NAME

gen_rng - Generic Range File

DESCRIPTION

The generic range file, gen rng, resides in an appropriate /type?? directory for each switching machine (SPCS) for which any one of the following SCCS features are supported:

RC:BUILD
Scheduled Common Analysis
Demand Common Analysis
3B Common Processor Features

The generic range file contains a number of entries that identify one or more groups of routines to be executed for each of the above features. Which group of routines should be executed is determined by such things as the feature to be performed, the feature function to be performed, and the series of SPCS generics and issues that is pertinent to the office for which the requested task is being performed.

Each generic range file entry has a fixed size and has the structure **GEN_RNG**, as defined in the header file, gen rng.h. All information in the entry is in ASCII; hence all elements are defined as character strings. Each entry must be initialized to contain blanks in all elements or unused portions of elements that do not contain data. Data in an entry is left-justified in each element.

The elements **gr_fgen** and **gr_tgen** specify the "from" and "to" generic ID's that are to be used for range checking. **Gr_fgen** specifies the lower bound and must always contain either an entire generic ID or the first few digits of the generic ID that identify the generic base. **Gr_tgen** specifies the upper bound and may contain an entire generic ID or the first few digits of the generic ID that identify the generic base. **Gr_tgen** may also contain a '-' to indicate that all generic ID's that are greater than or equal to **gr_fgen** are to be accepted. Note that if the value specified for **gr_fgen** contains only a generic base, then the value for **gr_tgen** must also contain only a generic base or a '-'.

The elements **gr_fiss** and **gr_tiss** specify the "from" and "to" abstract issue numbers that are to be used for range checking. These elements normally contain the value '-' except when it becomes necessary to perform range checking on an issue basis rather than on a generic basis. In such cases, the element **gr_fiss** identifies the "from" abstract issue number that specifies the lower bound for the range checking and the element **gr_tiss** identifies the "to" abstract issue number that specifies the upper bound. A '-' entry for **gr_tiss** means that all abstract issue numbers for the indicated generic ID that are "greater than or equal to" **gr_fiss** are to be accepted.

If it becomes necessary to perform range checking on an issue basis for a certain feature and function, the following steps must be followed:

1. New entries must be inserted into the generic range file for the affected feature, function, and generic ID. These new entries must specify the appropriate range of abstract issue numbers that are served by the routines specified in the generic range file entry.

2. Be certain to remove old entries, that pertain to the affected feature and function, from the generic range file.

The following is a listing of the _gen_rng.h_ header file.

```
/*
    This header file defines the structure for the "gen_rng" file
    presently used by RC:BUILD and COMMON ANALYSIS distributor
    routines to determine which routines must be executed to per-
    form the desired functions.  The programs to be executed are
    determined by the office generic and issue.
*/


/*
    Define the name of the generic range file.
*/

#define GEN_RNG_FIL "gen_rng"


/*
    Define supported features.
*/

#define GR_RCBLD      "rcb"    /* RC:BUILD */
#define GR_SCA        "sca"    /* Scheduled COMMON ANALYSIS Routines */
#define GR_DCA        "dca"    /* Demand COMMON ANALYSIS Routines */


/*
    Define supported functions for the above features.
*/

#define GR_SPA        "spa"    /* SPA - Switched Path Analysis */
#define GR_ECA        "eca"    /* ECA - External Circuit Analysis */
#define GR_TRK        "trk"    /* TRK - TRK Analysis */
#define GR_NCA        "nca"    /* NCA - Network Controller Analysis */
#define GR_SDA        "sda"    /* SDA - Signal Distributor Analysis */
#define GR_PPA        "ppa"    /* PPA - Pulse Path Analysis */
#define GR_AHA        "aha"    /* AHA - Audit History Analysis */
```

```
/*
    Define return codes for library subroutine, GEN_RNG().
*/

#define GRR_ERR -1    /*
                          An error has been detected.
                      */

#define GRR_ENF  0    /*
                          The requested record has not been
                          found in the generic range file.
                      */


/*
    Define "open bound" or "don't care" indicator.
*/

#define DONT_CARE '-'


/*
    Specify ending sequence and size of each structure element.
*/

#define GR_ENDSEQ  "*0

#define GR_FEATSZ  4
#define GR_FUNCSZ  6
#define GR_GENSZ   6
#define GR_ISSNOSZ 4
#define GR_MXPGM   4
#define GR_MXNAMSZ 12
#define GR_ENDSZ   2


/* Define a union for a record in the GEN_RNG_FIL file */

union GR_REC
{
    char *gr_recptr;    /* Pointer to start of record */
    struct GEN_RNG *gr_rec;    /* Pointer to generic range record */
};



/*
    Define the structure of a "generic range" record.
*/

struct GEN_RNG
{
        char gr_feat[GR_FEATSZ];
                                /*
                                   Feature to which record applies.  See
                                   Note 1.
                                */

        char gr_func[GR_FUNCSZ];
                                /*
                                   Identifies which function of the
                                   feature is to be performed.  See
                                   Note 2.
                                */

        char gr_fgen[GR_GENSZ];
                                /*
                                   Identifies the "from" generic ID
                                   (eg 10, 100, 101).  See Note 3.
                                */

        char gr_fiss[GR_ISSNOSZ];
                                /*
                                   If needed, identifies the "from"
                                   abstract issue number (eg -, 010, 081,
                                   101).  See Note 4.
                                */

        char gr_tgen[GR_GENSZ];
                                /*
                                   Identifies the "to" generic ID
                                   (eg -, 10, 100, 101).  See Note 3.
                                */

        char gr_tiss[GR_ISSNOSZ];
                                /*
                                   If needed, identifies the "to"
                                   abstract issue number (eg -, 010, 081,
                                   101).  See Note 4.
                                */

        char gr_pgms[GR_MXPGM][GR_MXNAMSZ];
                                /*
                                   A list of up to GR_MXPGM routine names
                                   that are to be executed.  See Notes 5
                                   and 6.
                                */

        char gr_end[GR_ENDSZ];
                                /*
                                   Record ending sequence.
                                */
    };
```

```
/*
    Declare the value returned by the subroutine, gen_rng().
*/

char *gen_rng();


/*
    Notes:

    1.  Supported features are defined near the beginning of
        this file.

    2.  Supported functions are defined near the beginning of
        this file.

    3.  The elements gr_fgen and gr_tgen specify the "from" and
        "to" generic ID's that are to be used for range checking.
        Gr_fgen specifies the lower bound and must always contain
        either an entire generic ID or the first few digits of the
        generic ID that identify the generic base.  Gr_tgen speci-
        fies the upper bound and may contain a '-' to indicate an
        open upper bound or may contain an entire generic ID or
        the first few digits of the generic ID that identify the
        generic base.  If the value specified for gr_fgen con-
        tains only a generic base, then the value for gr_tgen
        must also contain only a generic base or a '-'.  The
        value specified for either of these elements must be
        left-justified in the appropriate field and padded on
        the right with blanks.

    4.  The elements gr_fiss and gr_tiss specify the "from" and
        "to" abstract issue numbers that are to be used for range
        checking.  These elements should contain the value '-'
        except when it becomes necessary to perform range checking
        on an issue basis rather than just on a generic basis.
        When it does become necessary to perform range checking on
        an issue basis, the element gr_fiss must contain the "from"
        abstract issue number (abstract issue numbers are defined
        in the header file, chldata.h) which specifies the lower
        bound for the range checking.  The element gr_tiss must
        contain the "to" abstract issue number which specifies the
        upper bound for the range checking.  Gr_tiss may contain
        the value '-' as an indication that all abstract issue
        numbers greater than or equal to gr_fiss are to be accepted.
        The value specified for either of these elements must be
        left-justified in the appropriate field and padded on the
        right with blanks.

    5.  Routine names should be of the form:

                aaattbbbbbs
```

where

aaa	contains two or three characters that identify the feature to be performed, such as "rcb" for RC:BUILD and "sca" or "sa" for SCHEDULED ANALYSIS.
tt	is the office type, such as 01 for No. 1 ESS.
bbbbb	contains up to five characters that identify which major phase of the feature is to be performed by this routine, such as "swrf" for SPA reformatting.
s	is a sequence or series code; eg. 'a', 'b', etc., that distinquishes this routine from other routines performing a similiar function for other groups or series of generics and issues.

The program name must be left-justified in the appropriate field, ie. no leading blanks, and all unused characters to the right of the routine name must be filled with blanks.

6. If less than GR_MXPGM routines are required for this feature, then all unused elements of the array, gr_pgms, must be filled with GR_MXNAMSZ blanks. The elements of this array that are needed for each of the supported features, however, must be filled as follows:

FEATURE	GR_PGM[0]	GR_PGM[1]	GR_PGM[2]	GR_PGM[3]
rcb	RCB Main	Unused	Unused	Unused
sca	Analysis Phase	Reformatting Phase	Pre-Analysis Phase	Unused
dca	Analysis Phase	Reformatting Phase	Pre-Analysis Phase	Unused

*/

FILES

/type??/gen_rng	Data File
/usr/include/gen_rng.h	Header File