GNU Enterprise - Supply Chain Package Proposal

Neil Tiffin

GNU Enterprise - Supply Chain Package Proposal

by Neil Tiffin

Published $Id: SupplyChain.sgml,v 1.8 2001/12/09 14:08:46 ntiffin Exp $

Table of Contents

Chapter 1. Introduction

1.1. Description

This document describes a proposed architecture for the GNUe Supply Chain Package.

1.1.1. Source of Information

This document utilizes information from doc.txt originally written by Jorge-León (I think). It also incorporates information from GNU Enterprise Inventory Application Specs Edition 0.0.1 written by Alejandro Imass. This version was edited and expanded by Neil Tiffin (neilt@gnue.org).

1.1.2. Version

This introduction is $Revision: 1.3 $ $Date: 2003/01/05 00:00:25 $.

It is a preliminary draft proposal for discussion purposes only. Most of the business requirements are complete. That is they are listed in the particular module where they belong. They may not not completely defined yet. The forms and reports are just started and are not complete. Most of the business objects are named.

1.1.3. Purpose of Document

The purpose of this document is to define the scope of the modules in the Supply Chain Package. If any terms in this document are unclear please let me know. We are working on a glossary and I will gladly add them.

1.1.4. Definition of GNUe Supply Chain Package

The Supply Chain Package encompasses all of the transactions required to acquire product and deliver it to the customer. This includes the functions of requisitioning, purchasing, inventory management, simple inventory planning, distribution, sales transaction (order entry only), and shipping. (NOTE: Sales functionality will eventually moved to E-Commerce Package)

The Supply Chain Package does NOT include accounting (invoicing, general ledger, credit approval, or payables), MRP, routings, bills of material, manufacturing, sales prospecting, sales representatives, marketing, capital equipment, maintenance, customer care, customer support, management reporting, data warehousing and etc.

1.1.5. Scope of initial GNUe Supply Chain Package Functionality

The first version of the Supply Chain Package will be a proof of concept package to stress and test the GNUe tools architecture. As such it will have limited functionality. However, with that said the modules and objects developed for the initial package should be re-usable to address larger or industry specific implementations.

It is important that the first package have a solid design. To distribute the work and assure a sound design we need to decide what functionality will go into the initial scope and what will go into future scope.

1.1.6. Scope of Future GNUe Supply Chain Package Functionality

Future functionality describes requirements slated for a future version of GNUe. These requirements may appear in either standard GNUe functionality or in industry packages.

1.1.7. Initial modules for the Supply Chain Package

This document will lay the general groundwork for modules in the supply chain.

Each Module definition below contains a list of Business Requirements, Business Objects, and External Objects. The Business Requirements are the general business functionality that the module will support. The Business Objects are the objects whos definition is managed and controlled by the respective module. External Objects are interfaces to objects managed and controlled by other Module definitions. No attempt has been made to create an object structure in this document. That is the responsibility of the module developers. For example the Warehouse business object may be related to a general Location Business Object. It is also assumed that the requirements stated here will change as the Modules are defined and or refactored.

Chapter 2. Inventory Module

2.1. Requirements

This chapter is $Revision: 1.3 $ $Date: 2003/01/05 00:00:25 $.

2.1.1. Inventory Module Business Requirements

(TODO) This section has a lot of stuff that really should be in the warehouse module.

1. Each quantity can be designated with a definable status like good stock, reserved stock, waiting for quality approval, not-usable or etc. Each quanitity also has a definable storge location.
2. Provide capability to generate pick list from open sales orders. Pick list will have item names, quantity and warehouse locations and have a place to denote which items and quantity have actually been picked from inventory.
3. Reconcile annual or ad hoc physical inventory counts (does not include cycle counting). Print count tags and reconcile inventory.
4. Maintain standard cost of inventory items.
5. Inventory transactions. Each inventory movement will create a log entry that keeps track of the source of the transaction, (i.e. vendor receipt, customer receipt, scrap, lost, inventory adjustment, etc.) date, time, login user, item code, and quantity.
6. Maintain alternate part numbers.
7. Annotate items with stock replenishment scheme as stock (default, manual reorder) or non-stock status (special order). The Demand Management Module may add additional rules to the replenishment scheme.
8. Annotate items with resale status, product-resale, internal use, or service-resale.
9. Annotate items with MSDS information. MSDS stands for Material Safety Data Sheet. The point here is not to duplicate the MSDS, but rather to reference the correct MSDS for the item. This information will be available for printing on pick lists, purchase orders etc.

2.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

1. Cycle counting. Cycle counting is a method of issues a random count list on a daily, weekly or monthly basis with the item code and location (but now quantity), having the stock room staff count the number on hand. The computer then compares the number counted with the number expected and if they agree great. If not, then the item is counted again. If they still dont agree then the quantity is adjusted if approved by a supervisor.

2. ABC analysis and reports. Divide inventory by cost and throughput such that high impact items can be focused on.
3. Inventory performance analysis.
4. Lot tracking. Track an incoming lot of material and maintain enough information to trace the lot through manufacturing, assembly and distribution.
5. Serial number tracking. Inventory management that tracks individual items based on their serial numbers.
6. Shelf-life management. This functionality will include management of items that have an expiration date and therefore need to be removed after the expiration date and also the ability to dispense the oldest items first.

2.1.3. Business Objects

The following Business Objects are defined and maintained by this module.

1. Item

2.1.4. Forms

1. Enter results from physical count.
2. Inventory Look-up. This forms will allow look-up of inventory by description, part number, type, or location. For the group of items found it will display part number, location, location status (if more than one is defined), and quantity.

2.1.5. Reports

1. Inventory report. List of inventory items sorted in various ways. Contains item number, description, inventory types, quantity on-hand, location, cost, last receipt date, and last dispersement date. Sort by part number, description, or location.

2.1.6. Business Rules

1. Standard Cost. Standard cost is set manually by someone.

2.1.7. Business Object Definition

```
# $RCSfile: SupplyChain.ps,v $: Inventory item definition.
#
# Copyright (C) 2001 Free Software Foundation, Inc.
#
```

```
# This file is part of GNU Enterprise.
#
# GNU Enterprise is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# GNU Enterprise is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Enterprise; see the file COPYING.  If not, write to the
# Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
# 02111-1307 USA.
#
# This file originally written by Neil Tiffin (neilt@gnue.org).
#
# $Revision: 1.1 $ $Date: 2001/09/14 22:20:58 $ $Author: psu_gnue $
#

module inv
{
  # -------------------------------------------------------------------------
  # type - Type of quantity
  # -------------------------------------------------------------------------
  class i_type
  {
    char name<20>;
    char descr<70>;
  };

  # -------------------------------------------------------------------------
  # stock_quantity - Quantity tracked.
  # -------------------------------------------------------------------------
  class stock_quantity
  {
    unit::quantity   quantity; # various types of quantities
    inv::i_type     *i_type;   # on-hand, scrap, on order etc.
    whse::storage   *location; # in this one location.
  };

  # -------------------------------------------------------------------------
  # stock_item -
  # -------------------------------------------------------------------------
  class stock_item : base_item::item
  {

    inv::stock_quantity   [] quantity;   # various types of quantities

    unit::quantity    weight;
    unit::quantity    volume;
```

```
    char              msds_reference<100>;  # material data safety sheet reference
    text              inventory_notes;      # inventory handling note

  };

};
```

Chapter 3. Warehouse Module

3.1. Requirements

This chapter is $Revision: 1.2 $ $Date: 2003/01/05 00:00:25 $.

3.1.1. Business Requirements

1. Manage inventory. Provide way to track where items are stored in warehouse. Move items into new local storage locations. An item may be stored in multiple locations in the warehouse. Provide capability to combine locations and split locations. All movement that leaves a physical location will go out through the shipping module and come in through the receiving module.
2. A physical location (a building, a business address etc) may have many warehouses.
3. For each item in each storage location in each warehouse keep track of actual quantity on hand.
4. Bin Types. Each bin can utilize user definable types (i.e. static bin, movable bin, cart, reels etc.)
5. The storage locations will be defined using user definable locations like floors, isles, stands, columns, bins, etc.

3.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

1. None defined yet.

3.1.3. Business Objects

- Warehouse (Store Room)
- Bin Location within warehouse (Inventory Location)
- Pick List
- Inventory Movement Log

3.1.4. Forms

1. Move all inventory from one bin location to another bin location.

3.1.5. Reports

1. TODO

3.1.6. Business Rules

1. None defined yet.

3.1.7. Business Object Definition

```
# warehouse.gcd
#
# Copyright (C) 2001 Free Software Foundation, Inc.
#
# This file is part of GNU Enterprise.
#
# GNU Enterprise is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# GNU Enterprise is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Enterprise; see the file COPYING.  If not, write to the
# Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
# 02111-1307 USA.
#
# This file originally written by Neil Tiffin (neilt@gnue.org).
#
# $Revision: 1.1 $ $Date: 2001/07/04 19:09:00 $ $Author: psu_gnue $
#

module whse
{
  class site
  {
    char         name<25>;
  };

  class stock_room
  {
    whse::site  *site;
    char         name<25>;
  };
```

```
  class storage
  {
    whse::stock_room *room;
    text              location_desc; # TODO expand this to
                                     # floor, row, column, bin
  };
};
```

Chapter 4. Purchasing Module

4.1. Requirements

This chapter is $Revision: 1.5 $ $Date: 2003/01/05 00:00:25 $.

4.1.1. Business Requirements

1. Provide capability for non-purchasing staff to manually enter requisitions to purchase both inventory and non-inventory items.
2. Capability to requisition items without part numbers.
3. Provide capability to notify requisitioner that purchase order for the requisitioned items has been processed.
4. Provide place for requisitioner to notify purchasing that substitutions are or are not allowed.
5. Requisition will have required date which specifies the date the items must be received by.
6. Automated or batch conversion of requisitions to purchase orders based on policy
7. Provide list of open requisitions by buyer responsible and allow buyer to check each requisition and convert to purchase order.
8. Send purchase orders to vendors.
9. Keep track of which purchase orders have been printed and/or sent (via fax, EDI or mail).
10. Provide capability to resend Purchase Orders.
11. Provide capability keep track of purchase order history. When sent, when resent, when modified etc.
12. Provide capability to select a group of unsent Purchase Orders and send all at one time.
13. Maintain delivery status of line items on purchase orders. If an line item is going to be delivered by multiple deliveries then each delivery should be shown on a separate line item.
14. Provide buyer prompting for almost due or late material follow-up.
15. Provide capability for purchasing staff to enter buyer information and link buyers to simple inventory type.
16. Maintain simple user defined product types. Allow purchasing staff to enter product types in a master table and to assign inventory items to product types.

4.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

1. Maintain average lead time for each vendor supplying item.

2. Maintain level of service for each vendor.
3. Approvals and workflow.
4. Request for Quotations and Quotations.

4.1.3. Business Objects

The following Business Objects are defined and maintained by this module.

1. Purchase Order or Requisition Line Item (line number, item number, vendor item number, item description, price, quantity)
2. Purchase Order or Requisition which is an aggregation of Purchase Orders or Requisition Line Items (vendor name and address, company billing name and address, company shipping name and address, purchase order number, required delivery date, fob point, shipping method.)
3. Buyer (which is related to external object Employee)

4.1.4. External Business Object References

1. Item
2. Vendor
3. Employee
4. Currency

4.1.5. Forms

1. Purchase Order Entry
2. Requisition Entry
3. Enter, Update and Delete Buyers
4. Provide easy form to update delivery status.

4.1.6. Reports

1. Material due by due date
2. Open Purchase Orders by Vendor
3. Requisitions not converted to Purchase Orders by date
4. All purchases by item selected by time frame.

4.1.7. Business Rules

1. Lead time calculation is the time from issuing of the purchase order to vendor to the time the item is received into inventory in calendar days.
2. pur::approval - if approved and not_approved are false then the document has not been reviewed, upon review either approved can be selected or not_approved, but not both.

4.1.8. Business Object Definition

```
# purchasing.gcd
#
# Copyright (C) 2001 Free Software Foundation, Inc.
#
# This file is part of GNU Enterprise.
#
# GNU Enterprise is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# GNU Enterprise is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Enterprise; see the file COPYING.  If not, write to the
# Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
# 02111-1307 USA.
#
# This file originally written by Neil Tiffin (neilt@gnue.org).
#
# $Revision: 1.1 $ $Date: 2002/02/24 17:05:36 $ $Author: psu_gnue $
#
include "../../../base/units/classes/units.gcd"
include "../../../supply-chain/vendor/classes/vendor.gcd"

module pur
{

  # ---------------------------------------------------------------------------
  # Add purchasing info to vendor object.
  # ---------------------------------------------------------------------------
# EXTEND vendor::vendor
# {
#   text    notes;
#
#  # help en notes  "These notes will show up in the purchasing form"
#  #                "with an option to print on the purchase order.";
```

```
# };

  # ---------------------------------------------------------------------------
  # Recommended Vendor
  # ---------------------------------------------------------------------------
  class use_vendor
  {
    vendor::vendor  *recommended;
    text             notes;
  };

  # ---------------------------------------------------------------------------
  # Purchasing Item
  # ---------------------------------------------------------------------------
  class item : base_item::item
  {
    char    buyer_code<8>;
    text    purch_text;
    char    purch_desc<100>;

    pur::use_vendor [] recommend_vendor;

    # help en buyer_code  "The buyer code is a reference to the buyer responsible"
    #                     "for purchasing this product.";
    # help en purch_text  "The purchasing text is text that appears on a line item"
    #                     "on the purchase order.  It would normally be used for"
    #                     "special notes.";
    # help en purch_desc  "The purchasing description replaces the normal item"
    #                     "description for purchase orders only.";
  };

  # ---------------------------------------------------------------------------
  # buyer - employees responsible for this purchase group
  # ---------------------------------------------------------------------------
  class buyer
  {
    char                buyer_group<8>;
    person::employee [] buyers;
  };

  # ---------------------------------------------------------------------------
  # this is standard text that can be referenced by id and copied to p.o.
  # --------------------------------------------------------------------------
  class text
  {
    char      id<8>;     # is unique for id+buyer
    char      buyer<8>;  # buyer code
    text      text;      # for keeping standard text that is not by item
                         # normally for various special conditions
  };

  # ---------------------------------------------------------------------------
  # approval_list - a company wide list of all chains of approval.
```

```
# ------------------------------------------------------------------------
class approval_list
{
  char                name<25>;
  person::employee [] employee;
};

# -------------------------------------------------------------------------
# approval - the status of this po/req chain of approval
# ------------------------------------------------------------------------
class approval
{
  char              name<25>;  #list name
  person::employee *employee;
  boolean           approved     = false;
  boolean           not_approved = false;
  # pur::header      *what;      # what was approved (redundant)
};

# -------------------------------------------------------------------------
# status - overall status of this document
#   req, approved req, po, approved po, sent po, closed po
#   not approved req, not approved po.
# ------------------------------------------------------------------------
class status
{
  char name<25>;
};

# -------------------------------------------------------------------------
#
# -------------------------------------------------------------------------
class detail
{
  # pur::header     *header;          # redundant

  int              line_number;      # the line on the po
  char             item_number<25>;  # the item ordered
  pur::item       *item;             # optional
  char             purch_desc<100>;
  text             purch_text;       # for free form notes by item.
  currency::money  price;
  unit::quantity   quantity;
  date             required_date;
};

# -------------------------------------------------------------------------
# header - for requisitions or purchase orders
# -------------------------------------------------------------------------
class header
{
  char    id<10>;
```

```
    vendor::vendor *vendor;

    char    payment_terms<25>;  # TODO should be pointer
    char    delivery_terms<25>; # TODO should be pointer

    date    date_placed;
    date    date_completed;

    pur::approval [] approval;
    pur::status     *status;

    text    po_notes;        # free form notes for the total purchase order

    pur::detail [] detail;

    # help en po_notes  "These notes will show up in the purchasing form"
    #                   "with an option to print on the purchase order.";
  };
};
```

Chapter 5. Shipping Module

5.1. Requirements

This chapter is $Revision: 1.4 $ $Date: 2001/07/04 23:28:45 $.

5.1.1. Business Requirements

1. Generate generic shipping transactions for invoicing (not the invoice itself)
2. Process shipment of items to vendors (returns).
3. Process shipment of items to customers
4. Should handle shipping in Lean Manufacturing (Kanban) environment.

5.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

1. Packaging support, including box labeling, detailed list and box numbering.
2. Prepare shipping documents (bill of lading etc.)

5.1.3. Business Objects

The following Business Objects are defined and maintained by this module.

1. Shipment
2. Shipping Transaction for accounting

5.1.4. Forms

1. Ship something to customer. Be able to select sales order.
2. Ship something to supplier. Be able to select purchase order.

5.1.5. Reports

1. Daily, weekly, monthly, quarterly and yearly shipments.
2. Items to be shipped today.
3. Returns, by supplier, date range and currency value.

5.1.6. Business Rules

1. If enough items are in stock or otherwise available to ship complete order then a shipping order is generated.

5.1.7. Business Object Definition

```
# shipping.gcd
#
# Copyright (C) 2001 Free Software Foundation, Inc.
#
# This file is part of GNU Enterprise.
#
# GNU Enterprise is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# GNU Enterprise is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Enterprise; see the file COPYING.  If not, write to the
# Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
# 02111-1307 USA.
#
# This file originally written by Neil Tiffin (neilt@gnue.org).
#
# $Id: shipping.gcd,v 1.4 2001/07/05 18:38:30 ntiffin Exp $
#

include "../../../base/currency/classes/currency.gcd"
include "../../../base/units/classes/units.gcd"

module shipping
{
  class detail
  {
    unit::quantity   quantity;
    currency::money  unit_price;
  };

  class shipment
  {
    date                date;
    char                carrier<25>;         # carrier name
    char                delivery_terms<25>;  # overnight
    pur::header        *po_header;
```

```
    so::header         *so_header;
    pur::detail      [] po_lines;
    so::detail       [] so_lines;
    shipping::detail [] detail;
  };

  class journal
  {
    date                 date;
    shipping::shipment *shipment;
    currency::money     total;
  };
};
```

Chapter 6. Receiving Module

6.1. Requirements

This chapter is $Revision: 1.4 $ $Date: 2001/03/19 21:11:10 $.

6.1.1. Business Requirements

1. Process receipt of items from vendors.
2. If an item is delivered and the shipping documents do not specify a purchase order then proide a method to lookup the purchase order number by various means (at least vendor name and part number).
3. Entering a purchase order number causes the system to display a list of all open items on the purchase order.
4. By entering the quanitity or selecting 'all' the the system will generate a receipt transaction (for accounting) and generate a reciept with proposed storage location.
5. The receiving staff may override the storage location manually or define one if it does not already exist.
6. Approving the proposed storage location will automatically update the quantity on-hand.
7. Process receipt of items from customers (returns).
8. All items received against purchase orders will have the last actual cost and average cost updated.
9. Provisions for establishing policy for receiving less or more than ordered and maintaining policy.

6.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

1. none

6.1.3. Business Objects

The following Business Objects are defined and maintained by this module.

1. Receipt Transaction for accounting.

6.1.4. External Business Object Referencs

- Item
- Customer
- Vendor
- Purchase Order
- Purchase Order Line Item
- Warehouse
- Location

6.1.5. Forms

- Receive Item
- Update Cost

6.1.6. Reports

- none

6.1.7. Business Rules

1. Last Cost. Last cost is the line item purchase order cost for the item received.
2. Average Cost. Average on-hand cost is calculated as the ((average on-hand cost times quantity on-hand)+(purchase order line item cost * quanitity received)) / (total on-hand + total received).

6.1.8. Business Object Definition

```
# receiving.gcd
#
# Copyright (C) 2001 Free Software Foundation, Inc.
#
# This file is part of GNU Enterprise.
#
# GNU Enterprise is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# GNU Enterprise is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
```

```
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Enterprise; see the file COPYING.  If not, write to the
# Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
# 02111-1307 USA.
#
# This file originally written by Neil Tiffin (neilt@gnue.org).
#
# $Revision: 1.1 $ $Date: 2001/05/05 00:35:09 $ $Author: psu_gnue $
#

module receiving
{
  class receiving
  {
    char  dummy<8>;
  };
};
```

Chapter 7. Vendor Module

7.1. Requirements

This chapter is $Revision: 1.2 $ $Date: 2001/03/06 16:12:52 $.

7.1.1. Business Requirements

1. Maintain Vendor Information (name address, status etc).
2. Vendor status information maintained with separate form for control and approval purposes.

7.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

- none

7.1.3. Business Objects

The following Business Objects are defined and maintained by this module.

- Vendor

7.1.4. External Business Object References

- none

7.1.5. Forms

- Vendor Maintenance (Add, remove, update vendor information except status)
- Vendor Status Maintenance (Set vendor status active or don't use)

7.1.6. Reports

- Vendor Master List sorted by purchasing volume year to date

7.1.7. Business Rules

- none

7.1.8. Business Object Definition

```
# vendor.gcd
#
# Copyright (C) 2001 Free Software Foundation, Inc.
#
# This file is part of GNU Enterprise.
#
# GNU Enterprise is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# GNU Enterprise is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Enterprise; see the file COPYING.  If not, write to the
# Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
# 02111-1307 USA.
#
# This file originally written by Neil Tiffin (neilt@gnue.org).
#
# $Revision: 1.1 $ $Date: 2001/12/02 10:56:16 $ $Author: psu_gnue $
#

include "../../../base/org/classes/org.gcd"
include "../../../base/currency/classes/currency.gcd"

module vendor
{
  # ------------------------------------------------------------------------
  # vendor status i.e. approved, dont use, special purchase only
  # ------------------------------------------------------------------------
  class status
  {
    char         code<8>;
    char         description<25>;

    INDEX (code);
  };


  # ------------------------------------------------------------------------
```

```
  # type of vendor.  service, raw material, labor.
  # -------------------------------------------------------------------------
  class v_type
  {
    char       code<8>;            # vendor type
    char       description<25>;

    INDEX (code);
  };


  # -------------------------------------------------------------------------
  # vendor specific information
  # -------------------------------------------------------------------------
  class vendor : org::organization
  {
    text            ven_notes;    # notes specific to vendors

    vendor::status *status;   # Approved, dont use etc.
    vendor::v_type   *v_type;     # Service, manufacturer, office supplies etc.

    char            std_terms; # TODO should be pointer

    currency::money year_0;  # this year to date
    currency::money year_1;  # last year
    currency::money year_2;  # year before

    date            last_receipt;
  };

};
```

Chapter 8. Capital Equipment Module

8.1. Requirements

$Revision: 1.1 $ $Date: 2001/03/05 01:16:33 $

8.1.1. Business Requirements

Maintain a list of capital equipment used in the business, when it was acquired, initial cost, capital equipment number, and where it is located. This module will not appear in the initial GNUe effort. It is here merely to have a place to put requirements as they come up.

8.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

- none

8.1.3. Business Objects

- Equipment

8.1.4. External Business Object References

- none

8.1.5. Forms

- none

8.1.6. Reports

- none

8.1.7. Business Rules

- none

Chapter 9. Independent Demand Planning Module

9.1. Requirements

This chapter is $Revision: 1.4 $ $Date: 2001/03/06 14:55:44 $.

9.1.1. Business Requirements

1. Provide on-command and periodic scan of inventory items creating purchase proposals (requisitions) based on inventory reorder policy. The reorder policy is defined for each inventory item or each inventory type.
2. Optionally maintain item's reorder level and reorder quantity depending on reorder policy.
3. Optionally maintain item's reorder minimum quantity and maximum quantity depending on reorder policy.

9.1.2. Future Requirements

The following items will be implemented in future version of this module. They may be implemented in the base module or be added via an industry specific add on.

1. Implement Economic Order Quantity Calculation in scanner.
2. Maintenance of Economic Order Quantity in item files.

9.1.3. Business Objects

- Usage History - needs definition.

9.1.4. External Business Object References

- Item

9.1.5. Forms

- Add, update, and delete reorder policy and values.

9.1.6. Reports

- none

9.1.7. Business Rules

1. Order Point Rule. The system maintains a minimum quantity and reorder quantity. Anytime the on-hand quantity is at or below the minimum quantity the reorder quantity is proposed as a requisition. Note that the reorder quantity is a fixed amount regardless of the quantity on-hand.
2. Min-Max Rule. The system maintains a minimum quantity and a maximum quantity. When scanned any on-hand quantities that are below the minimum quantity will have purchase proposals generated. The proposed quantity will be the amount required to bring the on-hand quantity up to the maximum level.

I. Appendixes

Appendix A. GNU General Public License

```
          GNU GENERAL PUBLIC LICENSE
        Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

    Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
```

patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

```
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.  SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

     END OF TERMS AND CONDITIONS

    How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    -one line to give the program's name and a brief idea of what it does.-
    Copyright (C) 19yy  -name of author-

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA


Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:
```

```
    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  -signature of Ty Coon-, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix B. GNU Free Documentation License

GNU Free Documentation License
Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE