



HOW TO
Plan and Create a
Miva Merchant Module

Revision 1.0



Miva Corporation

2629 Ariane Drive
San Diego, CA 92117
Telephone: 858-490-2570
Telefax: 858-490-0548

<http://www.miva.com>

info@miva.com

This document and the software described by this document are copyright 2000 by Miva Corporation. All rights reserved. Use of the software described herein may only be done in accordance with the License Agreement provided with the software. This document may not be reproduced in full or partial form except for the purpose of using the software described herein in accordance with the License Agreement provided with the software. Information in this document is subject to change without notice. Companies, names and data used in the examples herein are fictitious unless otherwise noted.

Miva is a registered trademark of Miva Corporation. Miva Order, Miva Merchant, Miva Mia, Miva Empresa, the Miva "blades" logo, and the Miva Engine are trademarks of Miva Corporation. Windows is the registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. This document was developed and produced in San Diego, CA, USA.

MIVA CORPORATION WILL NOT BE LIABLE FOR (A) ANY BUG, ERROR, OMISSION, DEFECT, DEFICIENCY, OR NONCONFORMITY IN MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE; (B) IMPLIED MERCHANTABILITY OF FITNESS FOR A PARTICULAR PURPOSE; (C) IMPLIED WARRANTY RELATING TO COURSE OF DEALING, OR USAGE OF TRADE OR ANY OTHER IMPLIED WARRANTY WHATSOEVER; (D) CLAIM OF INFRINGEMENT; (E) CLAIM IN TORT, WHETHER OR NOT ARISING IN WHOLE OR PART FROM MIVA CORPORATION'S FAULT, NEGLIGENCE, STRICT LIABILITY, OR PRODUCT LIABILITY, OR (F) CLAIM FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES, OR LOSS OF DATA, REVENUE, LICENSEES GOODWILL, OR USE. IN NO CASE SHALL MIVA CORPORATION'S LIABILITY EXCEED THE PRICE THAT LICENSEE PAID FOR MERCHANTABILITY.

MM1031-01, Rev. 1.0

HOW TO

Plan and Create a Miva Merchant Module

Introduction

A module is a file that contains Miva Script functions that are called by the core Miva Merchant program to accomplish a task. Every Miva Merchant module requires a basic structure, using a set of required API functions. Miva Merchant defines several module types, and each of these types additionally require their set of API functions. Refer to the *Miva Merchant Module Development Reference Manual* for a list and description of all these functions.

The Miva Merchant Administrative Interface, shopping interface, setup and upgrade all call certain functions within modules. The point at which a function in a module will be called is predetermined, based on the type of module and what actions are being taken in the program calling it. The structure, requirements, and restrictions of modules are a set of rules, called the Miva Merchant API (Application Programming Interface) which determines how various programming elements work together.

This document contains the basic plan and design steps for creating any module.

Planning Phase

Your module creation can be easier, quicker and of higher quality with good planning. In the planning phase, you want to:

1. Define your basic goal
2. Write the functional specification
3. Determine the module type

Define your Basic Goal

Identify what you want the module to accomplish. Determine if this task is already being partly or fully accomplished. You may only need to access some information that is already within Miva Merchant to achieve your task.

You may also find that part or all of your task can be accomplished by entering code (HTML, Java Script, etc.) into headers, footers, or other text boxes of Miva Merchant.

Write the Functional Specification

The functional specification may be brief, and it should include the following items:

- **The purpose:** what the store administrator/manager or the shopper will be able to do as a result of your module being installed.
- **Requirements:** identify those events that must occur in order for your module to perform its purpose. Include some specifics, for example:
 - writing a file to a specific location
 - copying a file, and adding specific data to the new file
 - importing or exporting specific data

Determine the Module Type

You should refer to the *Miva Merchant Module Development Reference Manual*, which lists the functions for each module type, to help you determine which module most fits your goal, as stated in your Functional Specification.

- What areas of Miva Merchant does your goal act on or relate to?
If, for example, it relates to both shopping and administration, you can omit from your consideration the modules that run only in the administration interface, such as batch, export, import, and wizards.
- At what point does the event(s) take place that you want to accomplish?
 - For example, identify if the shopper will be viewing a product, logging in, adding item(s) to the basket, or completing the order. This will help you narrow your selection for the type of module to use.
- Does there seem to be a fit with more than one module?
 - Examine the functions for each Module type to see if all the functions you need are in one and not another.
 - See if creating one of the modules is a much longer or more involved task than the other.

Design Phase

In the planning phase, you want to:

1. Write the technical specification
2. Build your module
3. Test your module

You define the details in your specification before you actually build the module. You may want to build an optional function(s) or routine(s) as you write your specification to help you determine the approach.

Write Your Technical Specification

Include the purpose of the module, basic information, specific requirements, such as, structure of new databases, the content of a new administrative interface screen(s), and other details relevant to your specific module.

Purpose

Identify specifically what the store manager or shopper will be able to do.

Basic Information

Include the following items:

- Module filename
- Module type
- Module code
- Module name
- Module provider
- Module API version
- Module version
- Module upgrade

Note: Your module is given a unique identifier which is stored in the StoreModules database, storemod.dbf, along with the module type and number of stores using the module.

Module Requirements

Expand the requirements you identified in the functional specification, include the following types of information, for example:

- List the specific module functions that you will use to accomplish each of the requirements, and at what point each must be used.
- Identify the Miva Script tags and at what point each will be used.
- Identify HTML, Java Script, or other code, and at what point it will be used.
- If your module provides a new screen, what tags, switches are used, and what the administrator can override.
- If creating a new database, identify:
 - Database name(s), when it's created - within what function.
 - Index name(s), expression(s), and flag(s), if any.
 - Field names, type, description, and any related database and field.
- Identify what existing Miva Merchant database file(s), index(es) you need to read, and at what point. Refer to the *Miva Merchant Database Reference Manual*.
- Determine which database functions you need. Refer to the *Miva Merchant Database & Utility Functions Reference Manual*.
- When upgrading your module and adding new fields, copy existing databases to temporary ones, add new fields to existing databases, then recopy data from the temporary databases to the existing ones.

Build your module

Build a skeletal

A skeletal module includes all the required functions, including the basic ones that all modules have, and the specific functions for the module type you are creating.

Include the following, as required by each of the functions:

- function name
- standard output level
- parameters
- return value

Complete module coding

Refer to the *Miva Script Reference Manual*, along with the other manuals mentioned above, to complete coding for your module.

Test Your Module

Perform testing to ensure your module functions as you identified in the Functional and Technical Specifications. You may also want to perform other functions within Miva Merchant, then return to your module to verify integration functionality.

Rewrite your code, as necessary to correct problems or unanticipated occurrences, then retest your module.

The actions you should perform to test your module include the items listed below. For specific module type test, refer to the individual “How To” for the module type.

- Install module using add modules
- Check hyperlinks
- Make changes, and verify updates
- If new tabs are used, make sure changes are retained moving from one tab to another
- Uninstall your module, verifying it removes all files, directories, databases, indexes (mvx), and memo fields (dbt).

Implementation Phase

Beta Test

When you have completed your in-house testing, release the module to your selected customers for a beta test.

Based on the results from your beta test, make changes in your module, and retest your module for certification and release.