

**Forest Service Handbook
National Headquarters - Washington Office
Washington, DC**

**Forest Service Handbook 6609.13 – Application Developer's Handbook
Chapter 10 – Developer's Guidance**

Amendment: 6609.13-2008-1

Effective date: September 15, 2008

Duration: This amendment is effective until superseded or removed.

Approved by: Hank Kashdan, Deputy Chief

Date approved: September 12, 2008

Responsible Staff:

Last Change: 6609.13-91-1 to FSH 6609.13

Superseded Document(s): 6609.13, Contents, Amendment 6609.13-91-1, September 3, 1991; 6609.13,0 Code Contents, Amendment 6609.13-91-1, September 3, 1991; 6609.13,0 Code, Amendment 6609.13-91-1, September 3, 1991; 6609.13,10 Contents, 6609.13,10 Code, Amendment 6609.13-91-1, September 3, 1991; 6609.13,20 Contents, Amendment 6609.13-91-1, September 3, 1991; 6609.13,20, Amendment 6609.13-91-1, September 3, 1991; 6609.13, 30 Contents, Amendment 6609.13-91-1, September 3, 1991; 6609.13,30, Amendment 6609.13-91-1, September 3, 1991; 6609.13,50 Contents, Amendment 6609.13-91-1, September 3, 1991; 6609.13,60 Contents, , Amendment 6609.13-91-1, September 3, 1991; 6609.13,60, Amendment 6609.13-91-1, September 3, 1991; 6609.13,70 Contents, Amendment 6609.13-91-1, September 3, 1991; 6609.13,70, Amendment 6609.13-91-1, September 3, 1991

Digest: Following is an explanation of the changes throughout the directive by section.

05: Removes obsolete definitions and adds new definitions for Major Application and Major Information System.

10: Throughout chapter removes obsolete references to other chapters of this handbook and adds as a reference FSM 6680.

Adds reference to the System Development Life Cycle (SDLC) Roadmap.

20: Removes entire chapter. Development procedures are defined by the System Development Life Cycle (SDLC) Roadmap, available online from the FS BAO web page, at <http://fsweb.wo.fs.fed.us/irm/bao/>.

30: Throughout chapter removes obsolete references and direction and adds reference to FSM 6680 for direction pertaining to security. Makes minor edits in direction for clarity.

31.3: Clarifies records retention requirements and adds reference to FSH 6209.11, Records Management Handbook.

31.32: Removes code, caption, and associated direction for “The Application Security Certification Statement Document” and rennumbers the two sections following.

50: Removes entire obsolete chapter.

60: Removes entire chapter direction related to application security certification is now provided in FSM 6680.

70: Removes entire obsolete chapter. Direction related to application security certification is now provided in FSM 6680.

Table of Contents

11 - Importance of Developing Quality Applications.....	4
12 - Characteristics of Quality Software	4
13 - Principles to Follow to Produce Quality Applications	5

11 - Importance of Developing Quality Applications

In making application design decisions, do not try to save development time at the expense of the time to distribute, install, and assimilate the application into daily business practices. Avoid tradeoffs which reduce development time or cost at the expense of computer system security, information integrity, and/or non-redundant information storage since these expenses will be incurred on hundreds of sites. Do not release an application before quality control testing has been completed. Frequent revisions, requiring repeated installations, upgrades, or even de-installation at numerous sites are costly and should be avoided.

12 - Characteristics of Quality Software

Developers should check their designs and prototypes against the following to ensure their final products exhibit these characteristics. Quality Forest Service software is:

1. Secure. Secure applications provide management controls for every identifiable threat or vulnerability they introduce into the corporate information environment.
2. Useful. The application fills an organizational need as determined by the appropriate level of management. Design, development, implementation, and use are directed by management to accomplish Forest Service goals. Feedback from intended users is reflected in the application.
3. Efficient. Efficient applications use the minimum amount of human and computer resources required to achieve the objectives of the application.
4. Integrated. Integrated applications can interoperate and operate within the national and local components of the Forest Service Information Management Strategy, collectively referred to as the Forest Service Business Solution Architecture.
5. Correct. Applications comply with applicable laws and regulations and with management, financial, accounting, budget, and/or other appropriate standards. They process only valid data, that is, only data which represent legitimate events. The data they produce reflects the correct cycle, version, or period for the processing being performed.
6. Auditable. Applications generate an audit trail which establishes individual accountability for transactions and permits an analysis of breakdowns in the system and other anomalies. Further, the business rules and business algorithms used by applications are easy to discover and document when necessary.
7. Familiar. Applications are familiar when they exhibit the functionality and characteristics of other applications used daily by a majority of their intended users.
8. Clear. The functions and operation of applications are clearly understood from the user interface, help screens, and documentation.

9. Reliable. Applications accurately perform all functions, as described in the specifications and documentation, exactly and correctly. Their documentation accurately describes how they are used to accomplish their functions. Where practicable they provide for alternative processing in the event of prolonged hardware or software system failures. The data they produce is promptly available.
10. Resilient. Programs include routine error checks and recovery options for any possible user or data errors. Error messages guide the user to make appropriate choices to successfully operate the application. Inconsistent commands or input of unacceptable data never leaves data in an inconsistent state, causes abnormal termination, or otherwise places users in situations from which they cannot easily recover.
11. Testable. Applications, their documentation, and their associated installation utilities can quickly and economically be fully validated or re-validated on a repeatable basis.
12. Evolutionary. Applications are evolutionary if new capabilities can be easily built on an existing functional foundation.
13. Portable. Applications are portable if they can be easily installed and operated on numerous sites, having diverse operating systems and hardware configurations.
14. Maintainable. Application code is maintainable if other programmers can easily diagnose and correct problems and errors.
15. Reuse Oriented. Applications are reuse oriented if they create products (information as well as code) which can be easily used by other applications or themselves use products created by other applications.
16. Supportable. Applications can be supported by a user Support team.
17. Measurable. Applications provide the information needed to measure the business-level performance and success of the business function that the application supports.

13 - Principles to Follow to Produce Quality Applications

1. Give overriding priority to information security, sensitivity and integrity considerations. See section 05 for the definition of a sensitive application and its description of sensitive data. Take the following actions to ensure the security and integrity of Forest Service information:
 - a. Determine, as early as possible in the development process, whether an application will be accessing or generating sensitive information. For such applications, identify, throughout the entire development cycle, threats to the security and integrity of the information. When discovered, controls to counter the threats should be designed and implemented.

- b. For any application which has been identified as dealing with sensitive information initiate an "in-house certification procedure." Procedures for security certification are defined in FSM 6680.
2. Make efficient use of people as well as machines. Save human resources by assisting the user in every detail, even if it means that the application will need more disk storage, use more system resources, and require greater effort by the programmer. Accomplish this by using such mechanisms as familiar and powerful user interfaces, well-written informative error messages with graceful recovery, electronic user manuals accessible during application execution.

Strive simultaneously, to ensure that applications consume the minimum amount of system resources needed to provide quality service to the users.

3. Capitalize on the investments the Forest Service makes in information, people, and technology. Employ those tools made available as part of the Forest Service Business Architecture. Take actions such as the following:
 - a. Model application user interfaces on those interfaces used in daily Forest Service work. The objective is to capitalize on the substantial investment represented by the thousands of Forest Service employees who have become familiar and comfortable with these interfaces. See chapter 50 of this handbook for information about the recommended Forest Service formats for various types of interface screens.
 - b. Create information in a reusable electronic form. Don't create hard copy outputs as a default but rather make users explicitly choose to generate and print them.
 - c. Draw information from and put information back into corporate databases. These will be implemented as Structured Query Language, (SQL), relational data bases and will classify Forest Service information into subject matter areas, such as People, Money, Natural Resources, and Things (for example, facilities).
4. Operate within the constraints imposed by the information, people, and technology aspects of the Forest Service Information Management Strategy. Most Forest Service applications should be designed to operate on Forest Service equipment and should use, primarily, the software resources management has made available on this equipment. Do not require new data and information structures (for example, codes, files, databases) if corporate information structures exist which will suffice. Do not require extensive, application specific training of the intended users and/or of the people who are to support the application.

Be knowledgeable about which, of the many development tools on the market, the Forest Service has chosen for its use. Use these tools to the maximum extent before attempting to obtain different tools.

5. Be familiar with official publications and resources which bear on the application development activity. These include:

- a. FSM 1390 - Information Management
- b. FSH 1309.14 - Information Requirements Handbook
- c. FSH 6609.11 - System Management Handbook
- d. FSH 6609.13 - Application Developer's Handbook
- e. FSM 6620 - Computer Systems Management
- f. Forest Service System Development Life Cycle (SDLC) Roadmap, available from the CIO Business Application Office website.

6. Employ emerging software engineering concepts and technologies as they become mature and standardized. Stay abreast of advances in software engineering by, for example, reading professional journals and attending professional seminars. Employ these advances in your work in ways consistent with the agency's corporate information management strategy. Examples of new concepts which are mature enough for you to use include:

- a. The opportunity to use a rapid prototyping development approach.
- b. The opportunity to put many of the rules about information objects into the objects themselves and thus remove it from source code (for example, by using referential, key, and access constraints in SQL relational data bases).

7. Anticipate distribution and update needs. Write clear, concise installation documentation and provide fail-safe installation macros. These cost avoidance measures are especially necessary for applications which will be distributed to sites which have limited systems management resources. Avoid causing down time and wasted effort at hundreds of sites due to installation foul-ups related to your application.

If your application is to run at sites other than the development site, alert those sites in advance. For applications anticipated to have major system or personnel impacts, send out an initial alert as soon as development of the application is approved. Repeat alerts, if practical, as milestones in the development are reached. Include in the alerts information which predicts the application's implementation date and helps the sites plan for their system needs (for example, need for disk, memory, communications, database administration skills, and so forth). Send out a final alert, about 1-1/2 to 2 months before the application's release, which contains a final estimation of the application's expected impacts and information which will allow a system manager to anticipate how complicated and long the installation of the application is likely to be. In each alert "market" the application, that is, describe the benefits it will bring to its users and to the receiving site as a whole. This information is best if detailed enough to help the site's

management team prioritize the use of their system and personnel resources relative to their whole mix of applications.

Good installation procedures are discussed in chapter 20. An example of good installation documentation is given in chapter 60, exhibit 04. See FSM 6620 for policy covering distributing applications by means of the National Software Distribution Procedure.

8. Include a plan for managing all the various aspects of the application once it is released for general use. Among the various aspects which an application might have to plan for are:

- a. Backup and archiving requirements and procedures.
- b. Records management requirements including possible release of any and all information objects under the Freedom of Information Act and Privacy Act (FOIA/PA).
- c. Data administration (for example, managing standard information structures and coding conventions).
- d. Data base administration (for example, managing of user views and access privileges).
- f. Scheduling of security audits and application re-certifications.
- g. Local site version control of the application's software.

9. Build documentation for operating and maintaining the application into the application itself when practical. Apply this principle by:

- a. Automating data entry and edit rather than requiring precisely constructed files of input data. Supply application components which manage interactive sessions with users and employ such things as titled prompts, input by example, intuitively arranged sequences of menus, and on-line context-sensitive help.
- b. Giving variables used in source code and macros meaningful names. All variables are explicitly typed in a logical place and implicit typing features of languages are avoided. If possible use names employed by others for the same variable.
- c. Following structured programming practices, such as indenting successive nested loops and if-then-else constructs, using subprogram modules to break code into logical units, minimizing use of GO TO statements, and so forth.
- d. Liberal use of comment statements in source code and macros.