Use of MetaTags

There is a compiler directive METATAGS that registers tags against a profile. The mechanism is an aid to searching for profiles that have similar attributes. The metatags can also be used to aid program management and register development progress.

Usage:

```
metatags "Uses staging; Reads psftp; Makes MDX Query";
metatags "Drops DWH Table";
```

Formally:

```
METATAG_DIRECTIVE ::= METATAGS <delimited_string>;
delimited_string ::= "<tag_list>";
tag_list ::= <tag>[;<tag_list]
```

The code can contain as many metatags directives as you like, wherever you like. The tags are registered during compile time, whether or not the code successfully runs.

The tags are registered with the profile id. You can also add/edit/remove tags in the Profile tab, look for the tag icon.
You can search for profiles that have certain tags, in the Profile tab=>List of profiles, look for the tag icon. You can list tags against profiles in the Monitor tab, look for the tag icon.

The common functions like load_from_csv automatically register their tag, so you don't have to tag a profile to say "Loads from CSV" (say).

Remember this is a directive (like COMMON and HDOCS) and not a statement. So this does not do what you might think:

```
$a = true;
if $a is true begin
   metatags "I am a true case";
   end
else begin
   metatags "I am a false one";
   end
```

Both tags "I am a true case" and "I am a false one" will be registered for the profile. They are handled at compile time before executing the code image.

Similarly:

```
$hi = "foobar";
metatags $hi;
```

Will fail to compile, and:

```
metatags "{$hi}";
```

Will register the tag {$hi} not foobar.