AGSO

CREATING ARC/INFO COVERAGES FROM MICROSTATION DESIGN FILES - SOME PROCEDURES AND AML ROUTINES

BY R BRODIE

RECORD 1994/55

AUSTRALIAN GEOLOGICAL SURVEY ORGANISATION

AGSO
AUSTRALIAN
GEOLOGICAL SURVEY
ORGANISATION

Creating Arc/Info Coverages from Microstation Design Files - some Procedures and AML routines.

Record 1994/55

by

Ross Brodie

Australian Geological Survey Organisation
Environmental Geoscience and Groundwater Program

DEPARTMENT OF PRIMARY INDUSTRIES AND ENERGY

Minister for Resources: Hon. David Beddall, MP
Secretary: Greg Taylor

AUSTRALIAN GEOLOGICAL SURVEY ORGANISATION

Executive Director: Harvey Jacka

ISSN: 1039-0073
ISBN: 0 642 22117 0

Abstract

The bulk of AGSO's geoscientific maps are produced using Intergraph CAM facilities. In addition to providing the final hardcopy product, the design (IGDS) files are an important resource of digitised spatial data. A prime example is the Murray Basin Hydrogeological Map Series. The series consists of 28 mapsheets which delineate aquifer characteristics and highlight salinity hazard over the basin. The conversion of the information depicted on the map and stored in the IGDS files into an Arc/Info based GIS database is currently in progress.

Procedures and AML-based tools were designed to assist in the construction of the GIS coverages. These include tools to partially automate the conversion, add or confirm project definitions, derive feature attributes from elements of the IGDS file structure, enhance display of potential errors within ArcEdit and validate aspects of the final coverage. The graphical user interface ArcTools was customised into an interface appropriate for the project. In this way users can readily access these tools via integrated menus.

TABLE OF CONTENTS

APPENDICES

FIGURES

Creating Arc/Info Coverages from Microstation Design Files

1.) Introduction

In 1982, a local drafting service collaborated with AGSO (then BMR) in a pilot test in computer aided geological mapping using Intergraph technology (Hillier, 1992). For over a decade, the publication of AGSO's map products have relied significantly on the use of an expanding corporate Intergraph Microstation CAM facility. In addition to providing the final hardcopy product, the design (IGDS) files making up the map are an important source of digital data.

In 1987, the 1:250 000 scale Murray Basin Hydrogeological Map Series was initiated. The series, containing 28 mapsheets, is designed to show the influence of groundwater on salinisation, highlight present and potential salinity hazard, delineate useable groundwater resources and enhance community awareness of prevailing groundwater systems (Evans, 1992). Each mapsheet of the series contains a wealth of hydrogeological information, including salinity/yield characteristics, potentiometry, depth to watertable and structural tops to aquifers.

The process of constructing an Arc/Info GIS database from the Intergraph design (IGDS) files commenced in 1994. To date, a total of 209 Arc/Info coverages have been created, equating to an average of 14 groundwater-related themes derived from each mapsheet. To speed the conversion process, procedures and AML-based tools were developed. This Record outlines the use of these routines.

2) Construction of the GIS database

The construction of an Arc/Info coverage from a source Intergraph Microstation design file is a multi-step process. These steps are summarised in Table 1. Aspects of this processing are discussed at length in many ESRI publications and by other authors (eg. Musto, 1990).

Table 1: Summary of Data Processing Steps

STEP	PROCESS	COMMANDS
1	Dearchive and prepare design file for reformatting	ustation commands
2	Conversion of design file to raw GIS coverage	igdsinfo, igdsarc
3	Registration of tics	arcedit snapping, $ID editing
4	Clean data of offshoots, duplicate arcs, unclosed polygons	arcedit commands, clean
5	Topology building	createlabels (polys), arcedit commands, build
6	Identify and correct topological errors	intersecterr, labelerrors, nodeerrors arcedit draw environment
7	Add and populate attributes	additem, arcedit commands
8	Validate and correct attribute data	arcplot, arcview, arcedit
9	Transform to real world coordinates	transform, project
10	Rebuild and final check of mapsheet coverage Finalise documentation	build textedit

AML-based routines were developed to streamline this data processing. These include tools to:

- partially automate the IGDS-to-ARC conversion
- add standard feature attribute definitions
- derive feature attributes from elements of the IGDS file
- display potential topological or attribute errors within an editing environment
- validate aspects of the final GIS coverage.

The names and functions of theses routines are listed in Appendix 1. The AMLs are also maintained in a directory structure based on the model of shared work areas (Tucker et al, 1994). Members of the Murray Basin group (*mb*) have read access to the *mb* work area, enabling access to the AMLs. Write access is vested only to the GIS data administrator for the work area - the *mba* user. Pathways are established allowing an *mb* user to access an AML by simply typing its name (and any required arguments).

The directory also contains common projection files, symbol sets, unix scripts, documentation and GIS data, all accessible to *mb* users. With write access limited to a single user, management of GIS data and documentation operates in a controlled environment. Any *mb* user may copy a coverage and then perform edits on their version, but only the *mba* administrator can write the edited version back into the shared work area.

The tools are integrated into a modified version of ESRI's graphical user interface (ArcTools) called MBTools. ArcTools is a menu-driven system that provides a general interface to the Arc/Info toolbox. The modular design of ArcTools allows the development of customised interfaces, which Gallagher (1993) gives an example. For the Murray Basin project, selected ArcTools components were copied into the shared work area, and then added to or modified to provide an interface appropriate for the project.

Appendices 2 & 3 are examples of AML-based tools developed to assist in compiling topological and fully attributed Arc/Info coverages from IGDS files. The tools consist of three components - the underlying AML, the menu interface and the help document. The standards and routines developed for ESRI's ArcTools interface have been adopted. The tools have been designed to operate from the command line or within a menu interface.

For the tools to work, the components (menus, AMLs, icons, help files, data sets) need to be accessible. This requires the proper definition of *&amlpath* and *&menupath* in particular. A modified version of *setpaths.aml* available under the ArcTools directory may be useful in achieving this. The AMLs have been implemented using Arc/Info 6.1.1.

2.1) Source Intergraph design files

Primary digital data associated with the maps are stored as Intergraph design (dgn) files. These files need to be dearchived and prepared for reformatting into Arc/Info GIS coverages. A number of routine checks at this initial stage can minimise potential problems during conversion (Brodie & Tucker, 1992):

1) Information about the original data, such as bibliographic references, source, publication date and currency, scale and accuracy, map projection parameters and media type should be documented.

2) Details of any cartographic processes such as generalisation, reprojection or rescaling should also be recorded. The design files that are used should have gone through the polygonising stage of the cartographic process. This ensures that all polygons are closed, effectively reducing editing and cleaning of the GIS coverage.

3) Commonly, maps are rotated to fit orthogonal to the map sheet. This requires the mapping to be rotated back to a position appropriate for the projection used. Ideally, the coordinate system of the design plane in which the map is digitised should match the coordinates defined by the projection and map coordinate system. This omits the need to transform from an arbitrary design plane to real world coordinates. At the least, four registration points (typically IGDS cells at each mapsheet corner) should be provided to allow such a transformation.

4) Standardising design level definitions within a mapping project greatly assists the conversion process. The routine separating of themes into specified levels and the use of standard nomenclature for design files, assists in the orderly and complete transferral of data.

2.2) IGDS to ARC conversion

The *igdsarc* command is well documented in Arc/Info manuals (ESRI, 1991a; ESRI, 1991b) where the IGDS file structure and the options available for conversion are outlined. Within an IGDS file, map features are stored under different levels, each with defined specifications for colour, line weight, font and size. For a map series such as for Murray Basin hydrogeology these specifications are usually standardised across mapsheets (eg. all perennial lakes are on level 9). The *igdsinfo* command is useful in displaying information on the structure of an IGDS file. Data conversion is implemented by an interactive dialogue within the *igdsarc* command. Three approaches can be taken to separate the components of the source design file required for a coverage:

1) A secondary design file can be created within Intergraph Microstation, containing just the levels needed for the coverage. This is simply converted in total using the *$rest* subcommand of *igdsarc.*

```
Arc: igdsarc dgn/cover.dgn cover_o
Enter layer names and options (type END or $REST when done)
========================================
    Enter the 1st layer and options : $rest
```

2) Certain levels can be selected out from the original design file using the subcommands of *igdsarc*.

```
Arc: igdsarc dgn/cov.dgn cover_o
Enter layer names and options (type END or $REST when done)
========================================
Enter the 1st layer and options : sy 30 * * * *
Enter the 2nd layer and options :sytxt 31 * * * * textpoint
Enter the 3rd layer and options :tic 63 * * * * point
Enter the 4th layer and options : end
```

This option gives greater control on how specific elements are to be converted.

3) The original design file can be converted in its entirety and coverages separated out within ArcEdit at a later stage.

Within the ArcTools interface, the IGDS to Arc conversion tool is available in the *Data Conversion* category of the *Cover:Edit* type under *toolpicker*. This may be initiated from within *Edit Tools* via *Edit*. The user can choose to convert all IGDS levels or specify particular elements.

2.3) Correct Registration of Tics

The *igdsarc* command produces a coverage with tics corresponding to the minimum and maximum coordinate values. These should be repositioned exactly to the cells or monuments used to register the mapsheet, typically the four mapsheet corners. By default, these cells are converted to point features during conversion. This repositioning can be achieved using the snapping capability within ArcEdit:

Arcedit: *ec cover*
Arcedit: *ef tics*
Arcedit: *snapcoverage cover*
Arcedit: *snapfeature tic label*
Arcedit: *snapping closest **
Arcedit: *sel all*
Arcedit: *snap*

The tics may need to be moved to within the defined snapping distance before snapping is successful. The numbering of the tics is also critical. Tic IDs should match corresponding tics in the template coverage used in transforming to real world coordinates.

In MBTools, the *Snap* tool used to set the snapping environment may be initiated from the *Edit Tics* menu. The *Table Editor* tool was also modified to allow changing of tic IDs.

2.4) Cleaning of Digitising Errors

The cleaning of the digital data, in terms of mismatched nodes, offshoots, duplicate or missing arcs, should be done using the functionality of Arcedit (ESRI, 1991a). Using the global Arc *clean* command with unspecified or inappropriate tolerances can lead to erroneous results. Using a small tolerance may not be able to resolve existing duplicate arcs, slivers or undershoots. A large tolerance may induce distortions by collapsing polygons or merging inappropriate arcs. The *intersectarcs* command is used with appropriate node tolerances during an Arcedit session to calculate arc intersections and add nodes so that *clean* is not necessary.

Any Intergraph cells, such as those to register the mapsheet corners or arrow heads, are typically converted to label point features. These should be deleted so as not to be mistaken for valid data points such as boreholes or sample sites. By default, annotation displayed on the map is converted to a subclass called *anno.igds* in the resulting Arc/Info coverage. Redundant annotation which have no spatial significance, such as figure titles, should be deleted from this annotation.

The default ArcTools interface has a host of editing tools, eg *Edit Arcs & Nodes* has tools to correct under and overshoots, intersect, unsplit and delete arcs. The draw environment may be set to highlight arc intersections or node errors. Under MBTools, the *correct overshoot* tool was modified to delete dangling arcs which are less than a user-specified length. Duplicate arcs may be encountered in the final coverage after IGDS conversion. These can be difficult to detect and correct as the *arc intersect* option in the draw environment settings may not display all duplicate arcs. For this reason the *sel_int* tool was developed to select arcs found where intersection errors are detected. This is available in MBTools under the *More* select features available from the *Select graphic* menu.

2.5) Topology Building

The topology of the coverage can be constructed using *build* once the digitising errors have been corrected. The coverage type should be explicitly stated in the command as in some cases data may be corrupted. In particular, if a link coverage (arcs and points) is built as a polygon coverage, some of the attribute data (and points) may be lost. For coverages containing polygons, a label point is required for each polygon - *createlabels* is useful in this regard. A suite of Arc commands are available to check for topological errors, specifically *intersecterr, labelerrors* and *noderrors*. Alternatively a graphics file showing potential digitising errors can be generated using *editplot*.

Under Arctools, a host of tools is available under the *Cover:Edit* option of *Toolpicker*, including *Build features, Create labels* and *Error plot*. The *Rebuild features* tool was developed for MBTools to avoid building the wrong topology eg. polygons instead of points. The AML reads the current topology of the coverage and builds accordingly.

2.6) Adding of Attributes

Attributes may be attached to polygons, points or arcs by firstly adding items to the appropriate attribute table (*cover.pat, cover.aat*). This is achieved by using the *additem* command. Within MBTools, item definitions standard to the project can be added to coverage attribute tables by using the *Add all items* under the *Tables* type of *Toolpicker*. The underlying AML checks that the coverage is suitably named and has appropriate topological features. The item definitions can also be checked against project standards using the *Check all items* tool under the same category.

Updating the attributes is mainly done within Arcedit. The *forms* command is useful in the interactive selection and updating of features. For coverages containing many features with the same attributes, simply select these common features and use the *calc* or *moveitem* commands to update the attribute. With the *Table Editor* tool of ArcTools, feature attribute values may be added individually (*Edit*) or by selected sets (*Calc*).

Populating the attribute tables of GIS coverages can be a time consuming task. A number of AMLs were written to take advantage of features inherent within the original IGDS file. During the *igdsarc* conversion, an info table (*cover.acode*) is generated to store the graphical properties of the elements making up the IGDS file. This includes parameters such as level, colour, weight, style and annotation text. The coverage attribute table can be linked to the *cover.acode* by creating a relationship using *relate*. This allows GIS features to be selected on the basis of their IGDS properties:

```
Arc: relate add
Relation Name: cover
Table Identifier: cover.acode
Database Name: info
INFO Item: cover-id
Relate Column: cover-id
Relate Type: linear
Relate Access: ro
Relation Name:
Arc: relate save acode.rel
```

For example, to select all arcs that were on level 1 of the design file:

```
Arcedit: ec cover
Arcedit: de arcs tics
Arcedit: ef arcs
Arcedit: relate restore acode.rel
Arcedit: select cover//igds-level = 1
```

In many IGDS files, different themes are stored on standard levels eg. highways on level 1, secondary roads on level 3, minor roads on level 4, tracks on level 5. This means that attributes in the resulting coverage may be calculated on the basis of IGDS levels. The *igdslevel* AML takes advantage of the fact that the IGDS-level of each GIS feature is stored in the related *cover.acode* file. The user inputs the name of the *cover.acode* file, the target item in the attribute table to be updated, and up to ten codes which relate to a particular IGDS level. For example, in this way all highways (on level 1) can have type = 'H', secondary roads (level 3) have type = 'S', minor roads (level 4) have type = 'M' and so on. The AML is documented in Appendix 2.

Within the *igdsarc* command there is also a mechanism to preserve information stored as map annotation in the source design file. By using the *textpoint* option, the igds text is placed as a point positioned at the text origin. The first 12 characters of the text string are stored in the *igds-text* item of *cover.acode*. Text strings longer than the default can be accommodated by defining the *text_width* option of the *igdsarc* command. This is used to advantage in the *igdstext* tool. This AML was written to update attributes of contours, points or polygon labels based on the annotation depicted on the published map. Further details are outlined in Appendix 3. By relating the *cover.acode*, the information stored in the *igds-text* item can be copied into the appropriate item of the coverage attribute table. In this way, contour labels and spot heights on the map can be transferred to the attributes of contours and points of the GIS coverage, respectively.

The success of these AMLs largely depend on how the original map has been annotated. Attributing of contours depends on an adequate distribution of contour values in the original IGDS file. If not all contours have a labelled value, further editing will be required. Likewise, not all polygons are labelled on the map or labels may be separated from the polygon by lead lines. The coverage will require checking for unlabelled polygons or polygons with more than one label. The layout of the menus for the *igdslabel* and *igdstext* tools are presented in Figure 1.

In some cases, attributes of a particular feature are dependant on attributes of another feature within the same coverage. The colour fill of the published hydrogeological maps depict the salinity/yield characteristics of the aquifers - the colour (blue to red) indicates the salinity and the intensity (light to dark) indicates the yield. The resulting GIS coverages are networks, with both polygon and arc attributes. In essence, the arcs are contours of salinity and/or yield. Once the salinity/yield attributes of the polygon labels are updated, the arc attributes can be calculated automatically. An AML was implemented which, for each arc in turn, compares attributes of adjacent polygons and calculates the arc attributes accordingly. This results in significant time savings compared to manually updating each arc. The coverage can then be subdivided into an aquifer yield coverage and a groundwater salinity contour coverage.

Routines from other sources have also been incorporated into MBTools to minimise the task of attributing features. The *transector* tool by M.R. Muller (ESRI) and M. Heiner (US Bureau of Reclamation) is a means of attaching values to contour lines. The user adds an arc (transect) crosscutting the contours and defines a starting contour value and contour increment (or decrement). The tool progressively updates the contours intersected along the length of the transect. The tool was modified to disregard arcs which are not contours (eg faults, bounds). The tool is particularly useful with detailed contoured datasets with a constant contour interval.

Additional options controlling graphical display were also added to the MBTools interface. The *winsel* tool by E. Weitzman (ESRI) and L.A. Peltzlewis et al (USGS) allows the user to add or drop features from the selected set while repeatedly changing the map extent (zoom, pan, default etc). This is available from *Rove...* under the *Display* menu. Under the same menu is the *TipToe* tool by K.D. Cartier (USGS), which sequentially establishes mapextents and redraws the immediate vicinity of each element in the current selected set. This is useful in systematically examining small arcs which may have escaped the attributing process and are otherwise difficult to locate. A small overview icon display is available to position the selected feature in context with the whole coverage.

Figure 1: MBTools Menu Interface for IDSLEVEL and IGDSTEXT tools

The coverage attributes can be checked in a number of ways. A visual check using Arcplot or Arcview can be a quick and simple way of detecting gross errors. Some Arc commands are available to interrogate the info-based data; *codefind* identifies user-defined inconsistencies, *consist* identifies illogical coding combinations and *frequency* produces a list of unique code values and their frequency.

A visual check of the attributes can be achieved within an ArcEdit session by running ArcPlot routines. MBTools makes available from within the *General Draw Environment* menu the option of plotting a suite of these routines. A listing is provided by entering query (Button 3) under *Macro name.* This enables colour coding of attributes, colour fill, labelling of contour values, plotting of potential label errors, or the display of data emulating the published map, within the editing environment. To simplify display, arcs, nodes and labels should be turned off in the draw environment. Any corrections to item values will only be displayed by the macro after the coverage has been saved.

The concept of using ArcPlot within ArcEdit was further developed by integrating the ArcPlot routines within the *Options..* menu available in the *Draw Environment* menu. Figure 2 shows the enhancements made to the drawing options for arcs and labels, allowing the user to change symbol colour, label based on attributes, plot colour coded features or polygon colour fill. By giving the user greater flexibility in how the data is displayed, attribute errors can be better highlighted.

2.7) Transforming to Appropriate Coordinate System

At this stage, the coverage has topology and attributes. In many cases, digitising for the IGDS file has been done in a design plane with an arbitrary coordinate system. The resulting coverage will require transformation to the real world coordinate system (eg AMG) by *transform.* For the 1:250 000 map series, a template coverage was created using the appropriate AMG projection, from which tics were derived. This is the polygon coverage representing the 1:100 000 mapsheets over the area, generated and projected within Arc/Info. As well as the four mapsheet corner tics used in the transformation, the coverage also has additional tics corresponding to the constituent 1:100 000 scale maps. These are useful in registering the published 1:100 000 scale maps from which much of the digital data is sourced.

For the Murray Basin project, the coverages were deprojected into decimal degrees and appropriate coverages appended together to create basin-wide datasets.

In ArcTools, the *Transform* tool is available in the *Coordinates* category of the *Cover:Edit* toolpicker. A range of projection files are available within the *mb* work area, accessed by the pathway $MBP.

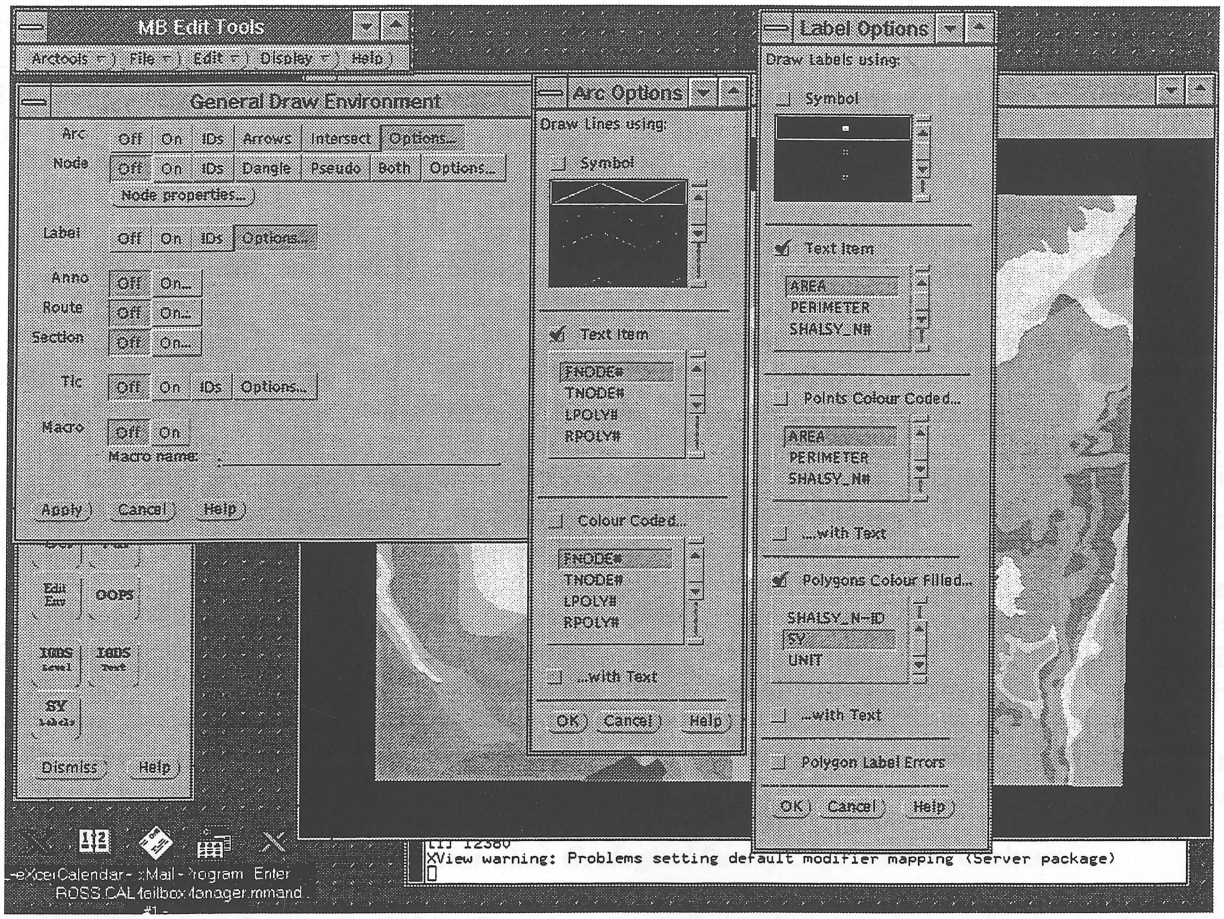

Figure 2: MBTools Menu Interface for Draw Environment Options

2.8) Validating and Documenting the Final Coverage

The final coverage is rebuilt and checked. To expedite validation, many of the Arc commands used to highlight potential errors were packaged into a single AML called *chkwork*. The user can check features of an entire workspace or an individual coverage, with output spooled to a file. Properties of the GIS coverage(s) checked by the AML include:

- double precision
- presence of edit masks
- correct topology
- appropriate annotation
- appropriate projection and coordinate system
- potential label, intersection or node errors
- correct item definitions in attribute tables
- potential errors in assigned attributes

Documentation for each coverage in the Murray Basin project simply uses a text file (Tucker et al, 1994). The file contents is based on the input requirements for the FINDAR directory system (Shelley, 1992) incorporating details on spatial identification, temporal coverage, currency, availability, lineage and quality. A standard documentation template (*cover_doc.std*) is available in the shared work area. Also available are templates defining the data items of particular coverages eg *sy_doc.std* for salinity/yield data items. These can simply be retrieved into the data items component of appropriate documentation files.

Within MBTools, the documentation file may be created, edited or shown from the *Document* tool in the *Coverage Manager* menu.

3) Conclusions

AML-based tools greatly enhance productivity and reduce data errors for the Murray Basin GIS project. The use of check and display routines makes the compliance of project standards for coverage names, item definitions and attribute coding easier to achieve. Some tools automate repetitious tasks, significantly reducing the conversion effort. These tools are easily made accessible to the user by being integrated into a modified version of ArcTools. The modular design of ArcTools allows modifications and additions to the interface to be easily made using a standard protocol.

4) Aknowledgments

Robyn Gallagher (GIS Solutions) and Gayle Young (AGSO CSU) were the reviewers for this record, and also provided insights in AML coding and the Intergraph system, respectively. Andrew Tucker (AGSO EGG) made available AML code developed by himself or modified versions of AMLs accessed from external ftp sites. Jenny Lane and Warren Overton have extensively tested the MBTools interface during ongoing conversion of the Murray Basin Hydrogeological Map Series into Arc/Info format. They also made comments on initial drafts of this Record.

5) References

Brodie, R. & Tucker, A.G., (1992) - Hydrogeological mapping and modelling using a GIS. in Geographic Information Systems, Cartographic and Geoscience Data Standards. Workshop Proceedings, Record 1992/27, *Bureau Mineral Resources*, Canberra.

ESRI, 1991a - ARC 6.0 Command References. *Environmental Systems Research Institute Inc* CA

ESRI, 1991b - ARC 6.0 Data Conversion *Environmental Systems Research Institute Inc* CA

Evans, W.R., 1992 - The Murray Basin Hydrogeological Map Series. *AWWA Water Journal* Vol 19, No 6, 20-23.

Gallagher, R., 1993 - Using ArcTools to build customised menu interfaces to Arc/Info. in *OZRI 7* Seventh Annual Australian Conference for ESRI and ERDAS Users. Melbourne

Hillier, J., (1992) - GIS and Digital Cartography - Output Options, Printing on Demand and Relative Pricing. in Geographic Information Systems, Cartographic and Geoscience Data Standards. Workshop Proceedings, Record 1992/27, *Bureau Mineral Resources*, Canberra.

Musto, I.P., 1990 - Project standards & system procedures for the use of geographical information systems. Unpubl. Report. *National Resource Information Centre*, Canberra

Shelley, P., 1992 - Access to data through FINDAR. in Geographic Information Systems, Cartographic and Geoscience Data Standards. Workshop Proceedings, Record 1992/27, *Bureau Mineral Resources*, Canberra.

Tucker, A.G. (1993) - Procedures to access point spatial and attribute data in an Oracle Database from within the Arc/Info GIS. Record 1993/73 *Australian Geological Survey Organisation*, Canberra

Tucker, A.G., Miller, P. & Brodie, R. (1994) - Procedures to configure shared work areas for the Arc/Info GIS. Record 1994/3 *Australian Geological Survey Organisation*, Canberra

Appendix 1) AMLs available under mb (Murray Basin) shared work area

As of 30/6/1994

Arc:

addit250k.aml:	adds items to attribute tables of 1:250 000 mapsheet coverage
bor2arc.aml:	makes an Arc/Info point coverage from coords held in RDBMS table (Tucker, 1993)
cgm16.aml:	converts .gra file into 16 bit version of CGM format for importing into Powerpoint
chkatt.aml:	outputs potential errors in attributes
chkitem.aml:	checks item definitions to project standards for 1:250 000 mapsheet coverages
chktoperr.aml:	outputs potential topological errors
chkwork.aml:	checks features of GIS coverage
dbleprec.aml:	copies coverage to double precision
docucov.aml:	initiates access to coverage documentation file
exportall.aml:	exports all Arc/Info geodata sets and files in current workspace
genhpgl.aml:	converts Arc plot file to HPGL format for 7595 plotter
gensht100.aml:	creates 100k mapsheet coverage (sheet100_p) for 250k workspace
rebuild.aml:	rebuilds coverages based on current topological features
splitsy.aml:	splits Salinity/Yield Coverages into components
statuscov.aml:	initiates access to status report for conversion process
updtic100.aml:	updates tics based on template SHEET100_p coverage

ArcEdit:

ap_*.aml:	suite of AP routines for plotting within ArcEdit
arcsy.aml:	updates salinity and yield items for salinity/yield covers based on label attributes
boreloc.aml:	creates Arc/Info point coverage from coords held in RDBMS table
digit9100.aml:	sets up 9100 digitiser (RM 475) defaults for 250k or 100k mapsheets
edtic_fat.aml:	sets up attribute tools appropriate for tics
edtic_fat_calc.aml:	table editor tool for calculating tic item values
edtic_snap.aml:	tool for tic snapping
igdslevel.aml:	calculate item values on basis of IGDS levels
igdstext.aml:	update label attributes based on label text on IGDS map
labelsy.aml:	create and attribute salinity/yield labels from IGDS map
laberr.aml:	sets up display of potential label errors in Arcedit
mb_transect.aml:	modified version of transect tool to attribute contours
oview.aml:	displays relative location of current mapextent to default mapextent
sel_int.aml:	selects arcs with intersection errors
tiptoe.aml:	sequentially draws elements of selected set
winsel.aml:	allows user to (un)select features while repeatedly changing mapextent

Appendix 2) The IGDSLEVEL Tool

Documentation (igdslevel.hlp):

Calculate Item Codes/Values based on IGDS-level Tool

In many IGDS files, different themes are stored on different levels. For example all highways on level 1, minor roads on level 2, tracks on level 3 etc. This means that attributes for the resulting coverage may be calculated on an IGDS-level basis.

The tool requires the coverage to be built appropriately (ie. the .AAT or .PAT file exists) and to have an appropriate item added to the attribute table.

The tool links the arcs to the IGDS information stored within the source .ACODE table, via the common $ID. Therefore, any previous changes to the $IDs of the arcs will lose this link and the tool will fail.

The tool requires the following inputs:

1. The source .ACODE file. This file is generated automatically from the initial IGDSARC conversion process and is prefixed by the name of the original coverage ie. cover.acode

2. The target item in the attribute table which is being updated

3. Up to ten codes which relate to a particular IGDS-level

For example, an arc coverage may have a STATUS item recording what the arc actually represents. Codes for STATUS are assigned on the basis of IGDS-level:

IGDS-Level	STATUS Code
1 (for structure contours)	C
10 (for aquifer boundaries)	A
20 (for flowlines)	F
etc	

[Calc] accepts settings and calculates values

[Cancel] reject settings and dismiss menu

[Help] get this information

AML Code (igdslevel.aml):

```
/*## calculates item values on basis of IGDS-level
/*-----------------------------------------------------------------------
/*        Australian Geological Survey Organisation
/*-----------------------------------------------------------------------
/*  Program: IGDSLEVEL.AML
/*  Purpose: Calculate item values on basis of IGDS-levels
/*
/*-----------------------------------------------------------------------
/*    Usage: IGDSLEVEL INIT {'position'} {'stripe'} {MODELESS | MODAL}
/*    Usage: IGDSLEVEL <routine> {}
/*
/* Arguments: routine  - name of the routine to be called.
/*        position - (quoted string) opening menu position.
/*        stripe   - (quoted string) menu stripe displayed.
/*        MODELESS | MODAL   - keyword for creating modal thread.
/*
/*  Globals: (none external)
/*        .igdslevel$cover   - edit coverage
/*        .igdslevel$ef      - edit feature
/*        .igdslevel$at      - type of attribute table (AAT or PAT)
/*        .igdslevel$acode   - source .ACODE table
/*        .igdslevel$item    - item in attribute table to be calculated
/*        .igdslevel$itemlength - defined length of item
/*        .igdslevel$itemtype - defined type of item
/*        .igdslevel$lev(n)  - nth IGDS-level (n = 1 to 10) used
/*        .igdslevel$val(n)  - value of item for nth IGDS-level
/*-----------------------------------------------------------------------
/*    Calls: igdslevel.menu, disp_help.aml, modal.aml
/*-----------------------------------------------------------------------
/*    Notes:
/*-----------------------------------------------------------------------
/*  History: Ross Brodie  -  04/12/93 - Original coding
/*=======================================================================
/*
&args routine arglist:rest
/*
&severity &error &routine bailout
/*
/* Check arguments
&if ^ [NULL %routine%] &then
 &call %routine%
&else
 &call INIT
&return

/*-----------
&routine INIT  /*  {'position'} {'stripe'} {MODELESS | MODAL}
/*-----------
/* set initial parameters
&set .igdslevel$cover = [SHOW EDIT]
/* check for null edit cover
&if [NULL %.igdslevel$cover%] &then
 &do
  &run msinform [QUOTE Need to define an edit cover]
  &call EXIT
  &return
 &end
&set .igdslevel$ef = [SHOW EDITFEATURE]
/* check for null edit feature
&if %.igdslevel$ef% = NONE &then
 &do
  &run msinform [QUOTE Need to define an edit feature]
  &call EXIT
  &return
 &end
/* ensure coverage has appropriate attribute tables
&if %.igdslevel$ef% = ARC &then &set .igdslevel$at = AAT
```

```
&if %.igdslevel$ef% = LABEL &then &set .igdslevel$at = PAT
&if not [EXISTS [ENTRYNAME %.igdslevel$cover%].%.igdslevel$at% -INFO] &then
  &do
    &run msinform [QUOTE Coverage needs to be built for %.igdslevel$ef%S]
    &call EXIT
    &return
  &end
/* initiate menu
&set position = [EXTRACT 1 [UNQUOTE %arglist%]]
&set stripe   = [EXTRACT 2 [UNQUOTE %arglist%]]
&set modality = [EXTRACT 3 [UNQUOTE %arglist%]]
&if [SHOW &thread &size [SHOW &thread &self]] = 0,0 &then
  &set launch = &thread &delete &self
&else
  &set launch
&if [NULL %position%] or %position%_ = #_ &then
  &set position = &cc &screen &cc
&if [NULL %stripe%] or %stripe%_ = #_ &then
  &set stripe = 'Calc on IGDS Levels'
&if [NULL %modality%] or %modality%_ = #_ &then
  &set .igdslevel$modal = .FALSE.
&else
  &do
    &if [TRANSLATE %modality%] = MODAL &then
      &set .igdslevel$modal = .TRUE.
    &else
      &set .igdslevel$modal = .FALSE.
  &end
&if [SHOW &thread &exists edit$igdslevel] &then
   &thread &delete edit$igdslevel
&thread &create edit$igdslevel ~
  &menu igdslevel.menu ~
  &position [UNQUOTE %position%] ~
  &stripe [QUOTE [UNQUOTE %stripe%]] ~
  &pinaction '&run igdslevel exit'
/*
&if %.igdslevel$modal% &then
  &run modal open edit$igdslevel
&else
  %launch%
&return

/* --------------
   &routine APPLY
/* --------------
/* calc values on basis of IGDS levels
/* extract definition of info item
&set .igdslevel$itemtype = [EXTRACT 3 [ITEMINFO %.igdslevel$cover%.%.igdslevel$at% -INFO ~
%.igdslevel$item%]]
&set .igdslevel$itemlength = [EXTRACT 2 [ITEMINFO %.igdslevel$cover%.%.igdslevel$at% -INFO ~
%.igdslevel$item%]]
/* establish relate between acode and attribute table
&data &arc
   RELATE ADD
   xx_igdslevel
   %.igdslevel$acode%
   info
   [entryname %.igdslevel$cover%]-ID
   [extract 1 [unquote [listitem %.igdslevel$acode% -info]]]
   linear
   ro
   [UNQUOTE ' ']
   RELATE SAVE XX_IGDSLEVEL.REL
   quit
&end
/* calc items on basis of levels
&severity &warning &ignore
RELATE RESTORE XX_IGDSLEVEL.REL
```

```
&do num = 1 &to 10 &by 1
 &if not [NULL [VALUE .igdslevel$lev%num%]] &then
  &do
   SEL xx_igdslevel//igds-level = [VALUE .igdslevel$lev%num%]
    &if [SHOW NUMBER SELECT] > 0 &then
     &do
      &if [KEYWORD %.igdslevel$itemtype% B I F N] > 0 and [TYPE [VALUE .igdslevel$val%num%]] > -1 ~
       &then &type Item %.igdlevel$item% requires a numeric value
      &if [KEYWORD %.igdslevel$itemtype% B I] > 0 and [TYPE [VALUE .igdslevel$val%num%]] = -2 ~
       &then &type Item %.igdslevel$item% requires an integer value
      &if [LENGTH [VALUE .igdslevel$val%num%]] > %.igdslevel$itemlength% ~
       &then &type Length of value exceeds defined item length
      &select %.igdslevel$itemtype%
       &when B,I,F,N
        CALC %.igdslevel$item% =  [VALUE .igdslevel$val%num%]
       &when C
        MOVEITEM [QUOTE [VALUE .igdslevel$val%num%]] TO %.igdslevel$item%
      &end
     &end
  &end
&end
&severity &error &routine bailout
&type \ Operation Completed
&call EXIT
&return

/* ---------------
  &routine CANCEL
/* ---------------
/* cancel addition/edit of.igdslevel
&call EXIT
&return

/* -------------
  &routine EXIT
/* -------------
/* remove variables and delete thread
/*
&if [EXISTS XX_IGDSLEVEL.REL -info] &then
  &set x = [DELETE XX_IGDSLEVEL.REL -INFO]
&if [VARIABLE .igdslevel$modal] &then
 &if %.igdslevel$modal% &then
  &do
   &dv .igdslevel$*
   &run modal close tool.igdslevel
  &end
&dv .igdslevel$*
&if [SHOW &thread &exists tool.igdslevel] &then
 &thread &delete tool.igdslevel
&return

/*----------------
&routine SAFETY_NET
/*----------------
&return
/*-----------
&routine HELP
/*-----------
&run disp_help igdslevel
&return
/*-------------
&routine USAGE
/*------------
&type Usage: IGDSLEVEL INIT {'"position"'} {"'stripe"'} {MODELESS I MODAL}
&type Usage: IGDSLEVEL <routine> {args}
&type Usage: IGDSLEVEL
&return &inform
/*--------------
```

```
&routine BAILOUT
/*-------------
&severity &error &ignore
&call EXIT
&return &warning An error has occurred in routine: %routine% (IGDSLEVEL.AML).
```

Menu (igdslevel.menu):

```
7
/*-----------------------------------------------------------------------
/*        Australian Geological Survey Organisation
/*-----------------------------------------------------------------------
/*   Menu: IGDSLEVEL.MENU
/* Purpose: Attribute contour lines based on IGDS levels tool interface.
/*-----------------------------------------------------------------------
/*  Calls: igdslevel.aml
/*-----------------------------------------------------------------------
/* Globals: .igdslevel$acode - source .ACODE table
/*        .igdslevel$item  - item in attribute table to calc
/*        .igdslevel$cover - edit coverage
/*        .igdslevel$at    - type of attribute table (AAT or PAT)
/*        .igdslevel$levn  - n IGDS-level
/*        .igdslevel$valn  - n value of attribute corresponding to n IGDS-level
/*-----------------------------------------------------------------------
/*  Notes:
/*-----------------------------------------------------------------------
/* History: Ross Brodie  -  20/02/94 - Original coding
/*=======================================================================
 Source File (.ACODE)
 %acode

 Target Item in Attribute Table
 Item: %item

  ^IGDS Levels ^Item Values
  %lev1    =  %val1
  %lev2    =  %val2
  %lev3    =  %val3
  %lev4    =  %val4
  %lev5    =  %val5
  %lev6    =  %val6
  %lev7    =  %val7
  %lev8    =  %val8
  %lev9    =  %val9
  %lev10   =  %val10


 %ok  %cancel  %help
%acode INPUT .igdslevel$acode 30 TYPEIN YES SCROLL NO ~
    REQUIRED ~
    HELP 'Specify name of source .ACODE file' ~
    FILE *.acode -INFO
%item INPUT .igdslevel$item 20 TYPEIN YES SCROLL NO ~
   REQUIRED~
   HELP 'Coverage item storing attribute' ~
   ITEM %.igdslevel$cover%.%.igdslevel$at% -INFO
%val1 INPUT .igdslevel$val1 10 TYPEIN YES SCROLL NO ~
   HELP 'First attribute value' ~
   CHARACTER
%lev1 INPUT .igdslevel$lev1 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to first value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val2 INPUT .igdslevel$val2 10 TYPEIN YES SCROLL NO ~
   HELP 'Second attribute value' ~
   CHARACTER
%lev2 INPUT .igdslevel$lev2 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to second value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val3 INPUT .igdslevel$val3 10 TYPEIN YES SCROLL NO ~
```

```
   HELP 'Third attribute value' ~
   CHARACTER
%lev3 INPUT .igdslevel$lev3 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to third value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val4 INPUT .igdslevel$val4 10 TYPEIN YES SCROLL NO ~
   HELP 'Fourth attribute value' ~
   CHARACTER
%lev4 INPUT .igdslevel$lev4 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to fourth value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val5 INPUT .igdslevel$val5 10 TYPEIN YES SCROLL NO ~
   HELP 'Fifth attribute value' ~
   CHARACTER
%lev5 INPUT .igdslevel$lev5 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to fifth value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val6 INPUT .igdslevel$val6 10 TYPEIN YES SCROLL NO ~
   HELP 'Sixth attribute value' ~
   CHARACTER
%lev6 INPUT .igdslevel$lev6 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to sixth value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val7 INPUT .igdslevel$val7 10 TYPEIN YES SCROLL NO ~
   HELP 'Seventh attribute value' ~
   CHARACTER
%lev7 INPUT .igdslevel$lev7 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to seventh value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val8 INPUT .igdslevel$val8 10 TYPEIN YES SCROLL NO ~
   HELP 'Eighth attribute value' ~
   CHARACTER
%lev8 INPUT .igdslevel$lev8 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to eighth value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val9 INPUT .igdslevel$val9 10 TYPEIN YES SCROLL NO ~
   HELP 'Ninth attribute value' ~
   CHARACTER
%lev9 INPUT .igdslevel$lev9 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to ninth value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%val10 INPUT .igdslevel$val10 10 TYPEIN YES SCROLL NO ~
   HELP 'tenth attribute value' ~
   CHARACTER
%lev10 INPUT .igdslevel$lev10 10 TYPEIN YES SCROLL NO ~
   HELP 'IGDS level corresponding to tenth value' ~
   UNIQUE %.igdslevel$acode% -info igds-level
%ok BUTTON ~
   HELP 'Apply settings and quit menu' ~
   Calc &run igdslevel apply;&return;
%cancel BUTTON CANCEL ~
   HELP 'Quit from this menu' ~
   'Cancel' &run igdslevel cancel; &return;
%help BUTTON RETURN KEEP ~
   HELP 'Get help about this menu' ~
   'Help' &run igdslevel help;
```

Appendix 3) The IGDSTEXT Tool

Documentation (igdstext.hlp):

Update feature values based on text labels on IGDS map Tool

This tool updates an item in the attribute table for features based on the text label depicted on the published map. For example, the contour value displayed on the map can be transferred to an appropriate item for the arc(s). The coverage should be of the same coordinate system as define by the IGDS file ie. it has not been transformed. The coverage also has to have been built for the relevant features and have the appropriate item (eg SPOT,SWL, TOP) added to the feature attribute table, to accommodate the text.

The tool requires the following input:

1. The name of the source IGDS file - the .dgn file from which the coverage was initially derived and where the text labels are stored

2. The level within the IGDS file where the relevant text labels reside. These elements should have been assigned to standard levels. The AML accesses the structure of the IGDS file and determines which levels have text elements within them.

3. There is an option to define the maximum width for the character string making up the text label (1-255). It most cases the default of 12 should be adequate.

4. The target item in the .AAT file where the text will reside. eg TOP, SWL, BASE, TOPO

5. The criteria used to select out the relevant features. For example, in many arc coverages, more than just contour lines are depicted. Faults, geological boundaries, flow lines may also be included. For the tool to work exclusively on the contours, there is an option to define a suitable selection criteria eg STATUS = 'C', $ID = 5, acode//igds-level = 5 etc. This criteria may be critical, particularly for arcs. In this case, text values may be incorrectly assigned to arcs other than contours.

6.The selection tolerance defining the maximum distance between the position of the text label and corresponding feature. By default, IGDS text is converted to anno.igds. This distance can be determined by displaying anno.igds with the feature. The tolerance can be set by entering a value, using the cursor or a default value.

[OK] accept settings and calculate values

[Cancel] reject settings and dissmiss menu

[Help] get this information

AML processing depends on the feature type. For arcs, the arc nearest to the text label is selected and arcs connected to and validated by the selection criteria are subsequently added. It is critical that arcs are properly connected (ie node errors corrected) and that the selection criteria is fulfilled. The success of the tool also depends largely on an adequate distribution of contour value labels in the original IGDS file. If not all contours are labelled on the map, further editing will be required.

For polygon labels, the current labels are replaced by labels matching the position of the text labels depicted on the map. These labels are attributed by the information stored by the IGDS text. The distribution of the text labels determines the success of the routine. Not all polygons may be labelled on the map or labels may be separated from the polygon by lead lines. The coverage will require checking for label errors - unlabelled polygons or polygons with more than one label.

AML Code (igdstext.aml):

```
/*## Update label attributes based on label text on IGDS map
/*-----------------------------------------------------------------------
/*        Australian Geological Survey Organisation
/*-----------------------------------------------------------------------
/*  Program: IGDSTEXT.AML
/*  Purpose: Update label values based on label text on IGDS map
/*
/*-----------------------------------------------------------------------
/*   Usage: igdstext INIT {'position'} {'stripe'} {MODELESS | MODAL}
/*   Usage: igdstext <routine> {args}
/*
/* Arguments: routine  - name of the routine to be called.
/*        position - (quoted string) opening menu position.
/*        stripe   - (quoted string) menu stripe displayed.
/*        MODELESS | MODAL   - keyword for creating modal thread.
/*
/*  Globals: (none external)
/*        .igdstext$cover          current edit coverage
/*        .igdstext$ef             current edit feature
/*        .igdstext$labelcover     temporary coverage with text labels
/*        .igdstext$featcover      temporary coverage with feature subset
/*        .igdstext$nearcover      coverage used in NEAR analysis
/*        .igdstext$type           feature type (LINE, POINT or POLYGON)
/*        .igdstext$at             attribute table (AAT or PAT)
/*        .igdstext$txtlev         levels available in IGDS file with text
/*        .igdstext$select         selection criteria for features
/*        .igdstext$dgn            source IGDS file
/*        .igdstext$level          text level used for updating
/*        .igdstext$textwidth      maximum length of IGDS text
/*        .igdstext$item           target item in .aat
/*        .igdstext$itemtype       type of item (B,I,F,C,D)
/*        .igdstext$itemlength     defined length of item
/*        .igdstext$next           used to extract info from IGDS file
/*        .igdstext$num            used to extract info from IGDS file
/*        .igdstext$lev(1-n)       available levels in IGDS file
/*        .igdstext$type(1-n)      element type for level n
/*        .igdstext$message        error message
/*        .igdstext$hash           id of feature (#)
/*        .igdstext$value          attribute value for feature
/*-----------------------------------------------------------------------
/*   Calls: igdstext.menu, disp_help.aml, modal.aml
/*-----------------------------------------------------------------------
/*   Notes:
/*-----------------------------------------------------------------------
/*  History: Ross Brodie  -  13/02/94 - Original coding
/*        routines DGN_INFO, READ_INFO based on igdsarcc.aml
/*=======================================================================
/*
&args routine arglist:rest
/*
&severity &error &routine bailout
/*
/* Check arguments
&if ^ [NULL %routine%] &then
  &call %routine%
&else
  &call INIT
&return

/*-----------
&routine INIT   /*  {'position'} {'stripe'} {MODELESS | MODAL}
/*-----------
/* set initial parameters
&set .igdstext$cover = [SHOW EDIT]
&set .igdstext$ef
&set .igdstext$labelcover
```

```
 &set .igdstext$featcover
 &set .igdstext$txtlev
 &set .igdstext$select = :
 /* check null edit cover
 &if [NULL %.igdstext$cover%] &then
  &do
   &run msinform [QUOTE Need to define an EDIT COVER]
   &call EXIT
   &return
  &end
 &set .igdstext$ef = [SHOW EDITFEATURE]
 /* Check null edit feature
 &if %.igdstext$ef% = NONE &then
  &do
   &run msinform [QUOTE Need to define an EDIT FEATURE]
   &call EXIT
   &return
  &end
 /* set parameters based on edit feature
 &if %.igdstext$ef% = ARC &then
  &do
   &set .igdstext$at = AAT
   &set .igdstext$type = LINE
  &end
 &if %.igdstext$ef% = LABEL &then
  &do
   &set .igdstext$at = PAT
   &describe %.igdstext$cover%
    &if %dsc$qtopology% &then
     &do
      &set .igdstext$type = POLYGON
      &run msconfirm init .igdstext$confirm ~
      'This tool replaces current polygon labels with ones on map' ~
      'Do you want to continue...'
      &if not %.igdstext$confirm% &then
       &do
        &call EXIT
       &return
       &end
     &end
    &else &set .igdstext$type = POINT
  &end
 /* Check attribute table
 &if not [EXISTS [ENTRYNAME %.igdstext$cover%].%.igdstext$at% -INFO] &then
  &do
   &run msinform [QUOTE Coverage needs to be built for %.igdstext$ef%S ]
   &call EXIT
   &return
  &end
 &set .igdstext$labelcover = XXL_[TRANSLATE [ENTRYNAME %.igdstext$cover%]]
 &set .igdstext$featcover = XXF_[TRANSLATE [ENTRYNAME %.igdstext$cover%]]
 /* initiate menu
 &set position = [EXTRACT 1 [UNQUOTE %arglist%]]
 &set stripe   = [EXTRACT 2 [UNQUOTE %arglist%]]
 &set modality = [EXTRACT 3 [UNQUOTE %arglist%]]
 &if [SHOW &thread &size [SHOW &thread &self]] = 0,0 &then
  &set launch = &thread &delete &self
 &else
  &set launch
 &if [NULL %position%] or %position%_ = #_ &then
  &set position = &cc &screen &cc
 &if [NULL %stripe%] or %stripe%_ = #_ &then
  &set stripe = Calc on IGDS Text
 &if [NULL %modality%] or %modality%_ = #_ &then
  &set .igdstext$modal = .TRUE.
 &else
  &do
   &if [TRANSLATE %modality%] = MODAL &then
```

```
      &set .igdstext$modal = .TRUE.
     &else
      &set .igdstext$modal = .FALSE.
    &end
  &if [SHOW &thread &exists tool$igdstext] &then
    &thread &delete tool$igdstext
  &thread &create tool$igdstext ~
   &menu igdstext.menu ~
   &position [UNQUOTE %position%] ~
   &stripe [QUOTE [UNQUOTE %stripe%]] ~
   &pinaction '&run igdstext exit'
  /*
  &if %.igdstext$modal% &then
   &run modal open tool$igdstext
  &else
   %launch%
  &return
  /*------------------
  &routine GET_DGN
  /*------------------
  /* get source design file
  &run getfile init .igdstext$tmp_dgn *.dgn '#' 'Select IGDS Design File'
  &if ^ [NULL %.igdstext$tmp_dgn%] &then
   &do
    &set .igdstext$dgn =  %.igdstext$tmp_dgn%
    &call DGN_INFO
    &end
   &else
    &set  .igdstext$dgn :=
  &return
  /*-------------------
  &routine DGN_INFO
  /*-------------------
  /* access info from IGDS file
  /* Make sure input file exists
  &if ^ [EXISTS %.igdstext$dgn% -FILE] &then
   &do
    &set .igdstext$message %.igdstext$dgn% does not exist
    &set .igdstext$dgn :=
   &end
  &run msworking init 'Gathering design file information' 'Please Wait'
  &data arc
   ~&messages &off
   ~&term 9999
   ~&dalines 9999
   ~&watch xx_igdstext.wat
   IGDSINFO  %.igdstext$dgn%
   ~&watch &off
   quit
  &end
  &call READ_INFO
  &run msworking close
  &return
  /*-------------------
  &routine READ_INFO
  /*-------------------
  /* extract levels with text elements
  /* open igdsinfo file
  &set file xx_igdstext.wat
  &if not [EXISTS %file%] &then
   &do
    &type File %file% does not exist
    &call EXIT
    &return
   &end
  &set .igdstext$next = .FALSE.
  &set amlunit = [OPEN %file% ostat -READ]
  &if %ostat% NE 0 &then
```

```
  &do
   &type Error opening file %file%, error code: %ostat%
   &call EXIT
   &return
  &end
 /* read a line
 &set rec = [AFTER [READ %amlunit% rstat] !]
 &do &until %rstat% ne 0
  &if %.igdstext$next% &then
   &do
    &do i = 1 &to %.igdstext$num% &by 1
      &set .igdstext$lev%i% = [EXTRACT 1 [UNQUOTE %rec%]]
      &set .igdstext$type%i%  = [EXTRACT 5 [UNQUOTE %rec%]]
       &if [VALUE .igdstext$type%i%] = 17 and ~
       [QUOTE %.igdstext$txtlev%] not cn [QUOTE [VALUE .igdstext$lev%i%]] &then
       &set .igdstext$txtlev = [UNQUOTE [QUOTE [UNQUOTE %.igdstext$txtlev%] [VALUE .igdstext$lev%i%]]]
      /* read a line
      &set rec [AFTER [READ %amlunit% rstat] !]
    &end
    &set .igdstext$next .FALSE.
   &end
  &if %rec% cn 'unique' &then
   &set .igdstext$num = [UNQUOTE [BEFORE [AFTER [QUOTE %rec%] '] unique ]]
  &if %rec% cn '-----' &then
   &set .igdstext$next .TRUE.
  &set rec [AFTER [read %amlunit% rstat] !]
 &end
 &set cstat [CLOSE -ALL]
 &return
 /*-------------------
 &routine CHK_SEL
 /*-----------------
 /* check that select criteria selects
 &severity &error &ignore
 UNSEL ALL
 &if not [NULL [UNQUOTE [AFTER [QUOTE %.igdstext$select%] :]]] &then
  &do
  SEL [UNQUOTE [AFTER [QUOTE %.igdstext$select%] :]]
   &if [SHOW NUMBER SELECT] = 0 &then
    &do
     &type Select Criteria does not select any features
     &set .igdstext$select = :
   &end
  &end
 &severity &error &routine bailout
 &return
 /* -----------------
 &routine CURSOR
 /*-------------
 /* set selection tolerance by cursor
 /* do this by edit distance
 &set old_ed = [SHOW EDITDISTANCE]
 EDITDISTANCE
 &set .igdstext$neartol = [SHOW EDITDISTANCE]
 EDITDISTANCE %old_ed%
 &return
 /*---------------
 &routine DEFAULT
 /*---------------
 /* default near tolerance
 &set .igdstext$neartol = #
 &return
 /*-------------------
 &routine APPLY
 /*-------------------
 &set .igdstext$itemtype =  [EXTRACT 3 [ITEMINFO %.igdstext$cover%  -%.igdstext$type% %.igdstext$item%]]
 &set .igdstext$itemlength =  [EXTRACT 2 [ITEMINFO %.igdstext$cover% -%.igdstext$type% %.igdstext$item%]]
 /* create coverage containing just text labels from IGDS file
```

```
 /* do this in ARC
 &data arc
  IGDSARC %.igdstext$dgn% %.igdstext$labelcover% ~
  # # # # # # %.igdstext$textwidth%
  LABEL %.igdstext$level% * * * * TEXTPOINT
  END
  Y
  quit
 &end
 /* build igds coverage for points
 &data arc
  BUILD %.igdstext$labelcover% POINTS
  quit
 &end
 /* establish relates between igds point.pat,  .acode and feature attributes
 &data &arc
  RELATE ADD
  igds
  %.igdstext$labelcover%.acode
  info
  %.igdstext$labelcover%-id
  %.igdstext$labelcover%-id
  linear
  ro
  feat
  %.igdstext$labelcover%.acode
  info
  [ENTRYNAME %.igdstext$cover%]-id
  %.igdstext$labelcover%-id
  linear
  ro
  [UNQUOTE ''']
 RELATE SAVE XX_IGDSTEXT.REL
  quit
&end
&if %.igdstext$type% = POLYGON &then
 &do
  &call CALC_POLYGON
  &call EXIT
  &type \ Operation Completed
  &return
 &end
&else
 &do
  &call NEAR
  &call CALC_%.igdstext$type%
  &call EXIT
  &type \ Operation Completed
  &return
 &end
&return
/*-------------------
&routine NEAR
/*-------------------
/* determine linkage between text labels and nearest feature
/* do this with NEAR
&if not [NULL [UNQUOTE [AFTER [QUOTE %.igdstext$select%] :]]] &then
 &do
  SELECT [UNQUOTE [AFTER [QUOTE %.igdstext$select%] :]]
  PUT %.igdstext$featcover%
  &data &arc
   BUILD %.igdstext$featcover% %.igdstext$type%
   quit
  &end
 &end
/* perform near analysis between IGDS text labels and coverage features
&if [EXISTS %.igdstext$featcover% -COVER] &then
 &set .igdstext$nearcover = %.igdstext$featcover%
```

```
 &else &set .igdstext$nearcover = %.igdstext$cover%
 &data &arc
   NEAR  %.igdstext$labelcover% %.igdstext$nearcover% ~
   %.igdstext$type% %.igdstext$neartol%
   quit
 &end
 &return
 /*------------------
  &routine CALC_LINE
 /*------------------
 /* deal with contours only
 /* for each IGDS text label get appropriate arc
 EC  %.igdstext$labelcover%
 EF LABELS
 /* set maximum number of point values
 RELATE RESTORE XX_IGDSTEXT.REL
 SEL ALL
 &set max = [SHOW NUMBER SELECT]
 &do num = 1 &to %max% &by 1
   SELECT %.igdstext$labelcover%# = %num%
   /* find id of nearest arc
   &set .igdstext$hash = [SHOW LABEL %num% ITEM [ENTRYNAME %.igdstext$nearcover%]#]
   /* find contour value
   &set .igdstext$value = [SHOW LABEL %num% item igds//igds-text]
   /* go to contour coverage and assign to appropriate arc(s)
   EC %.igdstext$cover%
   EF ARCS
   /* select the nearest arc
   SELECT [ENTRYNAME %.igdstext$cover%]# =  %.igdstext$hash%
   /* add the connecting arcs that are contours
   &do &until %sel_done%
     &set old_selnum = [SHOW NUMBER SELECT]
     ASELECT CONNECT
     /* remove arcs that are not contours
     &if not  [NULL [UNQUOTE [AFTER [QUOTE %.igdstext$select%] :]]] &then
       RESELECT [UNQUOTE [AFTER [QUOTE %.igdstext$select%] :]]
     &set new_selnum = [SHOW NUMBER SELECT]
     &if %new_selnum% = %old_selnum% &then
       &set sel_done = .TRUE.
     &else &set sel_done = .FALSE.
   &end
/* calculate contour value
   &if [SHOW NUMBER SELECT] > 0 &then
    &call CALC_VALUE
   EC  %.igdstext$labelcover%
   EF LABELS
 &end
 &return
 /*------------------
  &routine CALC_POINT
 /*------------------
 /* deals with point features only
 /* for each IGDS text label get appropriate coverage point
 EC  %.igdstext$labelcover%
 EF LABELS
 /* set maximum number of point values
 RELATE RESTORE XX_IGDSTEXT.REL
 SEL ALL
 &set max = [SHOW NUMBER SELECT]
 &do num = 1 &to %max% &by 1
   SELECT %.igdstext$labelcover%# = %num%
   /* find id of nearest coverage point
   &set .igdstext$hash = [SHOW LABEL %num% ITEM ~
   [ENTRYNAME %.igdstext$nearcover%]#]
   /* find point value
   &set .igdstext$value = [SHOW LABEL %num% item igds//igds-text]
   /* go to point coverage and assign to appropriate points
   EC %.igdstext$cover%
```

```
    EF LABELS
    /* select the nearest point
    SELECT [ENTRYNAME %.igdstext$cover%]# =  %.igdstext$hash%
    /* calculate contour value
    &if [SHOW NUMBER SELECT] > 0 &then
    &call CALC_VALUE
    EC  %.igdstext$labelcover%
    EF LABELS
  &end
  &return
  /*-------------------
  &routine CALC_POLYGON
  /*-------------------
  /* deals with polygon labels only
  /* replace current labels with IGDS text labels
  SELECT ALL
  DELETE
  GET %.igdstext$labelcover%
  /* update attribute
  RELATE RESTORE XX_IGDSTEXT.REL
  SEL ALL
  &set max = [SHOW NUMBER SELECT]
  &do num = 1 &to %max% &by 1
   &set .igdstext$value = [SHOW LABEL %num% item feat//igds-text]
   &call CALC_VALUE
  &end
  &return
  /*----------------
  &routine CALC_VALUE
  /*----------------
  &severity &error &ignore
  &if [KEYWORD %.igdstext$itemtype% B I F N] > 0 and [TYPE [VALUE .igdstext$value]] > -1 ~
  &then &type Item %.igdstext$item% requires a numeric value
  &if [KEYWORD %.igdstext$itemtype% B I] > 0 and [TYPE [VALUE .igdstext$value]] = -2 ~
  &then &type Item %.igdstext$item% requires an integer value
  &if [LENGTH %.igdstext$value%] > %.igdstext$itemlength% ~
  &then &type Length of value exceeds defined item length
  &select %.igdstext$itemtype%
   &when B,I,F,N
    CALC %.igdstext$item% =  %.igdstext$value%
   &when C
    MOVEITEM [QUOTE [VALUE %.igdstext$value%]] TO %.igdstext$item%
  &end
  &type Processed %num% of %max% text elements
  &return
  /*----------
  &routine HELP
  /*----------
  &run disp_help igdstext
  &return

  /*----------
  &routine USAGE
  /*----------
  &type Usage: igdstext INIT {'"position"'} {'"stripe"'} {MODELESS I MODAL}
  &type Usage: igdstext <routine> {args}
  &type Usage: igdstext
  &return &inform

  /*----------
  &routine EXIT
  /*----------
  &severity &warning &ignore
  &if [QUOTE [SHOW EDIT]]  not = [QUOTE %.igdstext$cover%] &then
   EC  %.igdstext$cover%
  &if [NULL %.igdstext$ef%] &then &set .igdstext$ef = NONE
  &if [QUOTE [SHOW EDITFEATURE]] not = [QUOTE %.igdstext$ef%] &then
   EF %.igdstext$ef%
```

```
&if [EXISTS %.igdstext$labelcover% -COVER] &then
  &sys arc KILL  %.igdstext$labelcover% ALL
&if [EXISTS %.igdstext$labelcover%.ACODE -INFO] &then
  &set x = [DELETE  %.igdstext$labelcover%.acode -INFO]
&if [EXISTS %.igdstext$featcover% -COVER] &then
  &sys arc KILL  %.igdstext$featcover% ALL
&if [EXISTS xx_igdstext.wat -FILE] &then
  &set x = [DELETE  xx_igdstext.wat -FILE]
&if [EXISTS XX_IGDSTEXT.REL -INFO] &then
  &set x = [DELETE   XX_IGDSARC.REL -INFO]
&if [VARIABLE .igdstext$modal] &then
  &if %.igdstext$modal% &then
    &do
      &dv .igdstext$*
      &run modal close tool$igdstext
    &end
&dv .igdstext$*
&if [SHOW &thread &exists tool$igdstext] &then
  &thread &delete tool$igdstext
&return

/*--------------
&routine BAILOUT
/*--------------
&severity &error &ignore
&severity &warning &ignore
&call EXIT
&return &warning An error has occurred in routine: %routine% (igdstext.AML).
```

Menu (igdstext.menu):

```
7
/*-----------------------------------------------------------------------
/*        Australian Geological Survey Organisation
/*-----------------------------------------------------------------------
/*   Menu: IGDSTEXT.MENU
/* Purpose: Update attributes based on IGDS text tool interface.
/*-----------------------------------------------------------------------
/*   Calls: igdstext.aml
/*-----------------------------------------------------------------------
/* Globals: .igdstext$*
/*-----------------------------------------------------------------------
/*   Notes:
/*-----------------------------------------------------------------------
/* History: Ross Brodie  -  20/02/94 - Original coding
/*=======================================================================
Source IGDS file:
%dgn

IGDS-level of Map Text:
%lev

Maximum Text Width:
%tw

Target Item
%item

Selection Criteria:
%sel

Selection Tolerance:
%stol        %cur %def


%ok  %cancel  %help
```

```
%dgn INPUT .igdstext$dgn 30 TYPEIN YES SCROLL NO ~
   REQUIRED ~
   HELP 'IGDS file containing text elements' ~
   QUERY '&run igdstext get_dgn' ~
   RETURN '&run igdstext dgn_info' ~
   FILE *.dgn
%lev INPUT .igdstext$level 10 TYPEIN YES SCROLL NO ~
   REQUIRED ~
   HELP 'IGDS level where text is stored' ~
   CHOICE %.igdstext$txtlev% -PROMPT 'Text Levels'
%tw SLIDER .igdstext$textwidth 30 ~
   INITIAL 12 ~
   HELP 'Select width of label text' ~
   step 1 integer 1 255
%item INPUT .igdstext$item 30 TYPEIN YES SCROLL NO ~
   HELP 'Coverage item storing text' ~
   REQUIRED ~
   ITEM %.igdstext$cover%.%.igdstext$at% -INFO
%sel INPUT .igdstext$select 30 ~
   RETURN '&run igdstext chk_sel' ~
   HELP 'Criteria to select out appropriate features only' ~
   CHARACTER
%stol INPUT .igdstext$neartol 12 ~
   HELP 'Set tolerance between feature and text label' ~
   REAL
%cur BUTTON RETURN ~
   HELP 'Define distance using cursor' ~
   ICON crshair16.icon * ~
   &run igdstext cursor
%def BUTTON RETURN ~
   HELP 'Use Default tolerance' ~
   'Default' ~
   &run igdstext default
%ok BUTTON ~
   HELP 'Apply settings and quit menu' ~
   OK &return; &run igdstext apply
%cancel BUTTON CANCEL ~
   HELP 'Quit from this menu' ~
   'Cancel' &return; &run igdstext exit
%help BUTTON RETURN KEEP ~
   HELP 'Get help about this menu' ~
   'Help' &run igdstext help
```