MASSACHUSETTS INSTITUTE OF TECHNOLOGY

19

Memo to Room Ext.

SPS sections
for dynamo

from Room Ext.

Not needed

dynamo

The module dynamo is entered as the first module of the dynamo command. It processes the argument list, sets up the option flags, and routes control through the remaining compiler modules based on these options, and the success of the compilation. It obtains temporary storage segments required by the compiler, and releases them when processing is finished.

Entry

dynamo

[illegible]

[illegible]

Usage

dynamo path -options-

| | |
|---|---|
| path | is the pathname of the segment to be compiled. |

-options-

| | |
|---|---|
| source | a source listing is produced |
| symbols | an attribute and cross reference listing and a source listing is produced |
| list | a source listing, attribute listing, and assembly listing is produced. |
| severity2 | only errors of severity 2 and higher are listed on the user's console |
| brief | error listing on the console is in brief format. |
| assembly | an assembly listing is produced |
| check | the dynamo model is not executed. |

dyn_sort_

This entry reorders the initial equation matrix produced by the semantic translator into the order in which the equations will be compiled by examining the usage of variables on the left and right side of the equations.

Usage

```
        declare dyn_sort_ entry;
        call dyn_sort_;
```

## dyn_trans_

The module dyn_trans_ performs syntactic analysis for the dynamo compiler. It also contains the code necessary to perform semantic translation; that is, it creates the initial equation matrix, which is subsequently reordered by dyn_sort_ into the order in which the equations will be compiled.

### Usage

```
        declare dyn_trans_ entry;

        call dyn_trans_;
```

dyn_lex_

The module dyn_lex_ performs lexical analysis for the dynamo compiler. It is called from dyn_trans_ with no arguments. When called, it adds one token to the parser's stack.

Entry

```
        dyn_lex_$setup
```

Performs setup and initialization of internal variables prior to beginning lexical analysis.

Usage

```
        declare dyn_lex_$setup entry;

        call dyn_lex_$setup;
```

Entry

```
        dyn_lex_
```

Performs lexical analysis for the dynamo compiler.

Usage

```
        declare dyn_lex_ entry;

        call dyn_lex_;
```

dyn_odump_

Where

The module dyn_odump_ provides a ~~xxxxx~~formatted listing of the order in which equations in the dynamo model will be executed. The listing is produced by variable type and contains the equation numbers in the order in which they will be compiled.

Give format

Entry

    dyn_odump_

Usage

    declare dyn_odump_ entry;

    call dyn_odump_;

## dyn_sdump_

The module dyn_sdump_ produces a formatted listing of all symbols defined in the compilation, their attributes, and a list of statement numbers where they have been used. It writes its output into the segment path.list, where path is the pathname of the segment being compiled. It is called when the compiler options symbols or list have been specified.

Entry

```
        dyn_sdump_
```

Usage

```
        declare dyn_sdump_ entry;

        call dyn_sdump_;
```

## dyn_codegen_

The module dyn_codegen_ is the code generation routine for the dynamo compiler. It generates 645 machine language into the segment dynamo_object_ in the process directory, and if the compiler options list or assembly have been specified, writes an assembly listing into the segment path.list, where path is the pathname of the segment being compiled.

### Entry

dyn_codegen_

### Usage

```
declare dyn_codegen_ entry;

call dyn_codegen_;
```

dyn_alm_

The module dyn_alm_ is an ALM program used as a transfer vector for external calls by the compiled dynamo program. It is called through a pointer stored in the compiled program by the code generator, with index register 5 containing the offset within the transfer vector for the desired external call. It is intended to be used only as an internal interface to the compiled dynamo program.

Entry

```
        dyn_alm_$dyn_tv_
```

Usage

```
        lxl5      offset,dl
        call      <dyn_alm_>/dyn_tv_
```

## dyn_c_error_

This module is the error hanler for the dynamo compiler. It issues a message to the user's console indicating the error number, a severity code, a brief description of the error (if the brief option is not specified) and an indication of the equation and line number where the error occurred.

### Usage

```
declare dyn_c_error_ entry (fixed bin,fixed bin,fixed bin,fixed bin,fixed bin);

call dyn_c_error_ (severity,stmt_no,eqn_no,err_no,symbol);
```

| | |
|---|---|
| severity | The severity code of the error (1,2,3,4); |
| stmt_no | The line number in which the error occurred. If 0, no line number is mentioned. |
| eqn_no | The equation number inwhich the error occurred. If 0, no euation number is mentioned. |
| err_no | The error number. |
| symbol | If the message contains the '$' character, it will be replaced by the name of anidentifier whose index in the symbol table is 'symbol'~~xxxf~~. This argument is normally specified as 0 if no '$' appears in the message. |