR'
           type *,'Hit RETURN to restart'
           accept 60035,answer
60035      format (A)
           RETURN
      END IF

C Investigate the number of records found,
C if no records were found then return to try another Rse

      CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')
      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF
      IF (NUM_RECS .EQ. 0) THEN
          type *,'No foreign alloys found similar to that designation'
          type *,'Hit RETURN to try another designation'
          accept 60045,answer
60045     format (A)
          GO TO 60000
      END IF

C Program will branch here only if RSE has been successful,
60100 TYPE 60105
60105 FORMAT (/' Select one of the following options: then hit RETURN'
     1        /' 1 = Print only standard fields'
     2       //' 2 = Print all fields'
     3       //' P = Do not print, just return to the previous menu'/)

     ACCEPT 60115,CHOICE
60115 FORMAT (A)

      IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           GO TO 400
```

```
        END IF
        IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
             type *,'Wrong entry, hit RETURN to try again'
             accept 60125,answer
 60125       format (A)
             go to 60100
        END IF

        CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')

 C Check for possible datatrieve errors
        IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
      1     (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
             CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
             type *,'There was a Fatal Datatrieve ERROR'
             type *,'Hit RETURN to restart'
             accept 60135,answer
 60135       format (A)
             RETURN
        END IF

 C Choose between screen display and printed report

 60140 TYPE 60145
 60145 FORMAT (/' Do you want to display the report on the screen'
      1        /' or print it to a temporary file for later use?'
      2       //' Please respond with S or F:'/)
       accept 60155,answer
 60155 format (A)
 C Input Error-Trap
             IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
      1          ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
             type *,'Wrong entry, please hit RETURN and try again'
                  accept 60165,answer
 60165            format (A)
                  GO TO 60140
             END IF
       IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
             WRITE (*,10715)
             ACCEPT 60175,FILENAME
 60175       FORMAT (A)
             PRT = 3
             OPEN (3,FILE=FILENAME,STATUS='NEW')
       ELSE
             PRT = 5
       END IF

       CALL LIB$ERASE_PAGE (1,1)
       IF ((ANSWER .EQ. 'S') .OR. (ANSWER .EQ. 's')) THEN
            CALL LIB$SPAWN ('SET TERM/WIDTH=132')
      END IF
      IPAGE  = 0
      RECPRT = 50

 60200 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
           type *,' '
           type *,'The Report has been printed'
           type *,'Hit the RETURN key to continue'
           accept 60205,answer
 60205     format (A)
           CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
```

```
            CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
            CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
          OPENPORT = .FALSE.
          END IF
          CALL LIB$SPAWN   ('SET TERM/WIDTH=80')
          RETURN
        END IF

60210   CALL DTR$GET_PORT (DAB, %REF(FULLREC))

        IF (CHOICE .EQ. '2') THEN
            go to 60300
        end if

C This section for abbreviated report

        RECPRT = RECPRT+1
        IF (RECPRT .GT. 50) THEN
            RECPRT = 1
            IPAGE  = IPAGE+1
            CALL LIB$ERASE_PAGE (1,1)
            WRITE (PRT,60215)  IPAGE,dsgkey
60215   FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
     1         '           Page ',I4,
     2        /' List of all foreign Alloys that are similar'
     3         ' to the U.S. alloy designation ',A30
     4       //' Rec. No. Designation',20X,'US_Equivalent',12X,
     5         ' Temper',10X,'Country',9X,'Form'/)
        END IF

        WRITE (PRT,60225) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
60225   FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
        GO TO 60200

C This section for full report
60300   CALL LIB$ERASE_PAGE (1,1)
        IPAGE = IPAGE+1
        WRITE (PRT,60305)  IPAGE,desg
60305   FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
     1         '                      Page ',I4,
     2        /' List of all foreign Alloys that are similar'
     3         ' to the U.S. alloy designation ',A30//)
        WRITE (PRT,60315) FADB,DESG,EQUIV,XCNTRY
60315   FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1          'Country: ',A15/)
        WRITE (PRT,60325) ALTYP,TEMPR,FORMNUM,ORIGIN
60325   FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
     1          ' Orig. Org: ',A10/)
        WRITE (PRT,60335)
60335   FORMAT (' COMPOSITION:')
        WRITE (PRT,60345)
60345   FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1      'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2      'Pb',6X,'Sn')
        WRITE (PRT,60355) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1       MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
60355   FORMAT (8X,'MIN: ',14(A7,1X))
        WRITE (PRT,60365) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1       MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
60365   FORMAT (8X,'MAX: ',14(A7,1X)/)
        WRITE (PRT,60375) OTHER1,OTHER2,SPECS1
60375   FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
```

```
      WRITE (PRT,60385) MINO1,MINO2,SPECS2
60385 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,60395) MAXO1,MAXO2,SPECS2
60395 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,60405) SPECS4
60405 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,60415) SPECS5
60415 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,60425) SCCRTG
60425 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,60435) MINYLD,MAXYLD,TYPYLD,YLUNIT
60435 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,60445) MINTNS,MAXTNS,TYPTNS,TNUNIT
60445 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 60460
      END IF

      type *,' '
      write (prt,60447)
60447 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)
      accept 60450,answer
60450 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if

60460 GO TO 60200


C ************************ * * * * * ******************************
C                                                                   *
C Option 7                                                          *
C For a range of foreign alloys, find all similar U.S. alloys       *
C                                                                   *
C This is Search 1.c in the specifications                          *
C                                                                   *
C The following logic is used:                                      *
C 1. Accept the first and last of the foreign Designations          *
C    as input                                                       *
C                                                                   *
C Just search for all U.S. alloys with their designation            *
C lying within the specified range, then print                      *
C                                                                   *
C ************************ * * * * * ******************************


70000 CALL LIB$ERASE_PAGE (1,1)
70010 TYPE 70015
70015 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Producing Standard Reports: [7]'
     2        /' -------------------------'
     3       //' For a range of foreign alloys, find all similar',
     4         ' U.S. alloys'
     5       //' Enter the range of foreign designations,'
     6         ' within double quotation marks'
     7       //' Or, To return to the previous Menu, enter P'
     8        /' Then hit the RETURN key'/)
```

```
      TYPE *,'Enter the lower value of the foreign designation range'

      ACCEPT 70025,DSGKEY1
70025 FORMAT (A)
      IF ((DSGKEY1(1:1) .EQ. 'P') .OR. (DSGKEY1(1:1) .EQ. 'p')) THEN
           GO TO 400
      END IF
      type *,' '
      TYPE *,'Enter the upper value of the foreign designation range'
      ACCEPT 70027,DSGKEY2
70027 FORMAT (A)
      type *,' '
      type *,'Searching for records, Please stand by'

      CALL DTR$COMMAND (DAB, 'FIND !CMD WITH COUNTRY = "17" AND
     1 DESIG BT !CMD AND !CMD;',DOMAIN,DSGKEY1,DSGKEY2)
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors
      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
           type *,'There was a Fatal Datatrieve ERROR'
           type *,'Hit RETURN to restart'
           accept 70035,answer
70035      format (A)
           RETURN
      END IF

C Investigate the number of records found,
C if no records were found then return to try another Rse

      CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')
      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF
      IF (NUM_RECS .EQ. 0) THEN
          type *,'No U.S. alloys found in that designation range'
          type *,'Hit RETURN to try another range'
          accept 70045,answer
70045     format (A)
          GO TO 70000
      END IF

C Program will branch here only if RSE has been successful,

70100 TYPE 70105
70105 FORMAT (/' Select one of the following options: then hit RETURN'
     1        /' 1 = Print only standard fields'
     2       //' 2 = Print all fields'
     3       //' P = Do not print, just return to the previous menu'/)

      ACCEPT 70115,CHOICE
70115 FORMAT (A)
      IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
           GO TO 400
      END IF
      IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
           type *,'Wrong entry, hit RETURN to try again'
           accept 70125,answer
70125      format (A)
```

```
          go to 70100
        END IF

        CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')

C Check for possible datatrieve errors

        IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
       1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
            CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
            type *,'There was a Fatal Datatrieve ERROR'
            type *,'Hit RETURN to restart'
            accept 70135,answer
70135       format (A)
            RETURN
        END IF

C Choose between screen display and printed report
70140 TYPE 70145
70145 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)
       accept 70155,answer
70155 format (A)

C Input Error-Trap
            IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
       1        ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 70165,answer
70165            format (A)
                 GO TO 70140
            END IF

        IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
             WRITE (*,10715)
             ACCEPT 70175,FILENAME
70175        FORMAT (A)
             PRT = 3
             OPEN (3,FILE=FILENAME,STATUS='NEW')
        ELSE
             PRT = 5
        END IF
        CALL LIB$ERASE_PAGE (1,1)
        IF ((ANSWER .EQ. 'S') .OR. (ANSWER .EQ. 's')) THEN
             CALL LIB$SPAWN ('SET TERM/WIDTH=132')
        END IF
        IPAGE  = 0
        RECPRT = 50
70200   IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
            type *,' '
            type *,'The Report has been printed'
            type *,'Hit the RETURN key to continue'
            accept 70205,answer
70205       format (A)
            CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
            IF (OPENPORT) THEN
              CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
              CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
            OPENPORT = .FALSE.
            END IF
            CALL LIB$SPAWN    ('SET TERM/WIDTH=80')
```

```
          RETURN
        END IF


70210 CALL DTR$GET_PORT (DAB, %REF(FULLREC))

C Extract the country code from the fortran buffer to be used to
C   pull out the country literal from the literal pool
        CNTRY = CONTRY
        CALL jsCNTRY (cntry,icntry,Xcntry)
        IF (CHOICE .EQ. '2') THEN
            go to 70300
        end if

C This section for abbreviated report
        RECPRT = RECPRT+1
        IF (RECPRT .GT. 50) THEN
            RECPRT = 1
            IPAGE  = IPAGE+1
            CALL LIB$ERASE_PAGE (1,1)
            WRITE (PRT,70215)  IPAGE,dsgkey1,dsgkey2
70215 FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
     1         '          Page ',I4,
     2        /' List of all U.S. Alloys that are similar'
     3         ' to the foreign alloys in the range'
     4        /' ',A30,' =< designation >= ',A30,
     5       //' Rec. No. Designation',20X,'US_Equivalent',12X,
     6         ' Temper',10X,'Country',9X,'Form'/)
        END IF

        WRITE (PRT,70225) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
70225 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
        GO TO 70200

C This section for full report
70300 CALL LIB$ERASE_PAGE (1,1)
        IPAGE = IPAGE+1
        WRITE (PRT,70305)  IPAGE,dsgkey1,dsgkey2
70305 FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
     1           '                    Page ',I4,
     2          /' List of all U.S. Alloys that are similar'
     3           ' to the foreign alloys in the range',
     4          /' ',A30,' =< designation >= ',A30//)

        WRITE (PRT,70315) FADB,DESG,EQUIV,XCNTRY
70315 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1        'Country: ',A15/)
        WRITE (PRT,70325) ALTYP,TEMPR,FORMNUM,ORIGIN
70325 FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
     1        ' Orig. Org: ',A10/)
        WRITE (PRT,70335)
70335 FORMAT (' COMPOSITION:')
        WRITE (PRT,70345)
70345 FORMAT (' [Wt.%]',6X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1     'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2     'Pb',6X,'Sn')
        WRITE (PRT,70355) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1      MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
70355 FORMAT (8X,'MIN: ',14(A7,1X))
        WRITE (PRT,70365) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1      MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
70365 FORMAT (8X,'MAX: ',14(A7,1X)//)
```

```
      WRITE (PRT,70375) OTHER1,OTHER2,SPECS1
70375 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
      WRITE (PRT,70385) MINO1,MINO2,SPECS2
70385 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,70395) MAXO1,MAXO2,SPECS2
70395 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,70405) SPECS4
70405 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,70415) SPECS5
70415 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,70425) SCCRTG
70425 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,70435) MINYLD,MAXYLD,TYPYLD,YLUNIT
70435 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,70445) MINTNS,MAXTNS,TYPTNS,TNUNIT
70445 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)

      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 70460
      END IF
      type *,' '
      write (prt,70447)
70447 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)
      accept 70450,answer
70450 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if

70460 GO TO 70200

C ************************ * * * * * ******************************
C                                                                  *
C Option 8                                                         *
C For a range of foreign alloys, find all similar foreign alloys*
C                                                                  *
C This is Search 1.d in the specifications                         *
C NOTE;  WE ARE HOLDING THIS SEARCH UNTIL FURTHER CLEARANCE        *
C        IT SEEMS TO BE AMBIGUOUS OR REDUNDANT                     *
C                                                                  *
C ************************ * * * * * ******************************


C ************************ * * * * * ******************************
C                                                                  *
C Option 9                                          *
C For a range of U.S. alloys, find all similar foreign alloys      *
C                                                                  *
C This is Search 1.h in the specifications                    *
C                                                   *
C The following logic is used:                                *
C 1. Accept the first and last of the U.S. Designations     as input*
C                                                   *
C Just search for all foreign alloys with their U.S. equivalent    *
C lying within the specified range of designations, then print     *
C                                                                 *
C ************************ * * * * * ******************************


79000 CALL LIB$ERASE_PAGE (1,1)
```

```
 79010 TYPE 79015
 79015 FORMAT (/T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
      1       //' Producing Standard Reports: [9]'
      2        /' -------------------------'
      3       //' For a range of U.S. alloys, find all similar'
      4         ' foreign alloys'
      5       //' Enter the range of U.S. designations, within'
      6         ' double quotation marks'
      7       //' Or, To return to the previous Menu, enter P'
      8        /' Then hit the RETURN key'/)


       TYPE *,'Enter the lower value of the U.S. designation range'

       ACCEPT 79025,DSGKEY1
 79025 FORMAT (A)

       IF ((DSGKEY1(1:1) .EQ. 'P') .OR. (DSGKEY1(1:1) .EQ. 'p')) THEN
            GO TO 400
       END IF

       type *,' '
       TYPE *,'Enter the upper value of the foreign designation range'

       ACCEPT 79027,DSGKEY2
 79027 FORMAT (A)


       type *,' '
       type *,'Searching for records, Please stand by'


       CALL DTR$COMMAND (DAB, 'FIND !CMD WITH COUNTRY NOT = "17" AND
      1 DESIG BT !CMD AND !CMD;',DOMAIN,DSGKEY1,DSGKEY2)

       CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors


       IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
      1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
            CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
            type *,'There was a Fatal Datatrieve ERROR'
            type *,'Hit RETURN to restart'
            accept 79035,answer
 79035      format (A)
            RETURN
       END IF

C Investigate the number of records found,
C if no records were found then return to try another Rse
       CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')
       IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
           CALL DTR$GET_PORT (DAB, NUM_RECS)
           CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
       END IF
       IF (NUM_RECS .EQ. 0) THEN
           type *,'No foreign alloys found in that designation range'
           type *,'Hit RETURN to try another range'
           accept 79045,answer
 79045     format (A)
```

```
          GO TO 79000
        END IF

C Program will branch here only if RSE has been successful,
79100 TYPE 79105
79105 FORMAT (/' Select one of the following options: then hit RETURN'
     1         /' 1 = Print only standard fields'
     2        //' 2 = Print all fields'
     3        //' P = Do not print, just return to the previous menu'/)

        ACCEPT 79115,CHOICE
79115 FORMAT (A)
        IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
            GO TO 400
        END IF
        IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
            type *,'Wrong entry, hit RETURN to try again'
            accept 79125,answer
79125       format (A)
            go to 79100
        END IF

        CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')

C Check for possible datatrieve errors
        IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1     (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
            CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
            type *,'There was a Fatal Datatrieve ERROR'
            type *,'Hit RETURN to restart'
            accept 79135,answer
79135       format (A)
            RETURN
        END IF

C Choose between screen display and printed report
79140 TYPE 79145
79145 FORMAT (/' Do you want to display the report on the screen'
     1         /' or print it to a temporary file for later use?'
     2        //' Please respond with S or F:'/)

        accept 79155,answer
79155 format (A)

C Input Error-Trap

             IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1           ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
             type *,'Wrong entry, please hit RETURN and try again'
                  accept 79165,answer
79165             format (A)
                  GO TO 79140
             END IF

        IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
             WRITE (*,10715)

             ACCEPT 79175,FILENAME
79175        FORMAT (A)
             PRT = 3
             OPEN (3,FILE=FILENAME,STATUS='NEW')
        ELSE
```

```
            PRT = 5
        END IF

        CALL LIB$ERASE_PAGE (1,1)
        IF ((ANSWER .EQ. 'S') .OR. (ANSWER .EQ. 's')) THEN
            CALL LIB$SPAWN ('SET TERM/WIDTH=132')
        END IF
        IPAGE  = 0
        RECPRT = 50
79200   IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
            type *,' '
            type *,'The Report has been printed'
            type *,'Hit the RETURN key to continue'
            accept 79205,answer
79205       format (A)
            CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
            IF (OPENPORT) THEN
              CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
              CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
              OPENPORT = .FALSE.
            END IF
            CALL LIB$SPAWN    ('SET TERM/WIDTH=80')
            RETURN
        END IF

79210   CALL DTR$GET_PORT (DAB, %REF(FULLREC))

C Extract the country code from the fortran buffer to be used to
C   pull out the country literal from the literal pool
        CNTRY = CONTRY
        CALL jsCNTRY (cntry,icntry,Xcntry)

        IF (CHOICE .EQ. '2') THEN
            go to 79300
        end if

C This section for abbreviated report

        RECPRT = RECPRT+1
        IF (RECPRT .GT. 50) THEN
            RECPRT = 1
            IPAGE  = IPAGE+1
            CALL LIB$ERASE_PAGE (1,1)
            WRITE (PRT,79215)  IPAGE,dsgkey1,dsgkey2
79215   FORMAT ('1',' NASA ALLOY DATABASE ABBREVIATED REPORT',
     1              '              Page ',I4,
     2             /' List of all foreign Alloys that are similar'
     3              ' to the U.S. alloys in the range',
     4             /' ',A30,' =< designation >= ',A30,
     5            //' Rec. No. Designation',20X,'US_Equivalent',12X,
     6              ' Temper',10X,'Country',9X,'Form'/)
        END IF


        WRITE (PRT,79225) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
79225   FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
        GO TO 79200


C This section for full report

79300   CALL LIB$ERASE_PAGE (1,1)
```

```
      IPAGE = IPAGE+1
      WRITE (PRT,79305)  IPAGE,dsgkey1,dsgkey2
79305 FORMAT ('1NASA ALLOY DATABASE FULL REPORT',
     1            '                          Page ',I4,
     2           /' List of all U.S. Alloys that are similar'
     3            ' to the foreign alloys in the range',
     4           /' ',A30,' =< designation >= ',A30//)

      WRITE (PRT,79315) FADB,DESG,EQUIV,XCNTRY
79315 FORMAT (' Rec#: ',A7,' Designation: ',A30,' US_Equivalent: ',A25,
     1        'Country: ',A15/)

      WRITE (PRT,79325) ALTYP,TEMPR,FORMNUM,ORIGIN
79325 FORMAT (' Type: ',A4,'    Temper: ',A15,'  Form: ',A30,
     1        ' Orig. Org: ',A10/)

      WRITE (PRT,79335)
79335 FORMAT (' COMPOSITION:')

      WRITE (PRT,79345)
79345 FORMAT (' [Wt.%]',7X,'Al',6X,'Si',6X,'Fe',6X,'Cu',6X,'Mn',6X,
     1     'Mg',6X,'Zn',6X,'V ',6X,'Ti',6X,'Zr',6X,'Cr',6X,'Ni',6X,
     2     'Pb',6X,'Sn')

      WRITE (PRT,79355) MINAL,MINSI,MINFE,MINCU,MINMN,MINMG,MINZN,
     1      MINV,MINTI,MINZR,MINCR,MINNI,MINPB,MINSN
79355 FORMAT (8X,'MIN: ',14(A7,1X))
      WRITE (PRT,79365) MAXAL,MAXSI,MAXFE,MAXCU,MAXMN,MAXMG,MAXZN,
     1      MAXV,MAXTI,MAXZR,MAXCR,MAXNI,MAXPB,MAXSN
79365 FORMAT (8X,'MAX: ',14(A7,1X)/)
      WRITE (PRT,79375) OTHER1,OTHER2,SPECS1
79375 FORMAT (13X,A10,2X,A10,42X,'Specifications:[1] ',A30)
      WRITE (PRT,79385) MINO1,MINO2,SPECS2
79385 FORMAT (8X,'MIN: ',A7,6X,A7,61X,'[2] ',A30)
      WRITE (PRT,79395) MAXO1,MAXO2,SPECS2
79395 FORMAT (8X,'MAX: ',A7,6X,A7,61X,'[3] ',A30)
      WRITE (PRT,79405) SPECS4
79405 FORMAT (92X,'[4] ',A30)
      WRITE (PRT,79415) SPECS5
79415 FORMAT (92X,'[5] ',A30)
      WRITE (PRT,79425) SCCRTG
79425 FORMAT (19X,'MIN  MAX  TYP  UNITS',11X,'SCC Rating: ',A4)
      WRITE (PRT,79435) MINYLD,MAXYLD,TYPYLD,YLUNIT
79435 FORMAT (' Yield   Strength: ',3(A3,2X),A6)
      WRITE (PRT,79445) MINTNS,MAXTNS,TYPTNS,TNUNIT
79445 FORMAT (' Tensile Strength: ',3(A3,2X),A6,10X,'NOTES: ',A60)
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
          GO TO 79460
      END IF

      type *,' '
      write (prt,79447)
79447 format (' Hit the RETURN key to continue printing, To stop',
     1        ' printing type S, then Hit the RETURN key: ',$)

      accept 79450,answer
79450 format (A)
      if ((answer .eq. 'S') .OR. (answer .eq. 's')) then
           call lib$spawn ('set term/width=80')
           return
      end if
79460 GO TO 79200
```

```
C ********************** * * * * * ******************************
C                                                                 *
C This section for building your own reports                      *
C                                                                 *
C ********************** * * * * * ******************************

C Build an array with the output from a "SHOW FIELDS" Command
C When the user chooses a field name this array will be searched

80000 CALL LIB$ERASE_PAGE (1,1)
81000 TYPE 81005
81005 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Building your own reports:'
     2        /' ------------------------'
     3       //' Do you wish to see the field names?'
     4        /' Please respond with Y or N'
     5       //' When the display begins you may hold the screen'
     6         ' at any point'
     7        /' Type CTRL-S to hold screen, and CTRL-Q to continue'/)

81100 ACCEPT 81105,ANSWER
81105 FORMAT (A)

C Input Error-Trap

           IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1         ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
           type *,'Wrong entry, please hit RETURN and try again'
                accept 81205,answer
81205           format (A)
                GO TO 80000
           END IF


      IF ((ANSWER .EQ. 'N') .OR. (ANSWER .EQ. 'n')) THEN
           GO TO 82810
      END IF

      type 82205
82205 format (/' Use abbreviated field names that are shown'
     1         ' in parenthesis,'
     2        /' where no parenthesized name is shown, use'
     3         ' the field name shown'/)

      type 82305
82305 format (' Hit the RETURN key to show the fields names'/)

      accept 82315,answer
82315 format (A)

82400 CALL DTR$COMMAND (DAB,'SHOW FIELDS !CMD;', DOMAIN)
      I = 0
      DO WHILE ((DAB$W_STATE .EQ. DTR$K_STL-MSG) .AND.
     1          (DAB$L_CONDITION .EQ. %LOC(DTR$_SHOWTEXT)))
         I = I+1
         SHOWFLDS(I) = MSG_BUFF(1:DAB$W_MSG_LEN)
         CALL DTR$CONTINUE (DAB)
      END DO

      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors
```

```
      IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
          GO TO 90000
      END IF

82500 TYPE 82505
82505 FORMAT (' Do you wish to see the field names again?'
     1       /' Please respond with Y or N'
     2      //' Remember, when the display begins'
     3        ' you may hold the screen at any point'/
     4       /' Type CTRL-S to hold screen, and CTRL-Q to continue'/)

82600 ACCEPT 82605,ANSWER
82605 FORMAT (A)

C Input Error-Trap
            IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 82705,answer
82705            format (A)
                 GO TO 82500
            END IF
      IF ((ANSWER .EQ. 'Y') .OR. (ANSWER .EQ. 'y')) THEN
          GO TO 82400
      END IF

C Erase the Screen and Prompt the user for an RSE - Record
C Selection Expression (Pass this expression to Datatrieve)
82810 CALL LIB$ERASE_PAGE (1,1)
82820 TYPE 82825
82825 FORMAT (T21,' NASA ALLOY DATABASE MANAGEMENT SYSTEM'
     1       //' Building your own reports:'
     2        /' ------------------------'
     3       //' Please type a record selection expression in the'
     4         ' following form, and hit RETURN:'
     5       //' Field-Name1 = Value1 [ AND Field_Name2 = Value2 ..]'
     6       //' You may use any of the following operators -'
     7        /'  > or GT    < or LT    = or EQUAL    BT or Between '
     8        /' starting with   '
     9         ' CONT or containing'
     A       //' Example 1: DESIG = 1050'
     B        /' Example 2: DESIG = "A9105" AND COUNTRY = 17'
     C        /' Example 3: DESIG = "A9105" AND MIN-SI GT 0.05'
     D        /' Example 4: DESIG = 2210  AND MAX-TNS LT 123'
     E        /' Example 5: fadb-no = 123'
     F        /' Example 6: DESIG CONTAINING abcd ...'
     G        /' Example 7: DESIG starting with "A19" ...'
     H       //' To return to the Main Menu, just type R,'
     I        /' To read HELP information, just type H,'
     J         ' Then Hit the RETURN Key')

82830 READ 82835, EXPRLINE
82835 FORMAT (A)

      IF ((EXPRLINE .EQ. 'R') .OR. (EXPRLINE .EQ. 'r')) THEN
          RETURN
      END IF
      IF ((EXPRLINE .EQ. 'h') .OR. (EXPRLINE .EQ. 'h')) THEN
           CALL LIB$ERASE_PAGE (1,1)
           CALL LIB$SPAWN ('bldownhelp')
            GO TO 82810
      END IF
```

```
      type *,'Searching for record - please stand by'

82840 CALL DTR$COMMAND (DAB, 'FIND !CMD WITH !CMD;',
     1 DOMAIN,EXPRLINE)

      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          GO TO 90100
      END IF


C Investigate the number of records found,
C if no records were found then return to try another Rse

82900 CALL DTR$COMMAND (DAB, 'STORE PT1 USING NUM = COUNT;')

      IF (DAB$W_STATE .EQ. DTR$K_STL_PGET) THEN
          CALL DTR$GET_PORT (DAB, NUM_RECS)
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
      END IF

      IF (NUM_RECS .EQ. 0) THEN
          type *,'Please hit RETURN to try another selection'
          accept 82905,answer
82905     format (A)
          GO TO 82810
      END IF

C Program will branch here only if RSE has been successful,

83000 TYPE 83005
83005 FORMAT (/' Select one of the following options:'
     1       //' 1 = Print all fields'
     2        /' 2 = Print only selected fields'
     3       //' Note: '
     4        /'      Under option 2, - It is best to list'
     5         ' only a few fields at one time;'
     6       //'      As the total size of all selected fields'
     7         ' approaches 130 characters'
     8        /'      The report line will wrap around, making it'
     9         ' difficult to read'/)

83100 ACCEPT 83105,CHOICE
83105 FORMAT (A)

C Input Error-Trap

            IF ((CHOICE .NE. '1') .AND. (CHOICE .NE. '2')) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 83205,answer
83205            format (A)
                 GO TO 83000
            END IF

      IF (CHOICE .EQ. '1') THEN
          GO TO 84400
          END IF
```

```
C Print only selected fields:
C Prompt user for fields to show on report

83300     TYPE 83305
83305     FORMAT (/' Type all the fields you wish to show on the report'
     1           /' in the order in which you wish to list them;'
     2           //' Seperate the fields by commas'/)


83400     ACCEPT 83405,PRTFLDS
83405     FORMAT (A)

C Choose between screen display and printed report

83500 TYPE 83505
83505 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)

83600 ACCEPT 83605,ANSWER
83605 FORMAT (A)

C Input Error-Trap

           IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1         ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
           type *,'Wrong entry, please hit RETURN and try again'
                accept 83705,answer
83705           format (A)
                GO TO 83500
           END IF


      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
           GO TO 84200
      END IF

C This section for screen display only

83800     CALL DTR$COMMAND (DAB, 'FOR CURRENT PRINT !CMD;',
     1                           PRTFLDS)

      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
           GO TO 90100
      END IF

83900 TYPE 83905
83905 FORMAT (' Do you now wish to print the same report?'
     1       //' Please respond with Y or N:'
     2        /' Then hit the RETURN key'/)

84000 ACCEPT 84005,ANSWER
84005 FORMAT (A)

C Input Error-Trap

           IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
```

```
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
           type *,'Wrong entry, please hit RETURN and try again'
                accept 84105,answer
84105           format (A)
                GO TO 83900
           END IF


      IF ((ANSWER .EQ. 'N') .OR. (ANSWER .EQ. 'n')) THEN
         RETURN
      END IF

C This section for Printed Report
C Pass the earlier RSE on to Datatrieve via DTR$COMMAND

84200 CALL DTR$COMMAND (DAB, 'REPORT CURRENT ON NASAREP.TXT;')

C Check for possible datatrieve errors

      IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
          GO TO 90000
      END IF

C Prompt user for a Report Title

84210 TYPE 84215
84215 FORMAT (//' Enter the report title enclosed in quotation marks'
     1         /' Separate lines with a slash "/"'
     2        //' Example :"LIST OF FOREIGN ALLOYS WITH"/"DESIGNATION'
     3          ' = 1090"'//)

84220 READ 84225, LGTH, REPHEADER
84225 FORMAT (Q, A)


C Now Set the Report Heading Based on the entry

84230 IF (LGTH .NE. 0) THEN
          CALL DTR$COMMAND (DAB, 'SET REPORT_NAME = !CMD;', REPHEADER)
      END IF

C Check for Datatrieve Errors Again
      IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
          GO TO 90000
      END IF

C Set additional Print Parameters

84240 CALL DTR$COMMAND (DAB, 'SET COLUMNS_PAGE = 132')

C Check for possible datatrieve errors

      IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
          GO TO 90000
      END IF

C Pass the earlier print list to Datatrieve
84250 CALL DTR$COMMAND (DAB, 'PRINT !CMD;', PRTFLDS)

C Check for possible datatrieve errors
      IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
          GO TO 90000
```

```
      END IF

84260 CALL DTR$COMMAND (DAB, 'END_REPORT;')
      CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
C Check for possible datatrieve errors
      IF (DAB$W_STATE .EQ. DTR$K_STL_MSG) THEN
          GO TO 90000
      END IF
84270 type 84275
84275 format (/' The report has been sucessfully generated'
     1        /' into your directory as NASAREP.TXT'
     2       //' You may browse it with EDT editor or'
     3        /' print it with "PRINT NASAREP.TXT"'
     4       //' Hit the RETURN key to continue'/)

84280 accept 84285,answer
84285 format (A)
      GO TO 82810


C*********************************************************
C
C Print All Fields - under Building your own reports
C
C*********************************************************

C Choose between screen display and printed report

84400 TYPE 84405
84405 FORMAT (/' Do you want to display the report on the screen'
     1        /' or print it to a temporary file for later use?'
     2       //' Please respond with S or F:'/)


84410 accept 84415,answer
84415 format (A)

C Input Error-Trap

            IF (((ANSWER .NE. 'S') .AND. (ANSWER .NE. 's')) .AND.
     1          ((ANSWER .NE. 'F') .AND. (ANSWER .NE. 'f'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 84425,answer
84425            format (A)
                 GO TO 84400
            END IF

      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
           WRITE (*,84435)
84435      FORMAT (/' Your report will be stored in a temporary'
     1              ' data file e.g. PRTTEMP.DAT which'
     2             /' you may brouse with EDT or print on your'
     3              ' local printer i.e. PRINT PRTTEMP.DAT'
     4             //' Choose and enter a name for your'
     5              ' temporary print file, e.g. PRTTEMP'/)

84440      ACCEPT 84445,FILENAME
84445      FORMAT (A)
           PRT = 3
           OPEN (3,FILE=FILENAME,STATUS='NEW')
      ELSE
```

```
            PRT = 5
        END IF

C Step 1:

84500 CALL DTR$COMMAND (DAB, 'PORT2 = CURRENT;')
      UNITY = 'Wt %'

C Check for possible datatrieve errors

      IF ((DAB$L_CONDITION .NE. %LOC(DTR$_SUCCESS)) .OR.
     1    (DAB$L_CONDITION .EQ. %LOC(DTR$_ERROR)))  THEN
          CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
          type *,'DTR ERROR'
          type *,'Just hit RTN to continue'
          accept 84515,answer
84515     format (A)
           RETURN
      END IF

C Step 2:

C The above command causes the DTR$K_STL_PGET stall point
C Whiles at this DTR stall point, we will continue to use
C DTR$GET_PORT to copy one record at a time from the port
C into our Fortran record buffer FULLREC


C IF NEXT CONDITION IS TRUE THEN RESET DOMAIN B4 RETURN

84520 IF (DAB$W_STATE .NE. DTR$K_STL_PGET) THEN
          type *,' '
          type *,'- no more records to print'
          type *,'Just hit RETURN to go back to the previous menu'
          accept 84535,answer
84535     format (A)
          CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
          IF (OPENPORT) THEN
             CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
             CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
           OPENPORT = .FALSE.
          END IF
          RETURN
      END IF

84540 CALL DTR$GET_PORT (DAB, %REF(FULLREC))

      CNTRY = CONTRY
      CALL jsCNTRY (cntry,icntry,Xcntry)

C     Print the detail line from the record buffer

84600 CALL LIB$ERASE_PAGE (1,1)
84610 WRITE (PRT,84615)
84615 FORMAT (T20,'NASA ALLOY DATABASE FULL REPORT'
     1       /T20,'------------------------------')
      WRITE (PRT,84625) UNS,FADB,DESG,TEMPR,FORMNUM,ALTYP
84625 FORMAT (/' UNS No. = ',A6,31X,'FADB Ref. No. = ',A7,
     1       //' Designation = ',A30,3X,'Temper = ',A15,
     2       //' Form = ',A30,2X,'Alloy type = ',A20)
      WRITE (PRT,84635) STDNUM,EQUIV,XCNTRY,ORIGIN
84635 FORMAT (///' Standard No. = ',A10,11X,
```

```
     1          ' U.S. Equivalent = ',A25,
     2        //' Country of Origin = ',A15,2X,
     3          ' Originating Organization = ',A10)
      WRITE (PRT,84645) SPECS1,SPECS2,SPECS3,SPECS4
84645 FORMAT (//' Specifications: '2A30/17X,2A30)
      WRITE (PRT,84655) UNITY,MINAL,MAXAL,MINSI,MAXSI,
     A           MINFE,MAXFE,MINCU,MAXCU,MINMN,MAXMN,
     B           MINMG,MAXMG,MINV,MAXV,MINTI,MAXTI,
     C           MINPB,MAXPB,MINSN,MAXSN,
     D           OTHER1,MINO1,MAXO1,OTHER2,MINO2,MAXO2
84655 FORMAT (/' Composition Data:',12X,'Units = ',A4
     1         /' ----------------'
     2        //' Element',12X,'Minimum   Maximum'
     3          '     Element',10X,'Minimum   Maximum'
     2         /' -------',12X,'-------   -------'
     3          '     -------',10X,'-------   -------'

     4        //' Al (Aluminum) ',6X,A7,6X,A7,3X,
     5          'Si (Silicon)   ',5X,A7,4X,A7
     6         /' Fe (Iron)     ',6X,A7,6X,A7,3X,
     7          'Cu (Copper)    ',5X,A7,4X,A7
     8         /' Mn (Manganese)',6X,A7,6X,A7,3X,
     9          'Mg (Magnesium) ',5X,A7,4X,A7
     A         /' Zn (Zinc)     ',6X,A7,6X,A7,3X,
     B          'V (Vanadium)  ',5X,A7,4X,A7
     C         /' Ti (Titanium) ',6X,A7,6X,A7,3X,
     D          'Zr (Zirconium) ',5X,A7,4X,A7
     E         /' Cr (Cromium)  ',6X,A7,6X,A7,3X,
     F          'Ni (Nickel)    ',5X,A7,4X,A7
     G         /' Pb (Lead)     ',6X,A7,6X,A7,3X,
     H          'Sn (Tin)       ',5X,A7,4X,A7

     I        //' ',A10,    '    ',6X,A7,6X,A7,3X,
     J          ' ',A10,    '    ',5X,A7,4X,A7/)

      WRITE (PRT,84665) MINYLD,MAXYLD,TYPYLD,YLUNIT,
     A     MINTNS,MAXTNS,TYPTNS,TNSUNIT,SCCRTG,
     B     REFR1,REFR2,REFR3,ALNOTES
84665 FORMAT (//' Property',11X,'Minimum   Maximum',4X,
     1          ' Typical   Units'
     2         /' --------',11X,'-------   -------',4X,
     3          ' -------   -----'
     4        //' Yield Strength      ',3(3X,A3,4X),A6
     5        //' Tensile Strength    ',3(3X,A3,4X),A6
     6        //' SCC Rating          ',3X,A4
     7        //' Data References     ',3(2X,A3,2X),
     8        //' Notes:  ',A60/)


C The remaining options are for Screen Reports only
C So if printing reports just go back from here
      IF ((ANSWER .EQ. 'F') .OR. (ANSWER .EQ. 'f')) THEN
           GO TO 84520
      END IF

84670 WRITE (*,84675)
84675 FORMAT (/' You may continue the report or return to'
     1          ' the Main Menu from this point'
     2         /' Select one of the following: '
     3          ' C = Continue;   M = Return to the Main Menu'/)
84680 ACCEPT 84685,ANSWER
84685 FORMAT (A)
```

```
C Input Error-Trap
            IF (((ANSWER .NE. 'M') .AND. (ANSWER .NE. 'm')) .AND.
     1          ((ANSWER .NE. 'C') .AND. (ANSWER .NE. 'c'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 84695,answer
84695            format (A)
                 GO TO 84670
            END IF
       IF ((ANSWER .EQ. 'M') .OR. (ANSWER .EQ. 'm')) THEN
            CALL DTR$COMMAND (DAB, 'RELEASE ALL;')
           IF (OPENPORT) THEN
              CALL DTR$COMMAND (DAB, 'FINISH PORT2;')
              CALL DTR$DTR (DAB,DTR$M_OPT_CMD)
            OPENPORT = .FALSE.
       END IF
            RETURN
       END IF

84700 TYPE 84705
84705 FORMAT (/' Do you wish to see the same record again?:'
     1         ' Respond with Y or N'
     2        /' When the display begins'
     3         ' you may hold the screen at any point'
     4        /' Type CTRL-S to hold screen, and CTRL-Q to continue'/)

      ACCEPT 84715,ANSWER
84715 FORMAT (A)
C Input Error-Trap
            IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 84725,answer
84725            format (A)
                 GO TO 84700
            END IF
       IF ((ANSWER .EQ. 'Y') .OR. (ANSWER .EQ. 'y')) THEN
            GO TO 84600
       ELSE
            GO TO 84520
       END IF

C Below is the general error message handling routine
C Call the Terminal Server to handle messages at the end of the report
90000 CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
C If there was any arror then promt user to retry again
       IF ((DAB$L_CONDITION .EQ. %LOC(DTR$_SUCCESS)) .AND.
     1    (DAB$L_CONDITION .NE. %LOC(DTR$_ERROR)))  THEN
            RETURN
       END IF
90100 TYPE 90105
90105 FORMAT (' There was a Datatrieve error,'
     1         ' Do you wish to try again?'
     2       //' Please respond with Y or N'/)
      accept 90205,answer
90205 format (A)
C Input Error-Trap
            IF (((ANSWER .NE. 'Y') .AND. (ANSWER .NE. 'y')) .AND.
     1          ((ANSWER .NE. 'N') .AND. (ANSWER .NE. 'n'))) THEN
            type *,'Wrong entry, please hit RETURN and try again'
                 accept 90305,answer
90305            format (A)
```

```
                GO TO 90100
              END IF

        IF ((ANSWER .EQ. 'Y') .OR. (ANSWER .EQ. 'y')) THEN
             GO TO 100
        END IF

 90999  RETURN
        END


C*************************************************************************
C                        SUBROUTINE PRINTFEW                             *
C                                                                        *
C Module Name:           SBPRINTFEW.FOR                                  *
C Date Written:          August XX, 1985                                 *
C Designer/Programmer: Joseph K. Amanfu, Fisk University, Nashville      *
C   Modified by J. Springer 6/9/86                                       *
C                           7/7/89                                       *
C*************************************************************************

        SUBROUTINE JSPRTFEW (prt)

        INTEGER        RECPRT
        INTEGER        IPAGE
        INTEGER        PRT
        CHARACTER*15   XCNTRY

C FULLREC is the space defined to receive the record from
C the Datatrieve buffer
        INCLUDE '[NASA3.JSEXREC]FULLREC5.INC'

        COMMON/DATAREC/FULLREC
        COMMON/COUNTRY/XCNTRY
        COMMON/KOUNTS/IPAGE,RECPRT
        RECPRT = RECPRT+1
        IF (RECPRT .GT. 50) THEN
            RECPRT = 1
            IPAGE  = IPAGE+1
            CALL LIB$ERASE_PAGE (1,1)
            WRITE (PRT,205)  IPAGE, Xcntry
  205 FORMAT ('1NASA ALLOY DATABASE ABBREVIATED REPORT',
     1          '              Page ',I4,
     2         /' List of Alloys from ',A15
     3         //' Rec. No. Designation',20X,'US_Equivalent',12X,
     4           ' Temper',10X,'Country',9X,'Form'/)
        END IF
        WRITE (PRT,1005) FADB,DESG,EQUIV,TEMPR,XCNTRY,FORMNUM
 1005 FORMAT (' ',A7,2X,A30,1X,A25,1X,A15,1X,A15,1X,A30)
        RETURN
        END

C       Declares FULLREC buffer and variable equivalences for database
C       subroutines using this method to transfer data between subroutines
C       or between Datatrieve and a subroutine.
C
C                  J. Springer - 6/11/86
C       Revised to match extended record with addition elements and altered
C       miscellaneous field length.
C
C       Revised to allow for increased tensile, yield, and SCC field widths.
C
```

```
C     Revised 9/25/87 to add US_TEMP_EQV, IRR_DESG, and READY fields.
C
C     Revised July 89 to match NASAREC5
C
      CHARACTER*1   FULLREC (836)
C        (Note that the size of fullrec is 3 characters larger
C         than the Datatrieve record size since the 8 character
C         date field is converted to an 11 character string when
C         output.)

      CHARACTER*30 DESG
      CHARACTER*7  FADB
      CHARACTER*6  UNS
      CHARACTER*30 FORMNUM
      CHARACTER*15 TEMPR
      CHARACTER*5  ALTYP
      character*20 ALCAT
      CHARACTER*10 STATUS
      CHARACTER*10 ORIGIN
      CHARACTER*2  CONTRY
      CHARACTER*20 EQUIV
      CHARACTER*15 EQVTEMP
      character*2  EQVRATE
      CHARACTER*6  IRRDESG
      character*20 ALTDESIG
      CHARACTER*20 SPECS1
      CHARACTER*20 SPECS2
      CHARACTER*20 SPECS3
      CHARACTER*20 SPECS4
      CHARACTER*20 SPECS5
      CHARACTER*1  UNITS
      CHARACTER*7  MINAL
      CHARACTER*7  MAXAL
      CHARACTER*7  MINSI
      CHARACTER*7  MAXSI
      CHARACTER*7  MINFE
      CHARACTER*7  MAXFE
      CHARACTER*7  MINCU
      CHARACTER*7  MAXCU
      CHARACTER*7  MINMN
      CHARACTER*7  MAXMN
      CHARACTER*7  MINMG
      CHARACTER*7  MAXMG
      CHARACTER*7  MINZN
      CHARACTER*7  MAXZN
      CHARACTER*7  MINV
      CHARACTER*7  MAXV
      CHARACTER*7  MINTI
      CHARACTER*7  MAXTI
      CHARACTER*7  MINZR
      CHARACTER*7  MAXZR
      CHARACTER*7  MINCR
      CHARACTER*7  MAXCR
      CHARACTER*7  MINNI
      CHARACTER*7  MAXNI
      CHARACTER*7  MINPB
      CHARACTER*7  MAXPB
      CHARACTER*7  MINSN
      CHARACTER*7  MAXSN
      CHARACTER*7  MINC
      CHARACTER*7  MAXC
      CHARACTER*7  MINCO
```

```
      CHARACTER*7  MAXCO
      CHARACTER*7  MINMO
      CHARACTER*7  MAXMO
      CHARACTER*7  MINW
      CHARACTER*7  MAXW
      CHARACTER*7  MINP
      CHARACTER*7  MAXP
      CHARACTER*7  MINS
      CHARACTER*7  MAXS
      CHARACTER*7  MINB
      CHARACTER*7  MAXB
      CHARACTER*7  MINBE
      CHARACTER*7  MAXBE
      CHARACTER*7  MINGA
      CHARACTER*7  MAXGA
      CHARACTER*10 OTHER1
      CHARACTER*7  MINO1
      CHARACTER*7  MAXO1
      CHARACTER*10 OTHER2
      CHARACTER*7  MINO2
      CHARACTER*7  MAXO2
      CHARACTER*10 OTHER3
      CHARACTER*7  MINO3
      CHARACTER*7  MAXO3
      CHARACTER*4  MINYLD
      CHARACTER*4  MAXYLD
      CHARACTER*4  TYPYLD
      CHARACTER*6  YLUNIT
      CHARACTER*4  MINTNS
      CHARACTER*4  MAXTNS
      CHARACTER*4  TYPTNS
      CHARACTER*6  TNUNIT
      character*2  ELONG
      character*6  ELTEST
      character*3  MINHARD
      character*3  MAXHARD
      character*3  TYPHARD
      character*4  HARDUNIT
      CHARACTER*5  SCCRTG
      CHARACTER*3  EQVREF
      CHARACTER*3  REFR1
      CHARACTER*3  REFR2
      CHARACTER*60 ALNOTES
      CHARACTER*1  READY
      character*11 UPDATE
      EQUIVALENCE (FULLREC(1),DESG)
      EQUIVALENCE (FULLREC(31),FADB)
      EQUIVALENCE (FULLREC(38),UNS)
      EQUIVALENCE (FULLREC(44),FORMNUM)
      EQUIVALENCE (FULLREC(74),TEMPR)
      EQUIVALENCE (FULLREC(89),ALTYP)
      equivalence (fullrec(94),ALCAT)
      EQUIVALENCE (FULLREC(114),STATUS)
      EQUIVALENCE (FULLREC(124),ORIGIN)
      EQUIVALENCE (FULLREC(134),CONTRY)
      EQUIVALENCE (FULLREC(136),EQUIV)
      EQUIVALENCE (FULLREC(156),EQVTEMP)
      EQUIVALENCE (FULLREC(171),EQVRATE)
      EQUIVALENCE (FULLREC(173),IRRDESG)
      EQUIVALENCE (FULLREC(179),ALTDESIG)
      EQUIVALENCE (FULLREC(199),SPECS1)
      EQUIVALENCE (FULLREC(219),SPECS2)
```

```
      EQUIVALENCE (FULLREC(239),SPECS3)
      EQUIVALENCE (FULLREC(259),SPECS4)
      EQUIVALENCE (FULLREC(279),SPECS5)
      EQUIVALENCE (FULLREC(299),UNITS)
      EQUIVALENCE (FULLREC(300),MINAL)
      EQUIVALENCE (FULLREC(307),MAXAL)
      EQUIVALENCE (FULLREC(314),MINSI)
      EQUIVALENCE (FULLREC(321),MAXSI)
      EQUIVALENCE (FULLREC(328),MINFE)
      EQUIVALENCE (FULLREC(335),MAXFE)
      EQUIVALENCE (FULLREC(342),MINCU)
      EQUIVALENCE (FULLREC(349),MAXCU)
      EQUIVALENCE (FULLREC(356),MINMN)
      EQUIVALENCE (FULLREC(363),MAXMN)
      EQUIVALENCE (FULLREC(370),MINMG)
      EQUIVALENCE (FULLREC(377),MAXMG)
      EQUIVALENCE (FULLREC(384),MINZN)
      EQUIVALENCE (FULLREC(391),MAXZN)
      EQUIVALENCE (FULLREC(398),MINV)
      EQUIVALENCE (FULLREC(405),MAXV)
      EQUIVALENCE (FULLREC(412),MINTI)
      EQUIVALENCE (FULLREC(419),MAXTI)
      EQUIVALENCE (FULLREC(426),MINZR)
      EQUIVALENCE (FULLREC(433),MAXZR)
      EQUIVALENCE (FULLREC(440),MINCR)
      EQUIVALENCE (FULLREC(447),MAXCR)
      EQUIVALENCE (FULLREC(454),MINNI)
      EQUIVALENCE (FULLREC(461),MAXNI)
      EQUIVALENCE (FULLREC(468),MINPB)
      EQUIVALENCE (FULLREC(475),MAXPB)
      EQUIVALENCE (FULLREC(482),MINSN)
      EQUIVALENCE (FULLREC(489),MAXSN)
      EQUIVALENCE (FULLREC(496),MINC)
      EQUIVALENCE (FULLREC(503),MAXC)
      EQUIVALENCE (FULLREC(510),MINCO)
      EQUIVALENCE (FULLREC(517),MAXCO)
      EQUIVALENCE (FULLREC(524),MINMO)
      EQUIVALENCE (FULLREC(531),MAXMO)
      EQUIVALENCE (FULLREC(538),MINW)
      EQUIVALENCE (FULLREC(545),MAXW)
      EQUIVALENCE (FULLREC(552),MINP)
      EQUIVALENCE (FULLREC(559),MAXP)
      EQUIVALENCE (FULLREC(566),MINS)
      EQUIVALENCE (FULLREC(573),MAXS)
      EQUIVALENCE (FULLREC(580),MINB)
      EQUIVALENCE (FULLREC(587),MAXB)
      EQUIVALENCE (FULLREC(594),MINBE)
      EQUIVALENCE (FULLREC(601),MAXBE)
      EQUIVALENCE (FULLREC(608),MINGA)
      EQUIVALENCE (FULLREC(615),MAXGA)
      EQUIVALENCE (FULLREC(622),OTHER1)
      EQUIVALENCE (FULLREC(632),MINO1)
      EQUIVALENCE (FULLREC(639),MAXO1)
      EQUIVALENCE (FULLREC(646),OTHER2)
      EQUIVALENCE (FULLREC(656),MINO2)
      EQUIVALENCE (FULLREC(663),MAXO2)
      EQUIVALENCE (FULLREC(670),OTHER3)
      EQUIVALENCE (FULLREC(680),MINO3)
      EQUIVALENCE (FULLREC(687),MAXO3)
      EQUIVALENCE (FULLREC(694),MINYLD)
      EQUIVALENCE (FULLREC(698),MAXYLD)
      EQUIVALENCE (FULLREC(702),TYPYLD)
```

```
      EQUIVALENCE (FULLREC(706),YLUNIT)
      EQUIVALENCE (FULLREC(712),MINTNS)
      EQUIVALENCE (FULLREC(716),MAXTNS)
      EQUIVALENCE (FULLREC(720),TYPTNS)
      EQUIVALENCE (FULLREC(724),TNUNIT)
      EQUIVALENCE (FULLREC(730),ELONG)
      EQUIVALENCE (FULLREC(732),ELTEST)
      EQUIVALENCE (FULLREC(738),MINHARD)
      EQUIVALENCE (FULLREC(741),MAXHARD)
      EQUIVALENCE (FULLREC(744),TYPHARD)
      EQUIVALENCE (FULLREC(747),HARDUNIT)
      EQUIVALENCE (FULLREC(751),SCCRTG)
      EQUIVALENCE (FULLREC(756),EQVREF)
      EQUIVALENCE (FULLREC(759),REFR1)
      EQUIVALENCE (FULLREC(762),REFR2)
      EQUIVALENCE (FULLREC(765),ALNOTES)
      EQUIVALENCE (FULLREC(825),READY)
      EQUIVALENCE (FULLREC(826),UPDATE)


C     Declaration of PORT2 using PIC X(7) for composition
C     limits and record structure for all character fields
C    in the NASADOM4 domain.  This new domain adds the
C    fields EQV_TEMP for equivalent US temper values,
C    IRRDESG for the AA Int. Registry Record numbers,
C    READY, a one character field to indicate whether
C    the record is valid for shipment to NASA, and
C    composition fields for Be and Ga.
C
C       JS - 9/30/87
C
C ========================================================
C
C    Changed temper fields to condition fields
C       JS - 5/4/88
C ========================================================
C
C     Revised to match NASAFILE_REC5, which includes
C     new fields for alloy category, alternate designation,
C     etc. JS - July 89
C
      CALL DTR$COMMAND (DAB, 'DECLARE PORT PORT2 USING ')
      CALL DTR$COMMAND (DAB, '01  TEMPREC. ')
      CALL DTR$COMMAND (DAB, '05  DESIG    PIC X(30). ')
      CALL DTR$COMMAND (DAB, '05  FADB_NO PIC X(7). ')
      CALL DTR$COMMAND (DAB, '05  UNS_NO  PIC X(6). ')
      CALL DTR$COMMAND (DAB, '05  FORM    PIC X(30). ')
      CALL DTR$COMMAND (DAB, '05  CONDITION PIC X(15). ')
      CALL DTR$COMMAND (DAB, '05  ALTYPE  PIC X(5). ')
      CALL DTR$COMMAND (DAB, '05  ALCAT   PIC X(20). ')
      CALL DTR$COMMAND (DAB, '05  STATUS  PIC X(10). ')
      CALL DTR$COMMAND (DAB, '05  ORIGORG PIC X(10). ')
      CALL DTR$COMMAND (DAB, '05  COUNTRY PIC X(2). ')
      CALL DTR$COMMAND (DAB, '05  US_EQV  PIC X(20). ')
      CALL DTR$COMMAND (DAB, '05  EQVCOND PIC X(15).')
      CALL DTR$COMMAND (DAB, '05  EQVRATE PIC X(2). ')
      CALL DTR$COMMAND (DAB, '05  IRR_DESIG  PIC X(6).')
      CALL DTR$COMMAND (DAB, '05  ALTDESIG PIC X(20). ')
      CALL DTR$COMMAND (DAB, '05  SPEC1   PIC X(20). ')
      CALL DTR$COMMAND (DAB, '05  SPEC2   PIC X(20). ')
      CALL DTR$COMMAND (DAB, '05  SPEC3   PIC X(20). ')
```

```
CALL DTR$COMMAND (DAB, '05  SPEC4    PIC X(20). ')
CALL DTR$COMMAND (DAB, '05  SPEC5    PIC X(20). ')
CALL DTR$COMMAND (DAB, '05  COMPO. ')
CALL DTR$COMMAND (DAB, '10  WT_ATNO PIC X(1). ')
CALL DTR$COMMAND (DAB, '10  MIN-AL   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-AL   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-SI   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-SI   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-FE   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-FE   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-CU   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-CU   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-MN   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-MN   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-MG   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-MG   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-ZN   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-ZN   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-V    PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-V    PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-TI   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-TI   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-ZR   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-ZR   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-CR   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-CR   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-NI   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-NI   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-PB   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-PB   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-SN   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-SN   PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-C  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-C  PIC x(7). ')

CALL DTR$COMMAND (DAB, '10  MIN-CO   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-CO   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-MO   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-MO   PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-W  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-W  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-P  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-P  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-S  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-S  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-B  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-B  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MIN-BE PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-BE PIC x(7). ')
```

```
CALL DTR$COMMAND (DAB, '10  MIN-GA  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-GA  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  OTHER1  PIC X(10). ')
CALL DTR$COMMAND (DAB, '10  MIN-O1  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-O1  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  OTHER2  PIC X(10). ')
CALL DTR$COMMAND (DAB, '10  MIN-O2  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-O2  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  OTHER3  PIC X(10). ')
CALL DTR$COMMAND (DAB, '10  MIN-O3  PIC x(7). ')
CALL DTR$COMMAND (DAB, '10  MAX-O3  PIC x(7). ')
CALL DTR$COMMAND (DAB, '05  MIN_YLD PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  MAX_YLD PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  TYP_YLD PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  YL_UNIT PIC X(6). ')
CALL DTR$COMMAND (DAB, '05  MIN_TNS PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  MAX_TNS PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  TYP_TNS PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  TN_UNIT PIC X(6). ')
CALL DTR$COMMAND (DAB, '05  ELONGATION PIC X(2). ')
CALL DTR$COMMAND (DAB, '05  ELTEST PIC X(6). ')
CALL DTR$COMMAND (DAB, '05  MIN_HARD PIC X(3). ')
CALL DTR$COMMAND (DAB, '05  MAX_HARD PIC X(3). ')
CALL DTR$COMMAND (DAB, '05  TYP_HARD PIC X(3). ')
CALL DTR$COMMAND (DAB, '05  HARD_UNIT PIC X(4). ')
CALL DTR$COMMAND (DAB, '05  SCC_RTG PIC X(5). ')
CALL DTR$COMMAND (DAB, '05  DATA_REFERENCES. ')
CALL DTR$COMMAND (DAB, '10  EQVREF  PIC X(3). ')
CALL DTR$COMMAND (DAB, '10  REF1    PIC X(3). ')
CALL DTR$COMMAND (DAB, '10  REF2    PIC X(3). ')
CALL DTR$COMMAND (DAB, '05  NOTES   PIC X(60). ')
CALL DTR$COMMAND (DAB, '05  READY_CODE   PIC X(1). ')
CALL DTR$COMMAND (DAB, '05  UPDATE PIC X(11).; ')
CALL DTR$DTR (DAB, DTR$M_OPT_CMD)
```

```
      SUBROUTINE SCRNENTR
      INCLUDE 'DATABUFF'
      INCLUDE 'IODRVCOM'
      INCLUDE 'CONTROLY'
      REAL*4 Z
      INTEGER*2 S,PAGE_NUM,J,BLINK,INVERSE,
     2      ERR_PAGE,ERR_I,LONG
      INTEGER*4 STAT,KZ
      CHARACTER*1 ENTER,CHARACTER,BAK,CR,ESC,BEL,DELIMIT,A
      CHARACTER*2 Keypad,Numeric
      CHARACTER*8 G,H,IX
      CHARACTER*11 SCRLBL
      CHARACTER*20 X,BLANKS

      DIMENSION G(10),IX(10)

!              Define Test Characters
      PARAMETER (ENTER=CHAR(255), ! Enter Key In Keypad Mode
     2        BAK=CHAR(8),      ! Backspace
     2        CR=CHAR(13),      ! Carriage Return
     2        ESC=CHAR(27),     ! Escape
     2        BEL=CHAR(7),      ! Bell
     2        DELIMIT=CHAR(127) ) ! Largest Allowable Character
!              Key Board Control
      PARAMETER (Keypad=CHAR(27)//CHAR(61),  !  Sets Keypad Mode
     2        Numeric=CHAR(27)//CHAR(62) ) ! Returns To Numeric

!              Display attributes
      PARAMETER (BLINK=4, !  Blink
     2        INVERSE=2)   ! Inverse Video

      DATA       SCRLBL /'Page x Of x'/

!              Formats For Numeric Editing
      DATA       G(1) /'(BNF1.0)'/,
     2     G(2) /'(BNF2.0)'/,
     2     G(3) /'(BNF3.0)'/,
     2     G(4) /'(BNF4.0)'/,
     2     G(5) /'(BNF5.0)'/,
     2     G(6) /'(BNF6.0)'/,
     2     G(7) /'(BNF7.0)'/,
     2     G(8) /'(BNF8.0)'/

      DATA IX(1) /'(BNI1)'/,
     2     IX(2) /'(BNI2)'/,
     2     IX(3) /'(BNI3)'/,
     2     IX(4) /'(BNI4)'/,
     2     IX(5) /'(BNI5)'/,
     2     IX(6) /'(BNI6)'/,
     2     IX(7) /'(BNI7)'/
```

```
      DATA BLANKS /'                          '/

      SCRLBL(11:11)=CHAR(MAX_PAGE+47)
      STAT=LIB$PUT_LINE(Keypad)    !  Put Terminal In Keypad Mode
!                 Unpack Input Buffer
      DO 100 I=1,MAX_PAGE
      DO 100 J=1,LIMITS(I,3)
      K=VAR(I,J)
100   LABEL(K)=FULLREC(START(K):START(K)+LEN(K)-1)

      ERR_PAGE=2
      ERR_I=0
      CHARACTER=' '
!               Display Header--Page One
      STAT=LIB$ERASE_PAGE(1,1)
      STAT=LIB$PUT_SCREEN(Flag,1,1,2)
      DO 120 I=LIMITS(1,1),LIMITS(1,2)
      IF (ATTRIB(I).EQ.INVERSE) THEN
          STAT=LIB$PUT_SCREEN(LABEL(I)(1:LEN(I)),ROW(I),COL(I),2)
      ELSE IF (ATTRIB(I).EQ.BLINK) THEN
          STAT=LIB$PUT_SCREEN(LABEL(I)(1:LEN(I)),ROW(I),COL(I),6)
      ELSE
          STAT=LIB$PUT_SCREEN(LABEL(I)(1:LEN(I)),ROW(I),COL(I),0)
      END IF
120   CONTINUE

!  Main Loop-Begin On Page 2 And Cycle Through Until ENTER Key
140   PAGE_NUM=ERR_PAGE
      DO 500 WHILE (CHARACTER.NE.ENTER)
150   STAT=LIB$ERASE_PAGE(HEADLINE,1)

!               Display Prompts And Labels, This Page

      SCRLBL(6:6)=CHAR(PAGE_NUM+47)
      STAT=LIB$PUT_SCREEN(SCRLBL,2,1)

      DO 200 I=LIMITS(PAGE_NUM,1),LIMITS(PAGE_NUM,2)
      IF (ATTRIB(I).EQ.2) THEN
          STAT=LIB$PUT_SCREEN(LABEL(I)(1:LEN(I)),ROW(I),COL(I),2)
      ELSE IF (ATTRIB(I).EQ.6) THEN
          STAT=LIB$PUT_SCREEN(LABEL(I)(1:LEN(I)),ROW(I),COL(I),6)
      ELSE
          STAT=LIB$PUT_SCREEN(LABEL(I)(1:LEN(I)),ROW(I),COL(I),0)
      END IF
200   CONTINUE

!             Cycle Through Variables On Current Page
      IF (LIMITS(PAGE_NUM,3).EQ.0) THEN
          GOTO 405
      END IF
      DO 400 I=1,LIMITS(PAGE_NUM,3)
      IF (ERR_I.NE.0) THEN
```

```
          I=ERR_I
          ERR_I=0
       END IF
220    K=VAR(PAGE_NUM,I)

!                Treat Current Variable Character By Character.

230    DO 300 S=0,LEN(K)-1

!                 Put Current Value Of Variable To Screen
       IF (ATTRIB(K).EQ.2) THEN
          STAT=LIB$PUT_SCREEN(LABEL(K)(1:LEN(K)),ROW(K),COL(K),2)
       ELSE IF (ATTRIB(K).EQ.6) THEN
          STAT=LIB$PUT_SCREEN(LABEL(K)(1:LEN(K)),ROW(K),COL(K),6)
       ELSE
          STAT=LIB$PUT_SCREEN(LABEL(K)(1:LEN(K)),ROW(K),COL(K),0)
       END IF
250    STAT=LIB$SET_CURSOR(ROW(K),COL(K)+S)
       CALL READER                                  !  Get Next Keystroke
         IF (CTRLY) THEN                            ! Exit On CTRL/Y
           GOTO 9000
         END IF
         CHARACTER=INPUT(1:1)
         J=IOSTAT_BLOCK(4)
         IF (CHARACTER.EQ.BAK)  THEN                ! Backspace
            CHARACTER=' '
            IF (S.GT.0) THEN
                 S=S-1
            END IF
         ELSE IF (CHARACTER.EQ.CR) THEN             ! Carriage Return-Next
variable
             GOTO 310
          ELSE IF (CHARACTER.EQ.ESC) THEN           ! Escape Sequences
             IF (INPUT(J:J).EQ.'C') THEN  !  Right Arrow-Skip A Space
                  IF (S.EQ.LEN(K)-1) THEN
                      S=0
                  ELSE
                      S=S+1
                  END IF
                  GOTO 250
             ELSE IF (INPUT(J:J).EQ.'D') THEN  !  Left Arrow-Backspace
                  IF (S.EQ.0) THEN
                      S=LEN(K)-1
                  ELSE
                      S=S-1
                  END IF
                  GOTO 250
             ELSE IF ((INPUT(J:J).GE.'p').AND.   !  Convert Key Pad
     2             (INPUT(J:J).LE.'y')) THEN !     To Numeric
                  CHARACTER=CHAR(ICHAR(INPUT(J:J))-64)
             ELSE IF (INPUT(J:J).EQ.'n') THEN
                  CHARACTER='.'
```

```
          ELSE IF ((INPUT(J:J).EQ.'M').OR.
     2          ((INPUT(J:J).GE.'P').AND.
     2          (INPUT(J:J).LE.'S')) ) THEN
                GOTO 310
          ELSE
                STAT=LIB$PUT_SCREEN(BEL)
                GOTO 250
          END IF
        END IF

!               Test Character Format And Insert In Variable

         IF ( ( (ALPHA(K).EQ.'N').AND.
     2      (CHARACTER.GE.'0').AND.
     2      (CHARACTER.LE.'9'))
     2    .OR.( (ALPHA(K).EQ.'N').AND.
     2      (CHARACTER.EQ.'.'))
     2    .OR.    ((ALPHA(K).NE.'N').AND.
     2      (CHARACTER.GE.' ').AND.(CHARACTER.LE.DELIMIT) )) THEN
           LABEL(K)(S+1:S+1)=CHARACTER
         ELSE
           STAT=LIB$PUT_SCREEN(BEL)
           GOTO 250
         END IF
300     END DO
!               Edit Single Variable Here
       IOS=0
310    J=LEN(K)
       IF (ALPHA(K).EQ.'N') THEN   ! Is This Variable Numeric?
!                Zero Supress, Right Justify, Blank Fill Numerics
          READ(LABEL(K)(1:J),G(J),IOSTAT=IOS,ERR=390) Z
          IF (Z.EQ.0) THEN
            WRITE(LABEL(K),G(J+1),IOSTAT=IOS,ERR=390) Z
          ELSE
            H=G(J+1)
            IF (Z.LT.1.0) THEN
           S=J
            ELSE
           S=LOG10(Z)
           S=J-1-S
            END IF
            H(7:7)=CHAR(S+48)
            WRITE(LABEL(K),H,IOSTAT=IOS,ERR=390) Z
          END IF
          IF (LABEL(K)(J:J).EQ.'.') THEN
           X=BLANKS(1:J)
           X(2:J)=LABEL(K)(1:J-1)
           LABEL(K)(1:J)=X(1:J)
          END IF
      ELSE IF (ALPHA(K).EQ.'I') THEN
        READ (LABEL(K)(1:J),IX(J),IOSTAT=IOS,ERR=390) KZ
        WRITE (LABEL(K),IX(J),IOSTAT=IOS,ERR=390 ) KZ
```

```
          Z=KZ
        END IF
        IF (ALPHA(K).NE.'A') THEN
            IF (MAX(K).NE.BLANKS(1:7)) THEN
             IF (Z.GT.XMAX(K)) THEN
               IOS=1
               GOTO 390                    ! Exit If gt Than Max
             END IF
            END IF
            IF (MIN(K).NE.BLANKS(1:7)) THEN
             IF (Z.LT.XMIN(K)) THEN
               IOS=1
               GOTO  390                   ! Exit If lt Than Min
             END IF
            END IF
        END IF
!               Check For TLU (Table Look Up) To Verify Coded Fields

        IF ( (TLU(K).NE.0).AND.(FILES_OPEN(TLU(K)-90)) ) THEN
            CALL TABLE_LOOK_UP(TLU(K),IOS,X,LONG,LABEL(K),LEN(K))
            STAT=LIB$PUT_SCREEN(X(1:LONG),ROW(K),COL(K)+LEN(K)+1)
        END IF

!              End Error Checking For This Field
!               Check To See If Errors Have Occured

390     IF (IOS.NE.0) THEN
            ATTRIB(K)=BLINK+INVERSE
        ELSE
            ATTRIB(K)=INVERSE
        END IF

!              Put Variable To Screen
        IF (ATTRIB(K).EQ.2) THEN
            STAT=LIB$PUT_SCREEN(LABEL(K)(1:LEN(K)),ROW(K),COL(K),2)
        ELSE IF (ATTRIB(K).EQ.6) THEN
            STAT=LIB$PUT_SCREEN(LABEL(K)(1:LEN(K)),ROW(K),COL(K),6)
            STAT=LIB$PUT_SCREEN(BEL)
        ELSE
            STAT=LIB$PUT_SCREEN(LABEL(K)(1:LEN(K)),ROW(K),COL(K),0)
        END IF

       IF (IOS.NE.0) THEN
        GOTO 230
        END IF

!              Analyze Escape Sequences

        IF (CHARACTER.EQ.ESC) THEN
            J=IOSTAT_BLOCK(4)
            IF (INPUT(J:J).EQ.'R') THEN  ! PF4-Previous Variable
                IF (I.EQ.1) THEN
```

```
                    I=LIMITS(PAGE_NUM,3)
                ELSE
                    I=I-1
                END IF
                GOTO 220
            ELSE IF (INPUT(J:J).EQ.'S') THEN   ! PF3-Skip To Next
 Variable
                IF (I.EQ.LIMITS(PAGE_NUM,3)) THEN
                    I=1
                ELSE
                    I=I+1
                END IF
                GOTO 220
            ELSE IF (INPUT(J:J).EQ.'Q') THEN  !  PF2-Next Page
                IF (PAGE_NUM.EQ.MAX_PAGE) THEN
                    PAGE_NUM=2
                ELSE
                   PAGE_NUM=PAGE_NUM+1
                END IF
                GOTO 410
            ELSE IF (INPUT(J:J).EQ.'P') THEN  !  PF1-Previous Page
                IF (PAGE_NUM.EQ.2) THEN
                    PAGE_NUM=MAX_PAGE
                ELSE
                    PAGE_NUM=PAGE_NUM-1
                END IF
                GOTO 410
            ELSE IF (INPUT(J:J).EQ.'M') THEN  ! Enter Key-Done
                CHARACTER=ENTER
                GOTO 500
            END IF
        END IF
400     END DO    !  End Of Loop For All Variables, This Page
405     IF (PAGE_NUM.GE.MAX_PAGE) THEN
            PAGE_NUM=2
        ELSE
            PAGE_NUM=PAGE_NUM+1
        END IF
410     CONTINUE
500     END DO   !  End Of Loop, All Pages
!   Edit All Pages Here
        CHARACTER=' '   ! Reset Control For Main Loop

!               Compare Pairs Of Variables

        DO 800 I=2,MAX_PAGE
        DO 800 J=1,LIMITS(I,3)
        K=VAR(I,J)
        IF (RECNUM(K).NE.0) THEN
            IF (COMPR(K).EQ.'LT') THEN
             IF (LABEL(K).GE.LABEL(RECNUM(K))) THEN
                 IOS=1
```

```
              END IF
             ELSE IF (COMPR(K).EQ.'LE') THEN
              IF (LABEL(K).GT.LABEL(RECNUM(K))) THEN
                  IOS=1
              END IF
             ELSE IF (COMPR(K).EQ.'GE') THEN
              IF (LABEL(K).LT.LABEL(RECNUM(K))) THEN
                  IOS=1
              END IF
             ELSE IF (COMPR(K).EQ.'GT') THEN
              IF (LABEL(K).LE.LABEL(RECNUM(K))) THEN
                  IOS=1
              END IF
             END IF
             IF (IOS.EQ.1) THEN
              IOS=0
              ERR_PAGE=I
              ERR_I=J
              STAT=LIB$PUT_SCREEN(BELL)
              ATTRIB(K)=BLINK+INVERSE
              GOTO 140
             ELSE
              ATTRIB(K)=INVERSE
             END IF
         END IF
800      CONTINUE

!    Reconstruct Data Buffer--FULLREC In /databuff/
         IF (CTRLY) THEN
             GOTO 9000
         ELSE
             DO 1000 I=1,MAX_PAGE
             DO 1000 J=1,LIMITS(I,3)
             K=VAR(I,J)
1000         FULLREC(START(K):START(K)+LEN(K)-1)=LABEL(K)
         END IF

! Exit Subroutine
9000 Succ=NORMAL
         STAT=LIB$PUT_LINE(Numeric)  ! Put Terminal In Numeric Mode
         DO 9100 I=1,MAX_PAGE
         DO 9100 J=1,LIMITS(I,3)
         K=VAR(I,J)
9100 ATTRIB(K)=INVERSE
         IOS=0

         RETURN
         END


C        Include file CONTROLY.FOR
```

```
C     Code to deal with control-Y break

      LOGICAL*4 CTRLY
      INTEGER*4 CTRLY_MASK,OLD_MASK
      COMMON /controly/ CTRLY,CTRLY_MASK,OLD_MASK

C     Include file for subroutine CTRLY_AST
C
      SUBROUTINE CTRLY_AST

      INCLUDE 'CONTROLY'
      CTRLY=.TRUE.
      RETURN
      END

C     Include file for subroutine INIT_IODRIVER

      SUBROUTINE INIT_IODRIVER(FILESPEC)
      INCLUDE 'IODRVCOM'
      INCLUDE '($IODEF)'

      IF (CODE.EQ.0) THEN
           CODE=IO$_READVBLK        ! Read Logical Block
     2     .OR. IO$M_NOECHO       ! Do Not Echo
     2     .OR. IO$M_TRMNOECHO    ! Do Not Echo Terminators
     2     .OR. IO$M_ESCAPE       ! Allow All ASCII Escape Sequences
!    2     .OR. IO$M_CVTLOW       ! Convert To All Upper Case
!    2     .OR. IO$M_NOFILTER     ! No Screen Editing

           P_FOUR='                '
           DO 100 I=1,16
           P_FOUR(I:I)=CHAR(255)
100        CONTINUE
      END IF

      STATUS=SYS$ASSIGN('SYS$INPUT',IN_CHAN, , )
      IF(STATUS.NE.SS$_NORMAL) CALL LIB$SIGNAL(%VAL(STATUS))

      RETURN
      END

      SUBROUTINE READER (CHAR_OUT)
      IMPLICIT INTEGER*4 (S)
      INCLUDE '($IODEF)'
      INCLUDE 'IODRVCOM'
      INTEGER*4 IN_BUF_SIZE

      PARAMETER (IN_BUF_SIZE=10)

      STATUS=SYS$QIOW(,
     2               %VAL(IN_CHAN),
     2          %VAL(CODE),
```

```
     2          IOSB,
     2          ,,
     2          %REF(INPUT),
     2          %VAL(IN_BUF_SIZE),
     2          ,
     2          %DESCR(P_FOUR),
     2          ,,)

        IF ( STATUS.NE. 1)   CALL LIB$SIGNAL(%VAL(STATUS))
      IF  ( IOSB.NE. 1)    CALL LIB$SIGNAL(%VAL(IOSB))
      RETURN
      END

      SUBROUTINE Q_AST_CTRLY

      EXTERNAL CTRLY_AST
      INTEGER*4 STATUS4,SYS$ASSIGN,LIB$DISABLE,SYS$QIOW
      INCLUDE 'IODRVCOM'
      INCLUDE '($IODEF)'
      INCLUDE 'CONTROLY'

      IF (CODE_AST.EQ.0) THEN
           CODE_AST=IO$_SETMODE.OR.IO$M_CTRLYAST
           CTRLY_MASK='02000000'X
      END IF

!     Get A Channel For CTRL/Y Interrupt

      STATUS4=SYS$ASSIGN('SYS$INPUT',IN_CHAN_AST, , )
      IF(STATUS4.NE.1) CALL LIB$SIGNAL(%VAL(STATUS4))

!     Disable CTRL/Y Interrupts At DCL Level
      STATUS4=LIB$DISABLE_CTRL(CTRLY_MASK,OLD_MASK)
      IF(STATUS4.NE.1)    CALL LIB$SIGNAL(%VAL(STATUS4))

!     Queue An AST To Handle CTRL/Y Interrupts

      STATUS4=SYS$QIOW(,
     2                %VAL(IN_CHAN_AST),
     2          %VAL(CODE_AST),
     2          IOSB_AST,
     2          ,,
     2          CTRLY_AST,       ! Queues This AST Until CTRL/Y
     2          ,,,,)

      IF ( STATUS4.NE.1)    CALL LIB$SIGNAL(%VAL(STATUS4))
      IF  ( IOSB_AST.NE.1)    CALL LIB$SIGNAL(%VAL(IOSB_AST))

      CTRLY=.FALSE.

      RETURN
      END
```

```
      SUBROUTINE DQ_AST_CTRLY

      INTEGER*4 STATUS4,SYS$QIOW,LIB$DISABLE

      INCLUDE 'IODRVCOM'
      INCLUDE 'CONTROLY'

!     De-Queue An AST To Handle CTRL/Y Interrupts

      STATUS4=SYS$QIOW(,
     2                %VAL(IN_CHAN_AST),
     2          %VAL(CODE_AST),
     2          IOSB_AST,
     2          ,,
     2          %VAL(0),      ! Disable All AST's This Channel
     2          ,,,,)

!     Check Status Of IO Request

      IF ( STATUS4.NE.1)   CALL LIB$SIGNAL(%VAL(STATUS4))
      IF  ( IOSB_AST.NE.1)   CALL LIB$SIGNAL(%VAL(IOSB_AST))

!     Enable CTRL/Y Interrupts At DCL Level

      STATUS4=LIB$DISABLE_CTRL(OLD_MASK,CTRLY_MASK)
      IF(STATUS4.NE.1)   CALL LIB$SIGNAL(%VAL(STATUS4))

      CTRLY=.FALSE.

      RETURN
      END

C     Include file for INIT_ARRAYS subroutine
C
      SUBROUTINE INIT_ARRAYS(Filespec)
      INCLUDE 'DATABUFF'

      CHARACTER*(*) Filespec
      CHARACTER*30 DLABEL,BLANK20
      CHARACTER*7 DNAME,DMAX,DMIN,DFIELD,BLANK7
      CHARACTER*1 DALPHA
      CHARACTER*2 DCOMPR
      INTEGER*2 DSTART,
     2       DLEN,
     2       DPAGE,
     2       DROW,
     2       DCOL,
     2       DTLU,
     2       DRECNUM,
     2       INVERSE,NORM,PAGE_NUM
```

```
      D       I       M       E       N       S       I       O       N
DLABEL(100),DNAME(100),DSTART(100),DLEN(100),DPAGE(100),
     2       DROW(100),DCOL(100),DALPHA(100),DMAX(100),DMIN(100),
     2       DFIELD(100),DCOMPR(100),DTLU(100)

     PARAMETER (BLANK20='                                        ',
     2         BLANK7='       ',
     2         INVERSE=2,
     2         NORM=0)

     DATA DLABEL(1) /'NASA Alloys Data Base'/,
     2    DLEN(1) /21/,
     2    DPAGE(1) /1/,
     2    DROW(1) /1/,
     2    DCOL(1) /30/,
     2    DLABEL(2) /'Page-Row-Column:'/,
     2    DLEN(2) /16/,
     2    DPAGE(2) /1/,
     2    DROW(2) /1/,
     2    DCOL(2) /58/,
     2    DLABEL(4) /'Description Maintenance'/,
     2    DLEN(4) /23/,
     2    DPAGE(4) /1/,
     2    DROW(4) /2/,
     2    DCOL(4) /29/,
     2    DNAME(3) /'KEY'/,
     2    DSTART(3) /122/,
     2    DLEN(3) /6/,
     2    DPAGE(3) /1/,
     2    DROW(3) /1/,
     2    DCOL(3) /75/
    DATA DLABEL(5) /'Screen Label:'/,
    2    DLEN(5) /13/,
    2    DPAGE(5) /2/,
    2    DROW(5) /4/,
    2    DCOL(5) /10/,
    2    DNAME(6) /'LABEL'/,
    2    DSTART(6) /1/,
    2    DLEN(6) /30/,
    2    DPAGE(6) /2/,
    2    DROW(6) /4/,
    2    DCOL(6) /24/,
    2    DALPHA(6) /'A'/,
    2    DLABEL(7) /'Field Name:'/,
    2    DLEN(7) /11/,
    2    DPAGE(7) /2/,
    2    DROW(7) /4/,
    2    DCOL(7) /55/,
    2    DNAME(8) /'NAME'/,
    2    DSTART(8) /31/,
    2    DLEN(8) /7/,
    2    DPAGE(8) /2/,
```

```
2     DROW(8) /4/,
2     DCOL(8) /67/,
2     DALPHA(8) /'A'/
DATA  DLABEL(9) /'Start Position:'/,
2     DLEN(9) /15/,
2     DPAGE(9) /2/,
2     DROW(9) /6/,
2     DCOL(9) /10/,
2     DNAME(10) /'START'/,
2     DSTART(10) /38/,
2     DLEN(10) /3/,
2     DPAGE(10) /2/,
2     DROW(10) /6/,
2     DCOL(10) /26/,
2     DALPHA(10) /'I'/,
2     DMAX(10) /'   656.'/,
2     DMIN(10) /'     . 0'/,
2     DLABEL(11) /'Field Length:'/,
2     DLEN(11) /13/,
2     DPAGE(11) /2/,
2     DROW(11) /6/,
2     DCOL(11) /32/,
2     DNAME(12) /'LEN'/,
2     DSTART(12) /41/,
2     DLEN(12) /3/,
2     DPAGE(12) /2/,
2     DROW(12) /6/,
2     DCOL(12) /46/,
2     DALPHA(12) /'I'/,
2     DMAX(12) /'    30.'/,
2     DMIN(12) /'     1.'/,
2     DLABEL(13) /'Screen Page:'/,
2     DLEN(13) /12/,
2     DPAGE(13) /2/,
2     DROW(13) /8/,
2     DCOL(13) /10/,
2     DNAME(14) /'PAGE'/,
2     DSTART(14) /44/,
2     DLEN(14) /1/,
2     DPAGE(14) /2/,
2     DROW(14) /8/,
2     DCOL(14) /23/,
2     DALPHA(14) /'I'/,
2     DMAX(14) /'      9'/,
2     DMIN(14) /'      1'/
DATA  DLABEL(15) /'Row:'/,
2     DLEN(15) /4/,
2     DPAGE(15) /2/,
2     DROW(15) /8/,
2     DCOL(15) /26/,
2     DNAME(16) /'ROW'/,
2     DSTART(16) /45/,
```

```
2     DLEN(16) /2/,
2     DPAGE(16) /2/,
2     DROW(16) /8/,
2     DCOL(16) /31/,
2     DALPHA(16) /'I'/,
2     DMAX(16) /'   24.'/,
2     DMIN(16) /'    1.'/,
2     DLABEL(17) /'Column:'/,
2     DLEN(17) /7/,
2     DPAGE(17) /2/,
2     DROW(17) /8/,
2     DCOL(17) /36/,
2     DNAME(18) /'COL'/,
2     DSTART(18) /47/,
2     DLEN(18) /3/,
2     DPAGE(18) /2/,
2     DROW(18) /8/,
2     DCOL(18) /44/,
2     DALPHA(18) /'I'/,
2     DMAX(18) /'    80.'/,
2     DMIN(18) /'    1.'/
DATA  DLABEL(19) /'Alpha/Numeric/Integer:'/,
2     DLEN(19) /22/,
2     DPAGE(19) /2/,
2     DROW(19) /8/,
2     DCOL(19) /49/,
2     DNAME(20) /'ALPHA'/,
2     DSTART(20) /50/,
2     DLEN(20) /1/,
2     DPAGE(20) /2/,
2     DROW(20) /8/,
2     DCOL(20) /72/,
2     DALPHA(20) /'A'/
2     DLABEL(21) /'Maximum Value:'/,
2     DLEN(21) /14/,
2     DPAGE(21) /2/,
2     DROW(21) /10/,
2     DCOL(21) /10/,
2     DNAME(22) /'MAX'/,
2     DSTART(22) /51/,
2     DLEN(22) /7/,
2     DPAGE(22) /2/,
2     DROW(22) /10/,
2     DCOL(22) /25/,
2     DALPHA(22) /'A'/
DATA  DLABEL(23) /'Minimum Value:'/,
2     DLEN(23) /14/,
2     DPAGE(23) /2/,
2     DROW(23) /10/,
2     DCOL(23) /34/,
2     DNAME(24) /'MIN'/,
2     DSTART(24) /58/,
```

```
2     DLEN(24) /7/,
2     DPAGE(24) /2/,
2     DROW(24) /10/,
2     DCOL(24) /49/,
2     DALPHA(24) /'A'/
2     DLABEL(25) /'This Field Must Be'/,
2     DLEN(25) /18/,
2     DPAGE(25) /2/,
2     DROW(25) /12/,
2     DCOL(25) /10/,
2     DNAME(26) /'COMPR'/,
2     DSTART(26) /72/,
2     DLEN(26) /2/,
2     DPAGE(26) /2/,
2     DROW(26) /12/,
2     DCOL(26) /29/,
2     DALPHA(26) /'A'/
DATA  DLABEL(27) /'The Field Named'/,
2     DLEN(27) /16/,
2     DPAGE(27) /2/,
2     DROW(27) /12/,
2     DCOL(27) /33/,
2     DNAME(28) /'FIELD'/,
2     DSTART(28) /65/,
2     DLEN(28) /7/,
2     DPAGE(28) /2/,
2     DROW(28) /12/,
2     DCOL(28) /50/,
2     DALPHA(28) /'A'/
DATA  DLABEL(29) /'Table Look Up # :'/,
2     DLEN(29) /17/,
2     DPAGE(29) /2/,
2     DROW(29) /14/,
2     DCOL(29) /10/,
2     DNAME(30) /'TLU'/,
2     DSTART(30) /74/,
2     DLEN(30) /2/,
2     DPAGE(30) /2/,
2     DROW(30) /14/,
2     DCOL(30) /28/,
2     DALPHA(30) /'I'/,
2     DMAX(30) /'    99.'/,
2     DMIN(30) /'     0.'/,
2     DTLU(30) /91/
DATA  DLABEL(31) /'Enter D To Delete:'/,
2     DLEN(31) /18/,
2     DPAGE(31) /2/,
2     DROW(31) /21/,
2     DCOL(31) /30/,
2     DNAME(32) /'DEL'/,
2     DSTART(32) /128/,
2     DLEN(32) /1/,
```

```
     2    DPAGE(32) /2/,
     2    DROW(32) /21/,
     2    DCOL(32) /49/,
     2    DALPHA(32) /'A'/
     DATA DLABEL(33) /'PF3-Prv Field'/,
     2    DLEN(33) /13/,
     2    DPAGE(33) /2/,
     2    DROW(33) /24/,
     2    DCOL(33) /1/,
     2    DLABEL(34) /'PF4-Nxt Field'/,
     2    DLEN(34) /13/,
     2    DPAGE(34) /2/,
     2    DROW(34) /24/,
     2    DCOL(34) /16/,
     2    DLABEL(35) /'PF1-Prv Scrn'/,
     2    DLEN(35) /12/,
     2    DPAGE(35) /2/,
     2    DROW(35) /24/,
     2    DCOL(35) /31/,
     2    DLABEL(36) /'PF2-Nxt Scrn'/,
     2    DLEN(36) /12/,
     2    DPAGE(36) /2/,
     2    DROW(36) /24/,
     2    DCOL(36) /45/,
     2    DLABEL(37) /'ENTER-Update Data'/,
     2    DLEN(37) /17/,
     2    DPAGE(37) /2/,
     2    DROW(37) /24/,
     2    DCOL(37) /61/

     STAT=LIB$ERASE_PAGE
     WRITE(6,90)

     I=1
     IF (Filespec.EQ.'FILESPEC') THEN
         SPECUNIT=89
         LRECL=128
         DO 100 WHILE (DPAGE(I).NE.0)
          LABEL(I)=DLABEL(I)
          NAME(I)=DNAME(I)
          START(I)=DSTART(I)
          LEN(I)=DLEN(I)
          PAGE(I)=DPAGE(I)
          ROW(I)=DROW(I)
          COL(I)=DCOL(I)
          ALPHA(I)=DALPHA(I)
          MAX(I)=DMAX(I)
          MIN(I)=DMIN(I)
          FIELD(I)=DFIELD(I)
          COMPR(I)=DCOMPR(I)
          TLU(I)=DTLU(I)
          I=I+1
```

```
100          END DO
            DO 110 J=I,300
             PAGE(J)=0
             START(J)=0
110         END DO
      ELSE IF(Filespec.EQ.'DATAFILE') THEN
            SPECUNIT=88
            LRECL=656
            OPEN(UNIT=90,FILE='FILESPEC',STATUS='OLD',
     2       ORGANIZATION='INDEXED',ACCESS='KEYED',
     2       IOSTAT=IOS)
            IF (IOS.EQ.0) THEN
              DO WHILE (IOS.EQ.0)
              READ(UNIT=90,IOSTAT=IOS)FULLREC(1:128)
              IF(IOS.EQ.0) THEN
                READ(FULLREC,500)LABEL(I),NAME(I),START(I),LEN(I),
     2           PAGE(I),ROW(I),COL(I),ALPHA(I),MAX(I),MIN(I),
     2           FIELD(I),COMPR(I),TLU(I)
             I=I+1
              END IF
              END DO
             CLOSE(UNIT=90,IOSTAT=IOS)
              DO 120 J=I,300
             PAGE(J)=0
             START(J)=0
             XMIN(J)=0
             XMAX(J)=0
120           END DO
           END IF
      END IF

      DO 130 I=1,10                        ! Mark All TLU Files Closed
      CLOSE(UNIT=I+90,STATUS='KEEP',IOSTAT=IOS,ERR=125)
125   WRITE(6,129)I
130   FILES_OPEN(I)=.FALSE.

      LIMITS(1,1)=1
      L=0
      PAGE_NUM=1
      I=1
      DO 300 WHILE (PAGE(I).NE.0)
      IF (PAGE(I).NE.PAGE_NUM) THEN    ! Found New Page, Set
          LIMITS(PAGE_NUM,2)=I-1       !  Data For This Page
          LIMITS(PAGE_NUM,3)=L
          L=0
          PAGE_NUM=PAGE(I)
          LIMITS(PAGE_NUM,1)=I
      END IF
      IF (START(I).NE.0) THEN          ! Variable Field
          L=L+1
          VAR(PAGE_NUM,L)=I
          ATTRIB(I)=INVERSE
```

```
            IF (ALPHA(I).NE.'A') THEN   ! Convert MAX, MIN To
              IF(MIN(I).NE.'        ') THEN
             READ (MIN(I),310,ERR=280) S  !  Numeric For
             XMIN(I)=S                    !  Numeric Data Fields
              ELSE
             XMIN(I)=-9999999
              END IF
 280          IF (MAX(I).NE.'        ') THEN
             READ(MAX(I),310,ERR=290) S
             XMAX(I)=S
              ELSE
             XMAX(I)=9999999
              END IF
            END IF
            IF (TLU(I).NE.0) THEN           ! Check For Table Look Up
             IF ((TLU(I).LE.100).AND.(TLU(I).GE.91)) THEN
                 IF (.NOT.FILES_OPEN(TLU(I)-90)) THEN
                   OPEN (UNIT=TLU(I),STATUS='OLD',DISP='KEEP',
     2               ORGANIZATION='INDEXED',ACCESS='KEYED',
     2               READONLY,SHARED,IOSTAT=IOS)
                    IF (IOS.EQ.0) THEN
                  FILES_OPEN(TLU(I)-90)=.TRUE.
                  WRITE(6,285)TLU(I),NAME(I)
                   ELSE
                  WRITE(6,287)TLU(I),NAME(I)
                   END IF
                 ELSE
                  WRITE(6,286)TLU(I),NAME(I)
                 END IF
             ELSE
                 WRITE(6,287)TLU(I),NAME(I)
             END IF
            END IF
        ELSE
            ATTRIB(I)=NORM                  ! Display (Prompt) Field
        END IF
 290    I=I+1
 300    END DO
        LIMITS(PAGE_NUM,2)=I-1
        LIMITS(PAGE_NUM,3)=L
        MAX_PAGE=PAGE_NUM
        HEADLINE=ROW(LIMITS(1,2))+1

! Set Up Cross Reference Array For Internal Comparisons
        DO 400 I=2,MAX_PAGE
        DO 400 J=1,LIMITS(I,3)
        K=VAR(I,J)
        IF (FIELD(K).NE.'        ') THEN
            DO 380 L=2,MAX_PAGE
            DO 380 M=1,LIMITS(L,3)
            N=VAR(L,M)
            IF (FIELD(K).EQ.NAME(N)) THEN
```

```
            RECNUM(K)=N
            GOTO 400
           END IF
380        CONTINUE
      END IF
400   CONTINUE

! End Of Subroutine
      RETURN

90    FORMAT(1H ,///,T10,'Verifying And Opening Data Files.')
129   FORMAT(1H ,T10,'File FOR0',I2,' Closed.')
285   FORMAT(1H ,T10,'File FOR0',I2,' Opened To Verify Variable ',
     2        A8)
286   FORMAT(1H ,T4,'* * * Warning!  File FOR0',I2,
     2        'Also Used To Verify Variable ',A8)
287   FORMAT(1H ,T4,'* * * Warning!  Invalid File Number ',I2,
     2        'For Validation Of Variable ',A8,
     2        '.  Validation Bypassed.')
310   FORMAT(BNF7.0)
500   FORMAT(A30,A7,I3,I3,I1,I2,I3,A1,A7,A7,A7,A2,I2)
      END


C     Include file for subroutine HEAD_SET
C
      SUBROUTINE HEAD_SET
      CHARACTER*9 Date_buff

      CALL DATE(Date_buff)

      STAT=LIB$ERASE_PAGE(1,1)
      STAT=LIB$SET_CURSOR(1,1)
      WRITE (6,100)Date_buff
100   FORMAT(1H ,T20,'N A S A                    Alloys Data Base',T70,A9)
      RETURN
      END

C     Include file defining FULLREC character array.

      CHARACTER*1000 FULLREC
      CHARACTER*30 LABEL
      CHARACTER*20 FILE_ERROR,Succ
      CHARACTER*7 NAME,MAX,MIN,FIELD
      CHARACTER*1 ALPHA
      CHARACTER*2 COMPR
      CHARACTER*6 Flag,MODIFY,INSERT,NORMAL,COMPLETE,FAILED

      LOGICAL*1 FILES_OPEN(10)

      INTEGER*2 START,
```

```
     2       LEN,
     2       PAGE,
     2       ROW,
     2       COL,
     2       TLU,
     2       RECNUM,
     2       LIMITS,
     2       VAR,
     2       MAX_PAGE,
     2       HEADLINE,
     2       ATTRIB,
     2       SPECUNIT,
     2       LRECL
     REAL*4  XMAX, XMIN

     DIMENSIONLABEL(300),NAME(300),START(300),LEN(300),PAGE(300),
     2       ROW(300),COL(300),ALPHA(300),MAX(300),MIN(300),
     2       FIELD(300),COMPR(300),TLU(300),RECNUM(300),
     2       XMAX(300),XMIN(300)
     DIMENSION VAR(9,100),LIMITS(9,3),ATTRIB(300)

     COMMON /databuff/ FULLREC,LABEL,NAME,START,LEN,PAGE,MAX_PAGE,
     2              ROW,COL,ALPHA,MAX,MIN,ATTRIB,XMAX,XMIN,
     2              FIELD,COMPR,TLU,RECNUM,LIMITS,VAR,HEADLINE,
     2              Succ,Flag,MODIFY,INSERT,NORMAL,COMPLETE,
     2              FAILED,SPECUNIT,FILE_ERROR,LRECL,
     2              FILES_OPEN
```