Mn/DOT Contract No.: 97689

Admin No.: 24155

OET No.: 2694

**IT Professional Technical Services**
**Master Contract Program**
**# 902TS**
**Statement of Work (SOW)**
**For Technology Services**
**Issued by**
**Minnesota Department of Transportation (Mn/DOT)**

**Project Title:** **P1267 Mobile Data Collection**

**Service Categories:**
- **Geographic Information Systems (GIS)**
- **Architecture Planning & Assessment – Technical**

# 1 Business Need

## 1.1 Project Overview

Mn/DOT's Office of Information & Technology Services (OI&TS) has a need to to provide business areas with the ability to rapidly develop, deploy, and maintain mobile data and applications, in addition to providing OI&TS staff the ability to deliver information to our customers through a mobile setting. This includes the ability to capture structured and unstructured field data, including non-survey mapping grade data, using mobile devices (Smartphones, PDAs, GPS Receivers, Notebooks, Laptops, Tablets, etc.).

This work will be accomplished in three (3) major steps: business analysis/requirements, conceptual design and recommendation, and implementation.

1.1.1 **Business Analysis/Requirements:** The Selected Responder will provide business analysis for the project. This includes meeting with Mn/DOT groups to gather requirements for how they envision their mobile workforce. This phase will help us understand our customer's needs and vision while building a culture at Mn/DOT that embraces mobile technologies.

1.1.2 **Conceptual Design and Recommendations:** The Selected Responder will provide a final report with recommendations on how Mn/DOT needs to equip/design for our mobile data collection needs, and establish standards for the mobile data collection process and data handling.

1.1.3 **Implementation**: After Mn/DOT review and approval of the requirements and recommendations, the Selected Responder will successfully construct and deploy mobile applications in the Mn/DOT development/test/production environment for 2 business cases identified by Mn/DOT, one for mapping data and one for standard data collection, applying all project deliverables. This will include template products Mn/DOT can extend and knowledge transfer for Mn/DOT staff.

## 1.2 History

Over the past three years there has been significant demand by business areas to have applications to collect or update data in the field. There are at least twelve separate mobile based projects that have been developed or are in the queue at Mn/DOT, with several more being discussed. Those that have been implemented used different approaches to mobile data collection of field data. Providing a common framework and associated specifications, tools and processes, etc., will significantly increase data integrity and provide new efficiencies for meeting field data collection needs.

### 1.3 Business Objectives

Obtain an understanding of and document clearly defined business needs, processes, and functional requirements related to mobile field data collection, including positional accuracy standards for Mn/DOT assets (Mapping Grade Standards). Create a Mobile Framework and associated processes including data models for spatial and non-spatial data (including mapping grade data), standard software platforms, standard hardware specifications, mobile application templates for each environment, a common bi-directional interface for data, common methods/processes and test cases, and guidance and support mechanisms that will allow business areas to rapidly setup and deploy applications and services to an internal mobile workforce and external customers, reducing the time required to modify or extend a database to better meet changing business needs. This framework and associated processes must:

- Be designed to optimize the equipment in the mobile environment (i.e. designed to maximize battery life of mobile devices, provide fast connectivity, tell geolocation, connect to a specific port or com, is camera accessible, allows easy, fast update of all devices with security patches, software updates, etc.).
- Improve data integrity by providing for the collection of data using common data structures, design, and built in quality control where possible (i.e. drop-down lists/ pick lists).
- Include automated processes to the extent possible.
- Be easily adaptable to future technology changes.
- Ensure all mobile applications are products under a single support model that is sustainable.

### 1.4 Location of Services

The Selected Responder will be required to work on-site part-time as follows:

- **Requirements Interviews:** On-site
- **Requirements Development/Documentation:** Off-site
- **Design:** Off-site (with on-site work as needed by the Selected Vendor Designers or Mn/DOT staff providing information/assistance for the design work)
- **Implementation:** On-site.
  *The Selected Responder must complete any required implementations at Mn/DOT's CO network operations center (NOC). The Selected Responder must ensure the system is accessible from offices within Mn/DOT through its local area network (LAN) and wide area network (WAN).*

All on-site work will be performed at the Mn/DOT Central Office located at 395 John Ireland Blvd., St Paul, MN 55155.

## 2 Product Requirements

The Selected Responder must deliver project products that meet all of the following requirements, with Mn/DOT having final approval of requirements satisfaction.

### 2.1 Requirements

2.1.1 Facilitate sessions to interview business and technical stakeholders department-wide, as identified by Mn/DOT. Identify and document business and technical needs (functional and non-functional requirements) for mobile data collection and use, including needs for field data collection at different mapping grade levels per asset category. This will include stakeholders from 8 Districts, approximately 12 Offices, and 2 to 3 other State agencies. It is anticipated it will take approximately 8 stakeholder meetings.

2.1.2 Document all requirements with applicable modeling or documentation techniques (i.e use cases, supplementary requirements, business rules, data models, etc.) using Mn/DOT templates.

2.1.3 Requirements must be managed according to section 3.3.

## 2.2 Conceptual Design and Recommendation

Develop a proposed system design and recommendation for a mobile data framework including the following:

2.2.1 Architecture

Design a common mobile data collection framework (including mapping grade data) and associated specifications and processes to rapidly develop, deploy and maintain new mobile applications and data according to section 3.4 and including:

- Data models for spatial and non-spatial data (including a mobile database and published data made available to mobile interfaces).
- Standard software platforms
- Standard specifications for mobile data collection devices (hardware)
- Re-usable template applications for the applications in section 2.3.
- Mechanisms for off-line work and data synchronization when connected;
- Common business workflows/methods/processes and test cases to rapidly develop, deploy and maintain new mobile applications and data.
- User best practices for mobile data collection and associated tools, workflows, methods, processes, application workflow, etc.
- Departmental policies and guidelines for mobile data collection hardware, software, and data to ensure data collection is consistent with established architecture and standards;
- Guidance and support mechanisms.

2.2.2 Data

- Design a flexible data structure that is easily selectable, extensible, and deployable to mobile applications, and build the database schemas and domains. Develop data transfer specifications for communication protocol (API) between the mobile field data collection system and source business systems. Include protocol and processes.
- Develop specifications for business systems to input (checkout/checkin) its data structure to a mobile environment.
- Develop a specification for how we publish our data to be consumable by a mobile device.

2.2.3 Develop a recommendation of standard Software Platforms

Define product specifications and identify software products meeting the needs of stakeholders for spatial and non spatial mobile data collection for all environments.

2.2.4 Develop standard specifications for mobile data collection devices (hardware)

- Develop standard specifications for mobile data collection devices to meet business area needs and spatial data accuracy requirements for mobile collection including map data.
- For non-survey mapping grade data, identify equipment capable of providing the accuracy needs and recommendations for standard equipment bundles for field data collection. This includes SBAS (Satellite Based Augmented Signal Correction (WAAS, EGNOS)), VRS from Mn/DOT's own CORS system, Beacon, Post Processed, or combination of the above. This will also include supplemental equipment such as range finders and cameras.

2.2.5 Workflows/Methods/Processes

- Develop common business workflows/methods/processes to rapidly develop and deploy new field applications and to add new field assets to the database.

- Develop remote data collection workflows, guidelines, and standards.

2.2.6 User Best Practices
Develop user best practices for mobile data collection and associated tools, workflows, methods, processes, application workflow, etc.

2.2.7 Departmental Standards and Guidelines
Develop enterprise standards and guidelines for mobile data collection hardware, software, and data to ensure data collection is consistent with established architecture and standards.

2.2.8 Guidance and Support Mechanisms
Develop the guidance and support mechanisms required to implement and maintain the system.

2.2.9 Knowledge Transfer
Develop complete documentation of all parts of the mobile framework and a knowledge transfer program that will provide Mn/DOT staff with the skills and knowledge to successfully implement all project deliverables.

2.3 **Implementation**

2.3.1 Successfully apply all deliverables for 2 business cases, one for mapping data and one for standard data collection, identified by Mn/DOT and following development processes outlined in section 3.5.

2.3.2 Develop a prototype of all parts of the mobile framework, present to Mn/DOT, and incorporate Mn/DOT modifications.

2.3.3 Complete the development based on Mn/DOT feedback on the prototype. This includes developing re-usable template(s) for the applications.

2.3.4 Develop and conduct testing and quality assurance according to sections 3.6 and 3.13

2.3.5 Deploy the mobile application and associated process project deliverables in the Mn/DOT development, test, then production environment in a state that allows immediate use, including any template application(s), tools, methods/process documentation, guidance and support mechanisms and documentation, etc., following the project approach for deployment in Section 3.8.

2.4 **Security Audit**

2.4.1 The solution must pass a security audit based on the OWASP Top Ten Security Vulnerabilities (http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project).

2.4.2 Mn/DOT uses a variety of mobile devices, a security protocol for on-device encryption for is required.

2.5 **Knowledge Transfer**

Conduct knowledge transfer at a Mn/DOT facility in the Twin Cities Metropolitan area covering implementation, update, and maintenance of all deliverables (Train the Trainer exercise to key experts at Mn/DOT).

2.6 **Warranty**

2.6.1 The Selected Responder must provide a one year warranty against defects within the scope of this project, beginning immediately following successful implementation in the Mn/DOT environment.

2.6.2 Any of the Selected Responder's in-scope defects found during the warranty period must be troubleshot and fixed free of charge by the Selected Responder. This warranty includes any 3rd party modules used by the Selected Responder.

### 2.7 Technical Standards

2.7.1 The project must comply with the State of Minnesota and Mn/DOT standards in Appendix A.

2.7.2 All products of this contract must be accessible to people with disabilities. Therefore, all reports and deliverables provided in electronic media, including web-based intranet and internet format shall conform to the relevant accessibility standards references. At a minimum, these provisions include:

- 1194.22 Web-based Intranet and Internet Information and Applications
- 1194.31 Functional Performance Criteria
- 1194.41 Information, Documentation and Support

## 3 Project Requirements

### 3.1 Project Management

3.1.1 The Selected Responder must provide a written, detailed project plan that identifies project resources and a detailed project schedule (in Microsoft Project). The plan should follow an incremental, iterative development methodology. The plan should include a detailed work breakdown structure for each phase and iteration.

3.1.2 The Selected Responder must complete all deliverables on time, within budget, and of the expected level of quality.

3.1.3 Mn/DOT will review and approve all deliverables. Mn/DOT must approve a deliverable before accepting and paying for the deliverable.

3.1.4 The Selected Responder must provide timely and informative status information, including, but not limited to: (also see the Reporting Requirements section)

- Written, monthly high-level status reports, in a format acceptable to the Mn/DOT Project Manager and the Mn/DOT Office of Information & Technology Services, delivered to the Mn/DOT Project Manager and the Mn/DOT Office of Information & Technology Services.
- Weekly status reports, in a format acceptable to the Mn/DOT Project Manager and the Mn/DOT Office of Information & Technology Services, delivered to the Mn/DOT Project Manager and the Mn/DOT Office of Information & Technology Services.
- Periodic status review meetings
- If needed, daily stand-ups or scrums or web/teleconferences

3.1.5 The Selected Responder must follow all applicable project management standards (see the Project Standards section).

### 3.2 Project Approach

3.2.1 The project must use an iterative methodology.

3.2.2 The project methodology must be as agile as possible given resource, contracting/ outsourcing, and other constraints.

3.2.3 The project must be broken into several iterations each lasting 2 to 8 weeks.

3.2.4 Each iteration must deliver an increment of the system that is deployable (as applicable), usable and testable.

3.2.5 Acceptance criteria for the project as a whole and for each iteration, must be developed with the full participation of the users. Such criteria must be established at the start of the project

and each iteration and formally assessed at the end of the project and each iteration. The project or iteration will not be considered complete, and Mn/DOT will not pay for any project or iteration deliverables, until acceptance criteria have been satisfied and approved by Mn/DOT.

3.2.6 Each iteration must end with a process-improvement session.

3.2.7 Each iteration must include:

- Completion of requirements elicitation, specification and analysis for the requirements to be developed and/or implemented by the iteration
- Design
- Construction/Development (as applicable)
- Testing
- Documentation, training, etc. as needed
- Deployment (as applicable)

3.3 **Requirements Management**

The Selected Responder must ensure that:

3.3.1 All requirements are recorded using appropriate modeling or documentation technique.

3.3.2 All requirements are of an acceptable level of quality as explicitly defined in the requirements management plan.

3.3.3 All requirements are uniquely identified.

3.3.4 All requirements are testable.

3.3.5 All requirements, for the implementation, will be traced to at least one test case (or document review as applicable) using a traceability matrix.

3.4 **Architecture**

The Selected Responder must:

3.4.1 Complete the mobile application database design (diagram) showing tables, relationships, and attributes.

3.4.2 Complete the system meta-data dictionary of all entities/ tables and attributes/ columns including definitions in business language.

3.4.3 Complete the screen-to-table mapping document to show the mapping of database data elements to application screens and reports.

3.4.4 Develop working prototypes of user interface screens.

3.4.5 Conduct design reviews with Mn/DOT technical and business experts.

3.4.6 Present the mobile application design to the Mn/DOT application system architect, report developer, and database administrator (DBA) for review and approval. The design must not be implemented without these approvals.

3.4.7 Develop whatever UML diagrams will be required to document the proposed design for the more complex requirements.

3.4.8 Deliver a supportable web services enabled Interface design and standard that could be implemented at runtime, is open, modular, and not tightly coupled with a COTS application. The Interface must be designed to allow dynamic code injection to the web services to allow for business systems to be connected in the future to the Mobile asset database.

3.5 **Development**

For the implementation, the Selected Responder must:

3.5.1 Provide and maintain application source code files, configuration files, developer documentation, and related artifacts as applicable in Mn/DOT's Concurrent Versioning System (CVS).

3.5.2 Check code into Mn/DOT's CVS at a frequency determined by the Mn/DOT Project Manager.
3.5.3 Provide database scripts.
3.5.4 Code, test, debug, document, and implement the application using the business rules and requirements specifications and according to the approved design and technical models.
3.5.5 Document the application classes.
3.5.6 Implement Oracle tables according to approved design and technical models.

3.6 **Testing**

For the implementation, the Selected Responder must:

3.6.1 Perform unit, integration, regression and system testing on the template application and document the results for each mobile environment.
3.6.2 Document, resolve, and re-test all defects identified during testing.
3.6.3 Provide defect reports at a frequency agreed upon by the Mn/DOT Project Manager and the Selected Vendor, not to exceed monthly. (See the Reporting Requirements section below)
3.6.4 Document and maintain the status of system defects, change requests, and related artifacts in JIRA, Mn/DOT's defect tracking tool.
3.6.5 Develop and maintain a detailed test plan, a test inventory, detailed test cases, defect reports, test logs, and test reports.

3.7 **Iteration Specific Project Requirements**

The Selected Responder must:

3.7.1 Deliver an updated project schedule that has been detailed and adjusted for the current iteration.
3.7.2 Develop refined prototypes of any GUI screens that will be implemented by the current iteration
3.7.3 Develop business test scenarios and system test cases to be executed for the current iteration.
3.7.4 Deliver metadata, an updated data dictionary, and database diagram for any Oracle database deliverable(s).
3.7.5 Deliver a template application installed in the system test environment prior to system, regression, and user acceptance testing.
3.7.6 Report all defects identified and fixed during the current iteration.
3.7.7 Coordinate iteration design with Mn/DOT's report developer if the project includes any reports other than the Project Management reports.
3.7.8 Deliver, at the end of each iteration, a deployable, testable build that includes all previous builds, as applicable.

3.8 **Implementation**

For the implementation the Selected Responder must:

3.8.1 Develop and implement application deployment plans that do not adversely affect other Mn/DOT systems.
3.8.2 Provide release notes and an installation guide, as applicable. Release notes must include what is being delivered (requirements, change requests, defects, etc.) and all known issues. Installation guides must be specific to the development, test, and production environments.
3.8.3 Support the deployment of the template mobile applications into Mn/DOT's development and test environments prior to system and end-user acceptance testing (UAT).
3.8.4 Deliver builds with no known defects prior to UAT.
3.8.5 Deliver builds with <u>ALL</u> defects resolved prior to deployment to the production environment.

3.8.6 Support the deployment of the template mobile applications into the Mn/DOT production environment after successful completion of testing in the test environment.

3.9 **Maintenance**

For the implementation, the Selected Responder must:

3.9.1 Develop the applicable information technology (IT) support documentation that includes: technical architecture, database and application design, installation procedures and other information required to support the template application.

3.9.2 Provide scripts to build, test, and deploy the template mobile applications (including database objects).

3.9.3 Provide any updates to documentation required by new features or enhancements.

3.10 **Risk Management**

The Selected Responder must develop and execute a risk management plan that must ensure:

3.10.1 All risks are recorded in a single risk management repository.

3.10.2 There is a documented mitigation plan for each risk.

3.10.3 All risks are communicated clearly to the Mn/DOT Project Manager is a timely manner.

3.10.4 The statuses of all risks can be tracked.

3.10.5 Risk mitigation or prevention is carried out appropriately.

3.11 **Issue Management**

For all project issues, the Selected Responder must ensure that:

3.11.1 All issues are recorded the Mn/DOT JIRA application.

3.11.2 The statuses of all issues can be tracked.

3.11.3 All issues are communicated clearly to the Mn/DOT Project Manager is a timely manner.

3.11.4 All issues are resolved appropriately and in a timely manner.

3.12 **Configuration Management**

For the implementation, ensure that:

3.12.1 All deliverables are identified that are to be placed under configuration management.

3.12.2 Build and deployment processes are clearly defined.

3.12.3 As applicable, provide Ant or Maven build script that builds the project without any integrated development environment (IDE) dependencies.

3.12.4 The project must follow Mn/DOT's Information Technology Infrastructure Library (ITIL) change and configuration management process before being put into production.

3.13 **Quality Assurance and Test Management**

The Selected Responder must develop and execute quality assurance and test management plans that focus on the verification, validation, and testing of documents, models, test cases, source code as applicable, for the implementations. These plans need not be extensive, but must address all items in this section.

3.13.1 These plans must ensure:

- All deliverables are identified that are to be covered by the quality assurance and test plans
- Appropriate techniques such as reviews, inspections, walk-throughs, and testing must be used to ensure the quality of deliverables
- Quality criteria must be developed for all deliverables and associated tasks
- All deliverables must meet quality criteria before being accepted and approved by Mn/DOT

- All defects are recorded in a single defect repository
- All system testing, user acceptance testing, and regression testing must take place in a Mn/DOT test environment.
- All defects are uniquely identified
- The statuses of all defects can be tracked
- The project has a defects manager
- All defects are fixed in a timely manner
- Measurements based on the following must be collected and reported
  - Number of defects and their statuses
  - Number of reviews, test cases, etc. and their statuses

3.13.2 The Selected Responder must:

- Track all defects using Mn/DOT's JIRA bug tracking software.
- Complete unit testing, integration testing, system testing, user acceptance testing, and regression testing.
- Develop a test inventory (list of all tests prioritized by Mn/DOT's priorities).
- Use a risk-based approach to prioritize testing and quality assurance activities based on Mn/DOT's priorities.
- Produce periodic reports on testing progress that show test cases planned versus test cases passed and failed.
- Provide test cases to Mn/DOT far enough in advance of testing for Mn/DOT to review and approve all test cases.
- Report all defects discovered during system, user acceptance, and regression testing in JIRA.
- Provide periodic reports on defect status including number of defects open, closed, resolved, etc.
- Fix all reported defects – at no extra cost – before the final system is accepted and deployed.
- Deliver a final testing report using the template [or a variation approved by the Mn/DOT Project Manager provided in Mn/DOT's QA/QC Manual.

3.14 **Reporting Requirements**

The Selected Responder must provide a report including the information below. The frequency and format of the report will be agreed upon by Mn/DOT and the Selected Responder, with Mn/DOT having final approval.

3.14.1 Project Management

- Planned schedule versus actual schedule including schedule variance
- Planned budget versus actual budget including budget variance
- QA and Testing progress reports
- Review progress reports
- Defect reports

3.14.2 Requirements

- Numbers of requirements and their statuses

3.14.3 Changes and Issues

- Numbers of change requests and their statuses
- Number of issues and their statuses

3.14.4 Risks

- Top Ten Risk List
- Number of risks and their statuses

## 4 Project Acceptance and Satisfaction Criteria

Acceptance criteria for the project as a whole and for each iteration will be developed with the full participation of the business and technical team and end-users. Such criteria will be established at the start of the iteration and/or project and formally assessed at the end of the iteration and/or project. The iteration and/or project will not be considered payable until acceptance criteria have been satisfied and approved by Mn/DOT.

## 5 Tasks and Deliverables (all deliverables are subject to Mn/DOT review and approval).

The Selected Responder will be responsible for client satisfaction and successful completion and delivery of all the tasks and/or deliverables in the table below.

| TASK | DELIVERABLE |
|---|---|
| Business & Technical Requirements for Mobile Data | • Business & Technical Requirements Specification (Functional & Non-functional Requirements)<br>• Requirements Traceability Matrices |
| Business & Technical Requirements for Mapping Grade Data | • Business & Technical Requirements Specification (Functional & Non-functional Requirements) and determine positional accuracy requirements for Mn/DOT assets for field data collection at different mapping grade levels per asset category.<br>• Requirements Traceability Matrices |
| Architecture: Common mobile data collection framework (Mobile Data) | • Architecture including:<br>o Data Model(s)<br>o Standard Software Platforms (Software)<br>o Standard Specifications for Mobile Data Collection Devices (Hardware)<br>o Common Bi-Directional Interface for Data (Translating Tool)<br>o Workflows/Methods/Processes<br>o User Best Practices<br>o Departmental Policies and Guidelines<br>o Guidance and Support Mechanisms |
| Architecture: Common mobile data collection framework (Mapping Grade Data) | • Architecture including:<br>o Data Model(s)<br>o Standard Software Platforms (Software)<br>o Standard Specifications for Mobile Data Collection Devices (Hardware)<br>o Common Bi-Directional Interface for Data (Translating Tool)<br>o Workflows/Methods/Processes<br>o User Best Practices<br>o Departmental Policies and Guidelines |

| | |
|---|---|
| | o Guidance and Support Mechanisms |
| Data (Mobile Data) | • Flexible data structure and a build database.<br>• Data transfer specifications for communication protocols (APIs) between the mobile field data collections system and other business systems, including legacy applications. Include protocols and processes.<br>• Specifications for business systems to input its data structure to a mobile environment.<br>• Develop a specification for how we publish our data to be consumable by a mobile device for each primary targeted architecture. |
| Data (Mapping Grade Data) | • Develop data transfer specifications for communication protocols (APIs) between the mobile field data collections system and other business systems, including legacy applications. Include protocols and processes.<br>• Develop specifications for business systems to input its data structure to a mobile environment.<br>• Define and deliver a flexible data structure that is easily selectable and deployable to mobile applications, and build the database.<br>• Develop a specification for how we publish our data to be consumable by a mobile device for each primary targeted architecture.<br>• Develop standards for accuracy and precision needed for mapping grade data. |
| Mobile Applications (Mobile Data) | • Define product specifications and identify software products meeting the needs of stakeholders for spatial and non spatial mobile data collection.<br>• Define and deliver template mobile applications that allow business areas to more rapidly collect asset information in the field. |
| Mobile Applications (Mapping Grade Data) | • Define product specifications and identify software products meeting the needs of stakeholders for spatial and non spatial mobile data collection.<br>• Define and deliver template mobile applications that allow business areas to more rapidly collect asset information in the field. |

| | |
|---|---|
| Hardware (Mobile Data) | • Develop standard specifications for mobile data collection devices to meet business area needs and spatial data accuracy requirements for non-mapping grade data and for mapping grade data. |
| Hardware (Mapping Grade Data) | • For mapping grade data, identify equipment capable of providing the accuracy needs and recommendations for standard equipment bundles for field data collection. This includes SBAS (Satellite Based Augmented Signal Correction (WAAS, EGNOS)), VRS from Mn/DOT's own CORS system, Beacon, Post Processed, or combination of the above. This also includes supplemental equipment such as range finders |
| Workflows/Methods/Process (Mobile Data) | • Develop common business workflows/methods/processes to rapidly develop and deploy new field applications and to add new field assets to the database.<br>• Develop remote data collection workflows, guidelines, and standards for disconnected and connected editing. |
| Workflows/Methods/Process (Mapping Grade Data) | • Develop common business workflows/methods/processes to rapidly develop and deploy new field applications and to add new field assets to the database.<br>• Develop remote data collection workflows, guidelines, and standards for disconnected and connected editing. |
| User Best Practices (Mobile Data) | • Provide user best practices for mobile data collection and associated tools, workflows, methods, processes, application workflow, etc. |
| User Best Practices (Mapping Grade Data) | • Provide user best practices for mobile data collection and associated tools, workflows, methods, processes, application workflow, etc. |
| Departmental Policies & Guidelines (Mobile Data) | • Create Departmental policies and guidelines for mobile data collection hardware, software, and data to ensure data collection is consistent with established architecture and standards. |
| Departmental Policies & Guidelines (Mapping Grade Data) | • Create Departmental policies and guidelines for mobile data collection hardware, software, and data to ensure data collection is consistent with established architecture and standards. |
| Guidance and Support Mechanisms (Mobile Data) | • Guidance and support mechanisms required to implement and maintain the system. |
| Guidance and Support Mechanisms (Mapping Grade Data) | • Guidance and support mechanisms required to implement and maintain the system. |
| Knowledge Transfer (Mobile Data) | • Knowledge Transfer Plan<br>• Mobile Framework Documentation covering all parts of the framework. |

| | |
|---|---|
| | • Complete presenting training that provided Mn/DOT staff with the skills and knowledge to successfully implement all project deliverables. This training followed the project approach for training in sections 3.9 and 3.16. |
| Knowledge Transfer (Mapping Grade Data) | • Knowledge Transfer Plan<br>• Mobile Framework Documentation covering all parts of the framework.<br>• Complete presenting training that provided Mn/DOT staff with the skills and knowledge to successfully implement all project deliverables. This training followed the project approach for training in sections 3.9 and 3.16. |
| Implementation (Mobile Data) | • All project deliverables are successfully deployed at Mn/DOT for the business case(s) identified by Mn/DOT to allow for immediate use, including re-usable template application(s), tools, methods/process documentation, departmental standards and guidelines, user best practices, guidance and support mechanisms and documentation, etc., following the project approach for deployment in section 3.9. |
| Implementation (Mapping Grade Data) | • All project deliverables are successfully deployed at Mn/DOT for the business case(s) identified by Mn/DOT to allow for immediate use, including re-usable template application(s), tools, methods/process documentation, departmental standards and guidelines, user best practices, guidance and support mechanisms and documentation, etc., following the project approach for deployment in section 3.9. |

| | |
|---|---|
| Project Management | • Project Plan including:<br>o Project Schedule<br>o Risk Management<br>o Configuration Management<br>o Change Management<br>o Issue Management<br>o Communications Management<br>o Staffing<br>• Weekly Project Status Report or meeting agenda and materials including the following may be requested by Mn/DOT:<br>o What was completed in the past week<br>o Major Issues<br>o Change Requests<br>o What is planned to complete in the next week<br>• Monthly Project Status Report<br>o What was completed in the past month<br>o Major Issues<br>o Change Requests<br>o What is planned to complete in the next month<br>• Iteration Project Schedules<br>• Project Acceptance Criteria<br>• Iteration Acceptance Criteria |
| Quality Assurance & Testing | • Quality Assurance Plan<br>• Test Plan<br>• Test Inventory<br>• Test Cases<br>• Test Log<br>• Defect Reports<br>• Test Reports |
| Warranty | • Warranty Documentation |

## 6 Project Constraints

The Selected Responder will:

6.1 Work on-site part-time as follows:

- **Requirements Interviews:** On-site
- **Requirements Development/Documentation:** Off-site
- **Design:** Off-site (with on-site work as needed by the Selected Vendor Designers or Mn/DOT staff providing information/assistance for the design work)
- **Implementation:** On-site.

6.2 Provide deliverables that satisfy all specified requirements.

6.3 Provide deliverables that meet all standards and processes identified in this SOW.

6.4 Maintain sufficient resources (staff) to stay on schedule and support change management incidents. Change management staff must be available for potential meetings with Mn/DOT staff.

6.5 Submit a best practice work plan/schedule, and estimated durations to complete each deliverable with response.

6.6 Provide training and knowledge transfer to key Mn/DOT staff as specified in this SOW.

6.7 Adhere to Mn/DOT's testing and acceptance criteria.

6.8 Report all work plans and completed requirements to the Mn/DOT Project Manager.
6.9 Complete unit and integration on the implementation deliverables and as applicable on the framework deliverables.
6.10 Complete regression and system testing in a Mn/DOT test environment as applicable on the implementation deliverables and as applicable on the framework deliverables.
6.11 Repair all defects including requirements defects, design defects, and code defects on the implementation deliverables and as applicable on the framework deliverables.
6.12 Provide a one year warranty against defects, and correct any defects found within the warranty period.
6.13 Ensure compliance with the Statewide Enterprise Architecture.
6.14 Ensure compliance with Statewide Project Management Methodology.
6.15 Ensure compliance with applicable industry/agency standards.

## 7 Project Environment (Mn/DOT Resources)

Mn/DOT will provide technical and business resources to assist with the project as follows:

7.1 Mn/DOT's Project Manager will be Susan Bousquet. For all inquiries regarding this SOW contact the Mn/DOT Contract Administrator, Melissa McGinnis at (651) 366-4644. Contact with any other Mn/DOT personnel regarding this SOW may result in disqualification.

7.1.1 Provide a core project team including:
7.1.2 Project Sponsor
7.1.3 Project Manager
7.1.4 Project Architects
7.1.5 Key Stakeholders
7.1.6 Business Subject Matter Experts
7.1.7 Senior Technical Lead
7.1.8 Database Analyst

7.2 Mn/DOT, for the various phases of the project and as determined applicable to the project by the Mn/DOT Project Manager, will provide the following:

7.2.1 General
- Provide access to the Mn/DOT JIRA issue/change tracking system.
- Provide access to the CVS configuration management system.
- Manage project scope, cost and schedule.
- Provide remote access to Mn/DOT as necessary.

7.2.2 Analysis
- Provide information on current system business rules and interfacing external systems.
- Provide applicable deliverable templates and guidelines.
- Review and approve financial requirements.
- Provide MDGPA review.

7.2.3 Design
- Review and approve design and prototype screens, all new application code, and any financial component design.

7.2.4 Testing
- Develop acceptance test criteria.

- Perform and document user acceptance testing.
- Identify defects and change requests during acceptance testing.
- Provide on-site test lab facility as applicable.
- Test financial components.

7.2.5 Data

- Provide information on applicable current and legacy database(s).
- Assist in database design and implementation.
- Provide existing data dictionary (system meta-data dictionary of all entities/ tables and attributes/ columns).
- Provide Business Experts to perform cleansing of current system data during the data migration process.

7.2.6 Infrastructure

- Build and maintain applicable development, system test, and production hardware, operating system, and database infrastructure.
- Provide database design and development tool access (Oracle/Toad), and application development tool licenses (MyEclipse).
- Provide all hardware and software licenses for the production environment.
- Maintain all applicable computer and network hardware after the expiration of the system warranty period.

7.2.7 Reporting

- Provide Crystal Enterprise report web page and report publishing.
- Code, test, and debug reports using Crystal Reports and deploy in a Crystal Enterprise environment.

## 8 Quality Criteria

Mn/DOT will accept a deliverable only when and if it meets all of the following quality criteria:

8.1 All deliverables must be successfully completed and delivered to the State within the timeframe agreed upon and approved by the State's Project Manager.

8.2 The completed deliverables must be reviewed by the State and approved by the State's Project Manager.

8.3 All deliverables must be successfully tested and tests passed prior to implementation. Testing results must be reviewed by the State project staff and signed off by the State's Project Manager.

8.4 Mn/DOT IT staff must be able to understand and apply all deliverables within the current Mn/DOT environment with a reasonable expenditure of effort.

8.5 Upon delivery of all deliverables, Mn/DOT must be able to use these deliverables to produce a mobile business application within days or weeks rather than months.

## 9 Required Skills

Required minimum qualifications are shown in the following table. The proposal must specifically indicate how members of the Responders team meet these minimum qualifications. This portion of the proposal review will be conducted on a pass/fail basis. If Mn/DOT determines, in its sole discretion, that the Responder fails to meet one or more of these requirements (or that the Responder has not submitted

sufficient information to make the pass/fail determination), then the proposal will be eliminated from further review.

| | Required Skill Type | Minimum Number of Years/Project Experience |
|---|---|---|
| 9.1 | • Systems Analysis | 4 years |
| 9.2 | • Mobile data collection architecture.<br>• Provide a description of at least 3 mobile data collection architecture projects you have completed, and the role(s) of each resource provided for these projects. | At least 3 projects |
| 9.3 | • Mobile web applications and services.<br>• Provide a description of at least 3 mobile web applications and web services developed and implemented, and the role(s) of each resource provided for these projects. | At least 3projects |
| 9.4 | • Mobile applications/services<br>• Provide a description of at least 3 mobile applications developed and implemented, and the role(s) of each resource provided for these projects. | At least 3 projects |
| 9.5 | • Android experience<br>• iOS experience<br>• Windows mobile experience (version 7 preferred)<br>• Blackberry<br>• S60 Environment (Nokia) | Experience with at least one |
| 9.6 | • Database design<br>• Updating databases with connected mobile applications, and disconnected mobile applications with a temporary database<br>• Use of Oracle 10g database management system | 4 years |
| 9.7 | • Provide a description of at least 3 mobile applications and services developed and implemented, and the role(s) of each resource provided for these projects. | At least 3 projects |
| 9.8 | • Securing/accessing data with connected and disconnected mobile applications and data<br>• Provide a description of at least 3 mobile data collection projects completed, and the role(s) of each resource provided for these projects. | At least 3 projects |
| 9.9 | • ESRI ArcGIS Mobile<br>• ESRI ArcGIS Server for .NET<br>• ESRI ArcPad<br>• Oracle database with an ArcSDE persistent connection<br>• Trimble | 4 years |

## 10 Desired Skills

Mn/DOT desires a project team with the skills shown in the table below. The extent to which the Responder meets or exceeds the desired skills will be included as part of the qualitative evaluation of the proposal.

| | Desired Skill Type | Desired Years Experience |
|---|---|---|
| 10.1 | • Mobile data collection architecture. | 4 years |

| | | |
|---|---|---|
| | • Provide a description of more than 3 mobile data collection architecture projects you have completed | |
| 10.2 | • Mobile web applications and services.<br>• Provide a description of more than 3 mobile web applications and web services developed and implemented, and the role(s) of each resource provided for these projects. | 3 years |
| 10.3 | • Mobile applications/services<br>• Provide a description of more than 3 mobile applications developed and implemented, and the role(s) of each resource provided for these projects. | 2 years |
| 10.4 | • Provide a description of more than 3 mobile applications and services developed and implemented, and the role(s) of each resource provided for these projects. | 3 years |
| 10.5 | • Securing/accessing data with connected and disconnected mobile applications and data<br>• Provide a description of more than 3 mobile data collection projects completed, and the role(s) of each resource provided for these projects. | 3 years |
| 10.6 | • Android experience | 1.5 years |
| 10.7 | • iOS experience | 2 years |
| 10.8 | • Windows OS (prefer version 7) | 3 years |
| 10.9 | • .NET | Any amount |
| 10.10 | • Java | Any amount |
| 10.11 | • APIs | Any amount |
| 10.12 | • SDKs | Any amount |

**11 Project Milestones and Schedule**

11.1 Project start date: **01/11**

11.2 Key deliverable dates:

11.2.1 Deliverables for Mobile Data

- Business Analysis/Requirements DUE DATE: 6/30/11
- Conceptual Design & Recommendations DUE DATE: 6/3/11
- Implementation DUE DATE: 4/30/12

11.2.2 Deliverables for Mapping Grade Data

- Business Analysis/Requirements DUE DATE: 6/30/11
- Conceptual Design & Recommendations DUE DATE: 6/30/11
- Implementation DUE DATE: 4/30/12

11.3 End date: **4/30/12**

**12 Process Schedule**

| | | |
|---|---|---|
| 12.1 | Deadline for Questions | 01/20/2011 2:00 PM Central Standard Time |
| 12.2 | Posted Response to Questions | 01/24/2011 2:00 PM Central Standard Time |
| 12.3 | Proposals due | 01/28/2011 2:00 PM Central Standard Time |
| 12.4 | Anticipated proposal evaluation begins | 01/31/2011 |
| 12.5 | Anticipated proposal evaluation & decision | 02/15/2011 |

## 13 Questions

All questions regarding this SOW must be addressed to the Mn/DOT Contract Administrator listed below. Proposers may not discuss the content of this SOW with other Mn/DOT staff. Any questions regarding this SOW must be received via e-mail by 01/20/2011, 2:00PM Central Standard Time.

Contract Administrator: Melissa McGinnis
Email Address: melissa.mcginnis@state.mn.us

It is anticipated that questions and answers will be posted on the Office of Enterprise Technology's web site by 01/24/2011, 2:00pm Central Standard Time (www.ot.state.mn.us). Note that questions may be posted verbatim as submitted.

## 14 Indemnification

In the performance of this contract by Contractor, or Contractor's agents or employees, the Contractor must indemnify, save, and hold harmless the State, its agents, and employees, from any claims or causes of action, including attorney's fees incurred by the state, to the extent caused by Contractor's:

1) Intentional, willful, or negligent acts or omissions; or
2) Actions that give rise to strict liability; or
3) Breach of contract or warranty.

The indemnification obligations of this section do not apply in the event the claim or cause of action is the result of the State's sole negligence. This clause will not be construed to bar any legal remedies the Contractor may have for the State's failure to fulfill its obligation under this contract.

The "Standard Indemnification Clause" (see above) will apply to this project and will be incorporated into the work order issued for this project. No exceptions to, or deviations from, this clause will be permitted. Do not submit a proposal if you cannot accept this indemnification clause. Proposals which the State determines, in its sole discretion, indicate non-acceptance of this indemnification clause, will be rejected by the State.

## 15 SOW Evaluation Process

Mn/DOT representatives will evaluate proposals received by the deadline. Proposals will be evaluated on a "Best Value" basis of 70% qualifications and 30% cost considerations. The review committee will not open the cost proposals until after the qualifications points have been awarded.

The selection process being used for this project involves a two step process. Step one will include the pass/fail assessment and a qualitative evaluation of Contractors' technical proposal. Step Two will be an analysis of the cost proposal.

Mn/DOT will review proposals according to the following criteria:

| | |
|---|---|
| • Company background and expertise | 5% |
| • Experience of personnel assigned to this project. This includes the extent to which they meet the Desired Skills, references provided and previous performance reviews | 25% |
| • Proposed work plan, including the apparent ability to complete project on time and on budget | 30% |
| • Extent to which services will be performed within the U.S.* | 10% |
| • Cost | 30% |

Mn/DOT reserves the right to check references and to review previous performance reviews for work performed for Mn/DOT or other state agencies, and to take such references and reviews into account for consultant selection purposes.

The following contains additional information describing the proposal evaluation process:

**Step One**

In step one the proposals will first be reviewed to verify whether the proposer meets the "Required Skills". Proposals receiving a "fail" on one or more of the required skills will not be reviewed further. Proposals which pass the Required Skills review will then be scored on the non-cost and non-interview factors listed above.

**Step Two**

Cost proposal will be evaluated and scored in accordance with the percentage listed above. Cost will not be revealed to selection committee members until after the technical scoring has been completed.

**16 Response Requirements**

16.1 Introduction.

16.2 Company overview.

16.3 Project overview.

16.4 An outline of the responder's background and experience with examples of similar work done and a list of personnel who will conduct the project, detailing their training and work experience. No change in personnel assigned to the project will be permitted without the written approval of Mn/DOT's Project Manager.

16.5 Detailed response to "Project Approach".
Explain how the responder will approach their participation in the project. This includes:

16.5.1 Organization and staffing. Include staff qualifications in a chart **AND** resumes that will allow Mn/DOT to easily determine if assigned key staff meet the required skills and the extent to which assigned staff meet or exceed the desired skills. **The chart must align with the skills identified in the resumes.**

| **Required / Desired Skill type** | **Personnel/ Years of Experience** | **Project(s) worked on demonstrating these skills** | **Reference (name, company, phone number)** |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

16.5.2 A detailed work plan that will identify the major tasks to be accomplished and be used as a scheduling and management tool, as well as the basis for invoicing. The work plan must present the responder's approach, task breakdown, deliverable due dates and personnel working on the project and the hours assigned to each individual to reach the project results. The work plan should also include a realistic plan to meet the projects target completion date.

16.5.3 Contract/change management procedures.

16.5.4 Project management (e.g. quality management, risk assessment/management, etc.).

16.5.5 Documentation of progress such as status reports.

16.5.6 Provide three references for similar work. Include contact person, firm, email address and phone number along with a brief description of the duties that the key personnel performed for that project.

16.6 Submit a cost proposal in a separate sealed envelope. Rates proposed may not exceed the rates approved under this program. Cost proposal must include the number of anticipated hours,

classifications of personnel, personnel hourly rates and a total project cost. If direct expenses are anticipated they must be detailed in the cost proposal. **The cost estimate must correspond to the detailed work plan and schedule that includes time estimates, associated deliverables, and staff assigned to each task.**

16.7 Required forms to be returned or additional provisions that must be included in proposal:

16.7.1 Location of Service Disclosure Form.

16.7.2 Conflict of Interest Form

Proposer must provide a list of all entities with which it has relationships that create, or appear to create, a conflict of interest with the work that is contemplated in this request for proposals. The list should indicate the name of the entity, the relationship, and a discussion of the conflict.

The proposer warrants that, to the best of its knowledge and belief, and except as otherwise disclosed, there are no relevant facts or circumstances which could give rise to organizational conflicts of interest. An organizational conflict of interest exists when, because of existing or planned activities or because of relationships with other persons, a proposer is unable or potentially unable to render impartial assistance or advice to Mn/DOT, or the proposer's objectivity in performing the contract work is or might be otherwise impaired, or the proposer has an unfair competitive advantage. The proposer agrees that, if after award, an organizational conflict of interest is discovered, an immediate and full disclosure in writing must be made to Mn/DOT which must include a description of the action which the proposer has taken or proposes to take to avoid or mitigate such conflicts. If an organization conflict of interest is determined to exist, Mn/DOT may, at its discretion, cancel the contract. In the event the proposer was aware of an organizational conflict of interest prior to the award of the contract and did not disclose the conflict to Mn/DOT, Mn/DOT may terminate the contract for default. The provisions of this clause must be included in all subcontracts for work to be performed similar to the service provided by the prime contractor, and the terms "contract," "contractor," and "contracting officer" modified appropriately to preserve Mn/DOT's rights. Proposers must complete the attached "Conflict of Interest Checklist and Disclosure Form" and submit it along with the response, but not as a part of the response.

16.7.3 Affidavit of non-collusion

Proposers must complete the attached "Affidavit of Noncollusion" and include it with the response. The successful proposer will be required to submit acceptable evidence of compliance with workers' compensation insurance coverage requirements prior to execution of the Contract.

16.7.4 Immigration Status Certification Form

For all Contracts estimated to be in excess of $50,000.00, responders are required to complete the attached "Immigration Status Certification Form" page and include it with the response.

16.7.5 Affirmative Action Certification

For all Contracts estimated to be in excess of $100,000.00, responders are required to complete the attached "Affirmative Action Certification" page and include it with the response.

16.7.6 Veteran-Owned/Service-Disabled Veteran-Owned Preference Forms.

Proposers must complete and submit this form if claiming preference as a qualified proposer.

## 17 Proposal Submission Instructions

Submit 7 copies of the response. Responses are to be submitted in a mailing envelope or package, clearly marked "Proposal" on the outside. Cost proposals are to be submitted in a separate sealed envelope. An authorized member of the firm must sign each copy of the response in ink.

All responses must be sent to:

**Melissa McGinnis, Contract Administrator**
**Minnesota Department of Transportation**
**395 John Ireland Boulevard**
**Consultant Services Section, Mail Stop 680**
**St. Paul, Minnesota 55155**

All responses must be received not later than 2:00 p.m. Central Standard Time on 01/28/2011, as indicated by the time stamp made by the Contract Administrator. **Please note that Mn/DOT Offices have implemented security measures**. These procedures do not allow non-Mn/DOT employees to have access to the elevators or the stairs. You should plan enough time and follow these instructions for drop-off:

- Enter through the Rice Street side of the Central Office building (1st Floor).
- Once you enter through the doors, you should proceed to the first floor Information Desk.
- **Proposals are accepted at the first floor Information Desk only**. The receptionist will call the Contract Administrator to come down and to time stamp the proposal. Please keep in mind Mn/DOT is very strict on the proposal deadline. Proposals will not be accepted after 2:00pm.

## 18 General Requirements

### 18.1 Proposal Contents

By submission of a proposal, Proposer warrants that the information provided is true, correct and reliable for purposes of evaluation for potential award of a work order. The submission of inaccurate or misleading information may be grounds for disqualification from the award as well as subject the proposer to suspension or debarment proceedings and other remedies available at law.

### 18.2 Disposition of Responses

All materials submitted in response to this SOW will become property of the State and will become public record in accordance with Minnesota Statutes, section 13.591, after the evaluation process is completed. Pursuant to the statute, completion of the evaluation process occurs when the government entity has completed negotiating the contract with the selected Proposer. If the Proposer submits information in response to this SOW that it believes to be trade secret materials, as defined by the Minnesota Government Data Practices Act, Minn. Stat. § 13.37, the Proposer must: clearly mark all trade secret materials in its response at the time the response is submitted, include a statement with its response justifying the trade secret designation for each item, and defend any action seeking release of the materials it believes to be trade secret, and indemnify and hold harmless the State, its agents and employees, from any judgments or damages awarded against the State in favor of the party requesting the materials, and any and all costs connected with that defense. This indemnification survives the State's award of a contract. In submitting a response to this RFP, the Proposer agrees that this indemnification survives as long as the trade secret materials are in possession of the State.

Mn/DOT will not consider the prices submitted by the Proposer to be proprietary or trade secret materials.

## 19 No State Obligation

Issuance of this Statement of Work does not obligate Mn/DOT to award a contract or complete the assignment, and Mn/DOT reserves the right to cancel this solicitation if it is considered to be in its best interest. Mn/DOT reserves the right to reject any and all proposals.

**20 Veteran-owned/Service Disabled Veteran-Owned Preference**

In accordance with Laws of Minnesota, 2009, Chapter 101, Article 2, Section 56, eligible certified veteran-owned and eligible certified service-disabled veteran-owned small businesses will receive a 6 percent preference in the evaluation of their proposal.

Eligible veteran-owned and eligible service-disabled veteran-owned small businesses should complete the Veteran-Owned/Service Disabled Veteran-Owned Preference Form in this solicitation, and include the required documentation. Only eligible, certified, veteran-owned/service disabled small businesses that provide the required documentation, per the form, will be given the preference.

Eligible veteran-owned and eligible service-disabled veteran-owned small businesses must be **currently** certified by the U.S. Department of Veterans Affairs prior to the solicitation opening date and time to receive the preference.

Information regarding certification by the United States Department of Veterans Affairs may be found at http://www.vetbiz.gov.

**21 IT Accessibility Standards**

Responses to this solicitation must comply with the Minnesota IT Accessibility Standards effective September 1, 2010, which entails, in part, the Web Content Accessibility Guidelines (WCAG) 2.0 (Level AA) and Section 508 Subparts A-D which can be viewed at:
http://www.mmd.admin.state.mn.us/pdf/accessibility_standard.pdf

**22 Nonvisual Access Standards**

Nonvisual access standards require:

1) The effective interactive control and use of the technology, including the operating system, applications programs, prompts, and format of the data presented, are readily achievable by nonvisual means;
2) That the nonvisual access technology must be compatible with information technology used by other individuals with whom the blind or visually impaired individual must interact;
3) That nonvisual access technology must be integrated into networks used to share communications among employees, program participants, and the public; and
4) That the nonvisual access technology must have the capability of providing equivalent access by nonvisual means to telecommunications or other interconnected network services used by persons who are not blind or visually impaired.

*The balance of this page has been intentionally left blank.*

***STATE OF MINNESOTA***
**LOCATION OF SERVICE DISCLOSURE AND CERTIFICATION**

**LOCATION OF SERVICE DISCLOSURE**

**Check all that apply:**

☐ The services to be performed under the anticipated contract as specified in our proposal will be performed ENTIRELY within the State of Minnesota.

☐ The services to be performed under the anticipated contract as specified in our proposal entail work ENTIRELY within another state within the United States.

☐ The services to be performed under the anticipated contract as specified in our proposal will be performed in part within Minnesota and in part within another state within the United States.

☐ The services to be performed under the anticipated contract as specified in our proposal DO involve work outside the United States. Below (or attached) is a description of:

- The identity of the company (identify if subcontractor) performing services outside the United States;
- The location where services under the contract will be performed; and
- The percentage of work (in dollars) as compared to the whole that will be conducted in each identified foreign location.

**CERTIFICATION**

By signing this statement, I certify that the information provided above is accurate and that the location where services have been indicated to be performed will not change during the course of the contract without prior, written approval from the State of Minnesota.

Name of Company: _____

Authorized Signature: _____

Printed Name: _____

Title: _____

Date: _____

Telephone Number: _____

Mn/DOT Contract No.: 97689

**STATE OF MINNESOTA**
**VETERAN-OWNED/SERVICE DISABLED VETERAN-OWNED PREFERENCE FORM**

In accordance with Laws of Minnesota, 2009, Chapter 101, Article 2, Section 56, eligible certified veteran-owned and eligible certified service-disabled veteran-owned small businesses will receive a 6 percent preference in the evaluation of their proposal.

Eligible veteran-owned and eligible service-disabled veteran-owned small businesses include certified small businesses that are majority-owned and operated by either (check the box that applies and attach the certification documents required with your response to this solicitation):

☐ (1) recently separated veterans, who are veterans as defined in Minn. Stat. §197.447, who have served in active military service, at any time on or after September 11, 2001, and who have been discharged under honorable conditions from active service, as indicated by the person's United States Department of Defense form DD-214 or by the commissioner of veterans affairs; or

Required Documentation:

- certification by the United States Department of Veterans Affairs as a veteran-owned small business
- discharge form (DD-214) dated on or after September 11, 2001 with condition honorable

☐ (2) Veterans who are veterans as defined in Minn. Stat. § 197.447, with service-connected disabilities, as determined at any time by the United States Department of Veterans Affairs.

Required Documentation:

- certification by the United States Department of Veterans Affairs as a service-disabled veteran-owned small business.

Eligible veteran-owned and eligible service-disabled veteran-owned small businesses must be **currently** certified by the U.S. Department of Veterans Affairs prior to the solicitation opening date and time to receive the preference.

Information regarding certification by the United States Department of Veterans Affairs may be found at http://www.vetbiz.gov .

You must submit this form and the documentation required above with your response in order to be considered for this preference.

**CONFLICT OF INTEREST CHECKLIST AND DISCLOSURE FORM**

**Purpose of this Checklist**. This checklist is provided to assist proposers in screening for potential organizational conflicts of interest. The checklist is for the internal use of proposers and does not need to be submitted to Mn/DOT, however, the Disclosure of Potential Conflict of Interest form should be submitted in a separate envelope along with your proposal.

**Definition of "Proposer".** As used herein, the word "Proposer" includes both the prime contractor and all proposed subcontractors.

**Checklist is Not Exclusive.** Please note that this checklist serves as a guide only, and that there may be additional potential conflict situations not covered by this checklist. If a proposer determines a potential conflict of interest exists that is not covered by this checklist, that potential conflict must still be disclosed.

**Use of the Disclosure Form.** A proposer must complete the attached disclosure form and submit it with their Proposal (or separately as directed by Mn/DOT for projects not awarded through a competitive solicitation). If a proposer determines a potential conflict of interest exists, it must disclose the potential conflict to Mn/DOT; however, such a disclosure will not necessarily disqualify a proposer from being awarded a Contract. To avoid any unfair "taint" of the selection process, the disclosure form should be provided separate from the bound proposal, and it will not be provided to selection committee members. Mn/DOT Contract Management personnel will review the disclosure and the appropriateness of the proposed mitigation measures to determine if the proposer may be awarded the contract notwithstanding the potential conflict. Mn/DOT Contract Management personnel may consult with Mn/DOT's Project Manager and Department of Administration personnel. By statute, resolution of conflict of interest issues is ultimately at the sole discretion of the Commissioner of Administration.

**Material Representation**. The proposer is required to submit the attached disclosure form either declaring, to the best of its knowledge and belief, either that no potential conflict exists, or identifying potential conflicts and proposing remedial measures to ameliorate such conflict. The proposer must also update conflict information if such information changes after the submission of the proposal. Information provided on the form will constitute a material representation as to the award of this Contract. Mn/DOT reserves the right to cancel or amend the resulting contract if the successful proposer failed to disclose a potential conflict, which it knew or should have known about, or if the proposer provided information on the disclosure form that is materially false or misleading.

**Approach to Reviewing Potential Conflicts**. Mn/DOT recognizes that proposer's must maintain business relations with other public and private sector entities in order to continue as viable businesses. Mn/DOT will take this reality into account as it evaluates the appropriateness of proposed measures to mitigate potential conflicts. It is not Mn/DOT's intent to disqualify proposers based merely on the existence of a business relationship with another entity, but rather only when such relationship causes a conflict that potentially impairs the proposer's ability to provide objective advice to Mn/DOT. Mn/DOT would seek to disqualify proposers only in those cases where a potential conflict cannot be adequately mitigated. Nevertheless, Mn/DOT must follow statutory guidance on Organizational Conflicts of Interest.

**Statutory Guidance**. Minnesota Statutes §16C.02, subd. 10 (a) places limits on state agencies ability to contract with entities having an "Organizational Conflict of Interest". For purposes of this checklist and disclosure requirement, the term "Vendor" includes "Proposer" as defined above. Pursuant to such statute, "Organizational Conflict of Interest" means that because of existing or planned activities or because of relationships with other persons: (1) the vendor is unable or potentially unable to render impartial assistance or advice to the state; (2) the vendor's objectivity in performing the contract work is or might otherwise be impaired; or (3) the vendor has an unfair advantage.

**Additional Guidance for Professionals Licensed by the Minnesota Board of Engineering**. The Minnesota Board of Engineering has established conflict of interest rules applicable to those professionals licensed by the Board (see Minnesota Rules part 1805.0300) Subpart 1 of the rule provides "A licensee shall avoid accepting a commission where

duty to the client or the public would conflict with the personal interest of the licensee or the interest of another client. Prior to accepting such employment the licensee shall disclose to a prospective client such facts as may give rise to a conflict of interest".

**An organizational conflict of interest may exist in any of the following cases:**

- ❑ The proposer, or its principals, own real property in a location where there may be a positive or adverse impact on the value of such property based on the recommendations, designs, appraisals, or other deliverables required by this Contract.
- ❑ The proposer is providing services to another governmental or private entity and the proposer knows or has reason to believe, that entity's interests are, or may be, adverse to the state's interests with respect to the specific project covered by this contract. **Comment:** the mere existence of a business relationship with another entity would not ordinarily need to be disclosed. Rather, this focuses on the nature of services commissioned by the other entity. For example, it would not be appropriate to propose on a Mn/DOT project if a local government has also retained the proposer for the purpose of persuading Mn/DOT to stop or alter the project plans.
- ❑ The Contract is for right-of-way acquisition services or related services (e.g. geotechnical exploration) and the proposer has an existing business relationship with a governmental or private entity that owns property to be acquired pursuant to the Contract.
- ❑ The proposer is providing real estate or design services to a private entity, including but not limited to developers, whom the proposer knows or has good reason to believe, own or are planning to purchase property affected by the project covered by this Contract, when the value or potential uses of such property may be affected by the proposer's performance of work pursuant to this Contract. "Property affected by the project" includes property that is in, adjacent to, or in reasonable proximity to current or potential right-of-way for the project. The value or potential uses of the private entity's property may be affected by the proposer's work pursuant to the Contract when such work involves providing recommendations for right-of-way acquisition, access control, and the design or location of frontage roads and interchanges. **Comment**: this provision does not presume proposers know or have a duty to inquire as to all of the business objectives of their clients. Rather, it seeks the disclosure of information regarding cases where the proposer has reason to believe that its performance of work under this contract may materially affect the value or viability of a project it is performing for the other entity.
- ❑ The proposer has a business arrangement with a current Mn/DOT employee or immediate family member of such employee, including promised future employment of such person, or a subcontracting arrangement with such person, when such arrangement is contingent on the proposer being awarded this Contract. This item does not apply to pre-existing employment of current or former Mn/DOT employees, or their immediate family members. **Comment:** this provision is not intended to supersede any Mn/DOT policies applicable to its own employees accepting outside employment. This provision is intended to focus on identifying situations where promises of employment have been made contingent on the outcome of this particular procurement. It is intended to avoid a situation where a proposer may have unfair access to "inside" information.
- ❑ The proposer has, in previous work for the state, been given access to "data" relevant to this procurement or this project that is classified as "private" or "nonpublic" under the Minnesota Government Data Practices Act, and such data potentially provides the proposer with an unfair advantage in preparing a proposal for this project. **Comment**: this provision will not, for example, necessarily disqualify a proposer who performed some preliminary work from obtaining a final design Contract, especially when the results of such previous work are public data available to all other proposers. Rather, it attempts to avoid an "unfair advantage" when such information cannot be provided to other potential proposers. Definitions of "government data", "public data", "non-public data" and "private data" can be found in Minnesota Statutes Chapter 13.
- ❑ The proposer has, in previous work for the state, helped create the "ground rules" for this solicitation by performing work such as: writing this solicitation, or preparing evaluation criteria or evaluation guides for this solicitation.
- ❑ The proposer, or any of its principals, because of any current or planned business arrangement, investment interest, or ownership interest in any other business, may be unable to provide objective advice to the state.

**DISCLOSURE OF POTENTIAL CONFLICT OF INTEREST**

Having had the opportunity to review the Organizational Conflict of Interest Checklist, the proposer hereby indicates that it has, to the best of its knowledge and belief:

☐ Determined that no potential organizational conflict of interest exists.

☐ Determined a potential organizational conflict of interest as follows:

Describe nature of potential conflict

Describe measures proposed to mitigate the potential conflict

Signature

Date

If a potential conflict has been identified, please provide name and phone number for a contact person authorized to discuss this disclosure form with Mn/DOT contract personnel.

Name

Phone

Mn/DOT Contract No.: 97689

STATE OF MINNESOTA
AFFIDAVIT OF NONCOLLUSION

I swear (or affirm) under the penalty of perjury:

1. That I am the Responder (if the Responder is an individual), a partner in the company (if the Responder is a partnership), or an officer or employee of the responding corporation having authority to sign on its behalf (if the Responder is a corporation);

2. That the attached proposal submitted in response to the _____ Statement of Work has been arrived at by the Responder independently and has been submitted without collusion with and without any agreement, understanding or planned common course of action with, any other Responder of materials, supplies, equipment or services described in the Request for Proposal, designed to limit fair and open competition;

3. That the contents of the proposal have not been communicated by the Responder or its employees or agents to any person not an employee or agent of the Responder and will not be communicated to any such persons prior to the official opening of the proposals; and

4. That I am fully informed regarding the accuracy of the statements made in this affidavit.

Responders' Firm Name: _____

Authorized Signature: _____

Date: _____

Subscribed and sworn to me this _____ day of _____
(day) (Month Year)

Notary Public _____

My commission expires: _____

## State of Minnesota — Immigration Status Certification

By order of the Governor (Governor's Executive Order 08-01), vendors and subcontractors MUST certify compliance with the Immigration Reform and Control Act of 1986 (8 U.S.C. 1101 et seq.) and certify use of the *E-Verify* system established by the Department of Homeland Security.

*E-Verify* program information can be found at http://www.dhs.gov/ximgtn/programs.

If any response to a solicitation is or could be in excess of $50,000.00, vendors and subcontractors must certify compliance with items 1 and 2 below. In addition, prior to the delivery of the product or initiation of services, vendors MUST obtain this certification from all subcontractors who will participate in the performance of the Contract. All subcontractor certifications must be kept on file with the Contract vendor and made available to the state upon request.

1. The company shown below is in compliance with the Immigration Reform and Control Act of 1986 in relation to all employees performing work in the United States and does not knowingly employ persons in violation of the United States immigration laws. The company shown below will obtain this certification from all subcontractors who will participate in the performance of this Contract and maintain subcontractor certifications for inspection by the state if such inspection is requested; and
2. By the date of the delivery of the product and/or performance of services, the company shown below will have implemented or will be in the process of implementing the *E-Verify* program for all newly hired employees in the United States who will perform work on behalf of the State of Minnesota.

**I certify that the company shown below is in compliance with items 1 and 2 above and that I am authorized to sign on its behalf.**

| | | | |
|---|---|---|---|
| Name of Company | | Date: | |
| Authorized Signature | | Telephone Number | |
| Printed Name: | | Title: | |

If the Contract vendor and/or the subcontractors are not in compliance with the Immigration Reform and Control Act, or knowingly employ persons in violation of the United States immigration laws, or have not begun or implemented the *E-Verify* program for all newly hired employees in support of the Contract, the state reserves the right to determine what action it may take. This action could include, but would not be limited to cancellation of the Contract, and/or suspending or debarring the Contract vendor from state purchasing.

**For assistance with the *E-Verify* Program**
Contact the National Customer Service Center (NCSC) at **1-800-375-5283** (TTY 1-800-767-1833).

**For assistance with this form, contact:**
Mail: 112 Administration Building, 50 Sherburne Avenue, St. Paul, Minnesota 55155
E-Mail: MMDHelp.Line@state.mn.us
Telephone: 651-296-2600
Persons with a hearing or speech disability may contact us by dialing 711 or 1-800-627-3529

**SAMPLE WORK ORDER LANGUAGE**

**STATE OF MINNESOTA**
**IT Professional Services Master Contract Work Order**

This work order is between the State of Minnesota, acting through its Commissioner of Transportation ("State") and [fill in name of contractor, be sure to indicate if corporation, partnership, limited liability company, sole proprietor, etc] ("Contractor"). This Work Order is issued under the authority of Master Contract T-Number 502TS, CFMS Number [fill in CFMS number from the contractor's master contract], and is subject to all provisions of the Master Contract which is incorporated by reference.

**Recitals**

1. Under Minn. Stat. § 15.061 [Insert additional statutory authorization if necessary] the State is authorized to engage such assistance as deemed necessary.
2. The State is in need of [Add brief narrative of the purpose of the contract].
3. The Contractor represents that it is duly qualified and agrees to perform all services described in this work order to the satisfaction of the State.

**Work Order**

**1 Term of Work Order; Incorporation of Exhibits; Survival of Terms**

**1.1 Effective date.** This Work Order will take effect on the date the State obtains all required signatures as required by Minn. Stat. § 16C.05, subd. 2.
***The Contractor must not begin work under this work order until it is fully executed and the Contractor has been notified by the State's Authorized Representative to begin the work.***

**1.2 Expiration date.** This Work Order will expire on [fill in date], or when all obligations have been satisfactorily fulfilled, whichever occurs first.

**1.3 Exhibits.** Exhibits [fill in, e.g. A – D] are attached and incorporated into this Work Order.

**1.4 Survival of terms.** All clauses which impose obligations continuing in their nature and which must survive in order to give effect to their meaning will survive the expiration or termination of this Work Order.

**2 Contractor's Duties**

The Contractor, who is not a state employee, will:

[Provide a <u>detailed</u> scope of services. The services must define specific duties, deliverables, and deliverable completion dates. Do not simply attach the same scope that was used in the "Statement of Work" (RFP) as a greater level of detail is needed in this work order. If using a separate attachment, use "Perform the duties specified in Exhibit A, "Scope of Services".]

**3 Consideration and Payment**

**3.1 Consideration**

The State will pay for all services performed by the Contractor under this work order as follows:

3.1.1 *Compensation.* The Contractor will be paid as follows:
[Provide a detailed explanation of how the Contractor will be paid, for example a fixed hourly rate, or a lump sum per deliverable, some examples may be:
an Hourly Rate of $_____up to maximum of ____hours, but not to exceed $_____.

a Lump Sum of $_____.]
[Rate: rates paid may not exceed the Contractor's rates specified in their Master Contract.]

3.1.2 *Travel Expenses*. Reimbursement for travel and subsistence expenses actually and necessarily incurred by Contractor, as a result of this Work Order, will be reimbursed for travel and subsistence expenses in the same manner and in no greater amount than provided in the current Minnesota Department of Transportation Travel Regulations. Contractor will not be reimbursed for travel and subsistence expenses incurred outside Minnesota unless it has received State's prior written approval for out of state travel. Minnesota will be considered the home state for determining whether travel is out of state. See Exhibit ____ for the current Minnesota Department of Transportation Reimbursement Rates for Travel Expenses.

3.1.3 *Total Obligation*. The total obligation of the State for all compensation and reimbursements to the Contractor under this Work Order will not exceed $ [fill in].

**3.2 Payment**

3.2.1 *Invoices*. State will promptly pay Contractor after Contractor presents an itemized invoice for the services actually performed and State's Authorized Representative accepts the invoiced services. Invoices must be submitted in the format prescribed in Exhibit ____ and according to the following schedule:

[INDICATE WHEN YOU WANT THE CONTRACTOR TO SUBMIT INVOICES, FOR EXAMPLE: "MONTHLY" OR "UPON COMPLETION OF SERVICES," OR IF THERE ARE SPECIFIC DELIVERABLES, LIST HOW MUCH WILL BE PAID FOR EACH DELIVERABLE. THE STATE DOES NOT PAY MERELY FOR THE PASSAGE OF TIME.]

3.2.1.1 Each invoice must contain the following information: Mn/DOT Contract Number, Mn/DOT Contract invoice number (sequentially numbered), billing address if different from business address, and Contractor's original signature attesting that the invoiced service and costs are new and that no previous charge for those services or goods has been included in any prior invoice.

3.2.1.2 Direct nonsalary costs allocable to the work under this Work Order must be itemized and supported with invoices or billing documents to show that such costs are properly allocable to the work. Direct nonsalary costs are any costs that are not the salaried costs directly related to the work of Contractor. Supporting documentation must be provided in a manner that corresponds to each direct cost.

3.2.1.3 The original of each invoice must be sent to State's Authorized Representative for review and payment. A copy of the invoice will be sent to State's Project Manager for review.

3.2.1.4 Contractor must provide, upon request of State's Authorized Representative, the following supporting documentation:

3.2.1.5 Direct salary costs of employees' time directly chargeable for the services performed under this Work Order. This must include a payroll cost breakdown identifying the name of the employee, classification, actual rate of pay, hours worked, and total payment for each invoice period; and

3.2.1.6 Signed time sheets or payroll cost breakdown for each employee listing dates and hours worked. Computer generated printouts of labor costs for the project must contain the project number, each employee's name, hourly rate, regular and overtime hours, and the dollar amount charged to the project for each pay period.

3.2.1.7 If Contractor is authorized by State to use or uses any subcontractors, Contractor must include all the above supporting documentation in any subcontractor's contract, and Contractor must make timely payments to its subcontractors. Contractor must require subcontractors' invoices to follow the same form and contain the same information as set forth above.

3.2.2 *Retainage*. Under Minnesota Statutes § 16C.08, subdivision 5(b), no more than 90% of the amount due under this Contract may be paid until State's agency head has reviewed the final product of this Contract. The balance due will be paid when State's agency head determines that Contractor has satisfactorily fulfilled all the terms of this Contract.

3.2.3 *Federal Funds*. If federal funds are used, Contractor is responsible for compliance with all federal requirements imposed on these funds and accepts full financial responsibility for any requirements imposed by Contractor's failure to comply with these federal requirements.

3.2.4 *Progress Reports*. Contractor will submit progress reports in a format and timeline designated by the State's Project Manager.

**4 Indemnification**

In the performance of this contract by Contractor, or Contractor's agents or employees, the Contractor must indemnify, save, and hold harmless the State, its agents, and employees, from any claims or causes of action, including attorney's fees incurred by the state, to the extent caused by Contractor's:

1) Intentional, willful, or negligent acts or omissions; or
2) Actions that give rise to strict liability; or
3) Breach of contract or warranty.

The indemnification obligations of this section do not apply in the event the claim or cause of action is the result of the State's sole negligence. This clause will not be construed to bar any legal remedies the Contractor may have for the State's failure to fulfill its obligation under this contract.

**5 Foreign Outsourcing**

The Contractor agrees that the disclosures and certifications made in its Location of Service Disclosure and Certification Form submitted with its proposal are true, accurate and incorporated into this work order contract by reference.

**6 Authorized Representatives**

**6.1 State's Authorized Representative.** State's Authorized Representative will be:

NAME, TITLE
ADDRESS
TELEPHONE NUMBER
FAX NUMBER
E-MAIL ADDRESS

State's Authorized Representative or his /her successor, will monitor Contractor's performance and has the authority to accept or reject the services provided under this Work Order.

**6.2 State's Project Manager**. State's Project Manager will be:

NAME, TITLE
ADDRESS
TELEPHONE NUMBER
FAX NUMBER

E-MAIL ADDRESS

State's Project Manager, or his/her successor, has the responsibility to monitor Contractor's performance and progress.
State's Project Manager will sign progress reports, review billing statements, make recommendations to State's Authorized Representative for acceptance of Contractor's good or services and make recommendations to State's Authorized Representative for certification for payment of each Invoice submitted for payment.

**6.3** **Contractor's Authorized Representative**. Contractor's Authorized Representative will be:
NAME, TITLE
ADDRESS
TELEPHONE NUMBER
FAX NUMBER
E-MAIL ADDRESS

If Contractor's Authorized Representative changes at any time during this contract, Contractor must immediately notify State.

**6.4** Contractor's Key Personnel. Contractor's Key Personnel will be:
(names, titles)

Key Personnel assigned to this project cannot be changed without the written approval of the State's Project Manager. Contractor will submit a change request in writing to the State's Project Manager along with a resume for each potential candidate. Potential new or additional personnel may be required to participate in an interview. Upon approval of new or additional personnel, the State's Authorized Representative may issue a change order to add or delete key personnel.

**7 Time**

The Contractor must comply with all the time requirements described in this Work Order. In the performance of this Work Order, time is of the essence.

**8 Employee Status**

Pursuant to the Governor's Executive Order 08-01, if this contract, including any extension options, is or could be in excess of $50,000, Contractor certifies that it and its subcontractors:

**8.2** Comply with the Immigration Reform and Control Act of 1986 (U.S.C. 1101 et. seq.) in relation to all employees performing work in the United States and do not knowingly employ persons in violation of United States immigrations laws; and

**8.3** By the date of the performance of services under this contract, Contractor and all its subcontractors have implemented or are in the process of implementing the E-Verify program for all newly hired employees in the United States who will perform work on behalf of the State of Minnesota.

Contractor must obtain certifications of compliance with this section from all subcontractors who will participate in the performance of this contract. Subcontractor certifications must be maintained by Contractor and made available to the state upon request. If Contractor or its subcontractors are not in compliance with 1 or 2 above or have not begun or implemented the E-Verify program for all newly hired employees performing work under the contract, the state reserves the right to determine what action it may take including but not limited to, canceling the contract and suspending or debarring the contractor from state purchasing.

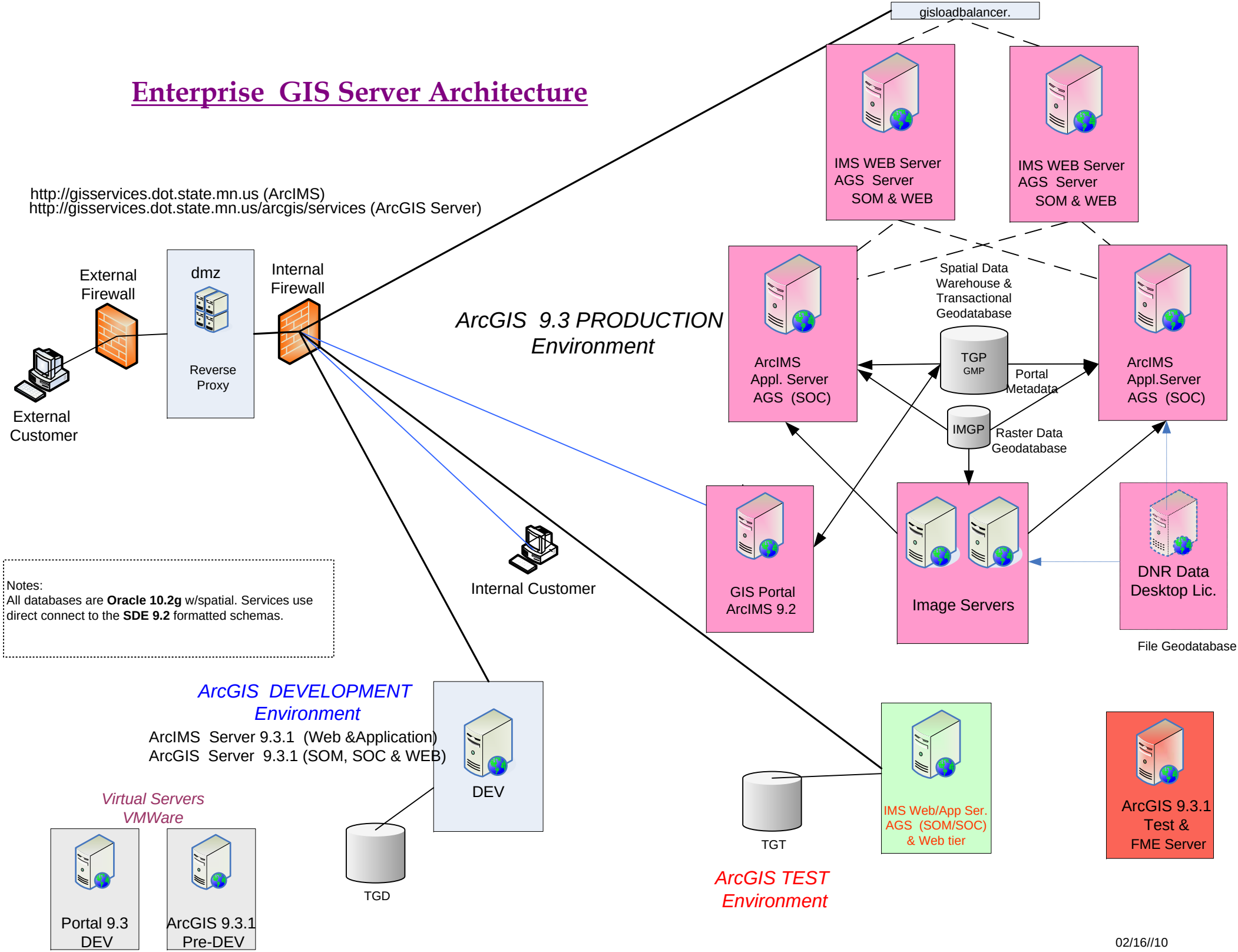
## 9 Additional Provisions

[Use this space to add information not covered elsewhere in this Work Order. If not needed, delete this section or state "None". The following should be used in any Work Order that includes web design:

The Contractor will comply with the "Minnesota Office of Enterprise Technology: Web Design Guidelines" available at the URL: http://www.state.mn.us/portal/mn/jsp/content.do?programid=536911233&id=-536891917&agency=OETweb.

*The balance of this page has been intentionally left blank.*

# Enterprise GIS Server Architecture



**Minnesota Department of Transportation**

395 John Ireland Blvd.
St. Paul, MN 55155

Information and Technology Services - Applications Management:

# Java Software Development Standards

Version 5.3
March 10, 2010

Authored by: Rick Meyer, Information and Technology Services, Systems Architect
e-mail: Rick.Meyer@dot.state.mn.us

# 1. Introduction

This document describes the software development environment to be used for Java / Java EE applications and services. It follows the Information and Technology Services strategic technology recommendations for Java applications.

# 2. Benefits

The following benefits are expected from this architecture:

- Not tied to any particular operating system or server product / platform.
- Conforms to an environment that is easy for Information and Technology Services to support.
- Is object-oriented, which promotes component-based, reusable, and extendable code. This makes it relatively easy to change, add functionality, and interface to other systems / services.
- Is service-oriented, which promotes the use of common services such as directory-based authentication, Crystal Enterprise Reports service, and Web services.
- Emphasizes W3C compliance for browsers and discourages the use of IE or Netscape proprietary functionality.

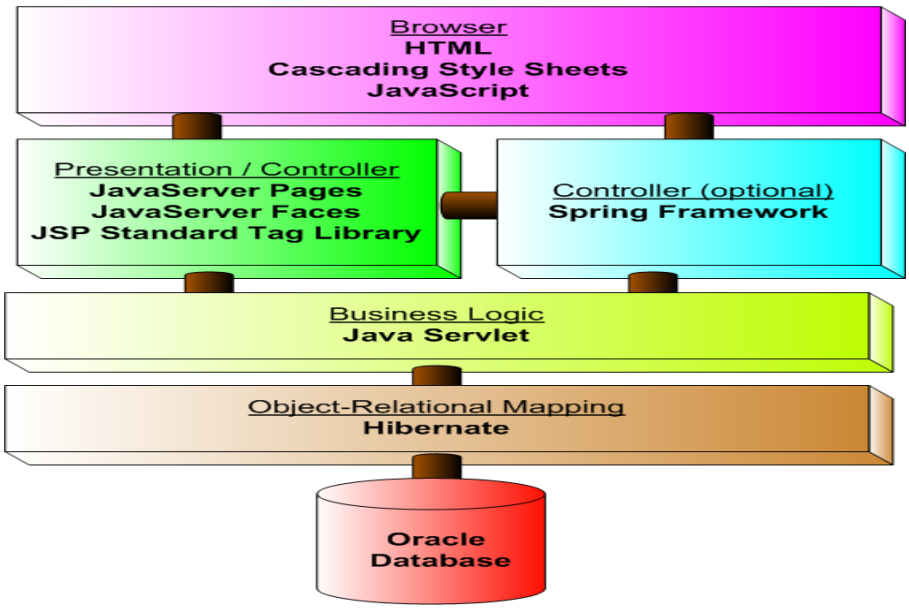
# 3. Development Environment

## 3.1 Application Technology

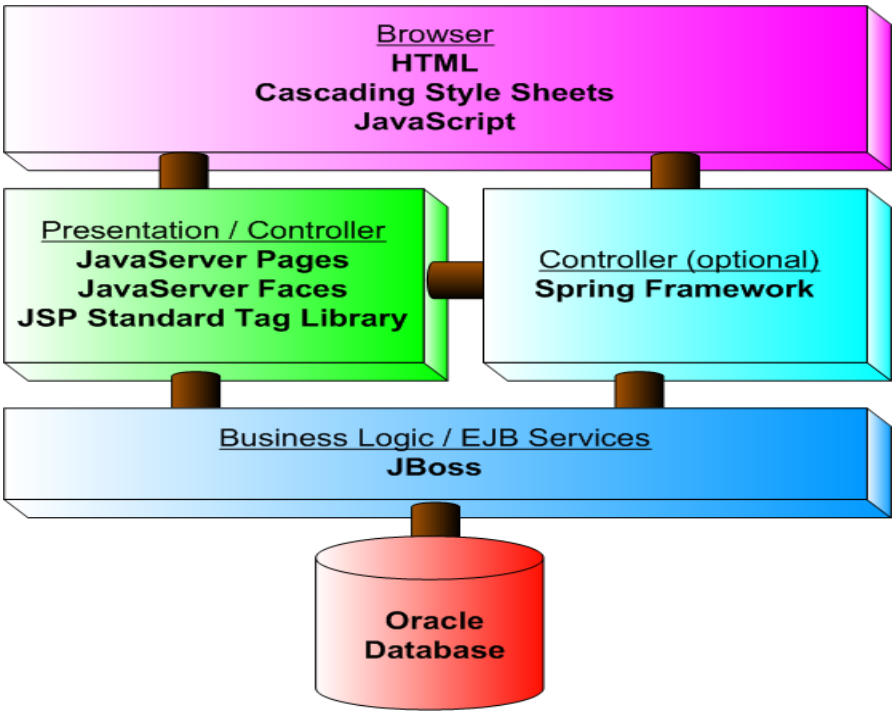
The application development / maintenance technology for Java applications is as follows:

| | |
|---|---|
| 1. Application architecture: | Java Enterprise Edition |
| 2. Application development language: | Java (SDK 1.5.x) |
| 3. Scripting Language: | JavaScript, PHP |
| 4. Presentation: | JSF, JSP, JSTL, HTML, CSS |
| 5. Reports solution: | Business Objects (Crystal) Enterprise XI |
| 6. Database: | Oracle 10G |
| 7. Object / relational mapping: | Hibernate 3 |
| 8. Controller framework: | JSF (Apache MyFaces), Spring |
| 9. Servlet application server: | Tomcat (ver 5.5) embedded in JBoss 4 |
| 10. EJB application server: | JBoss Application Server (ver 4.0.5) |
| 11. Code version control system / repository: | CVS |
| 12. Build / deploy / dependency tool: | Ant & Maven2 |
| 13. Integrated Development Environment (IDE): | Eclipse IDE with MyEclipse IDE plug-in |
| 14. Modeling / design tools: | Eclipse, Visio, ArgoUml |
| 15. Primary programming tool: | Eclipse IDE |
| 16. Primary testing tools: | Eclipse, JUnit, HTTP Unit |
| 17. CVS interface tools: | Eclipse, TortoiseCVS, ViewCVS |
| 18. Defect / Issue tracking tool: | JIRA |
| 19. Logging tool: | Log4J |

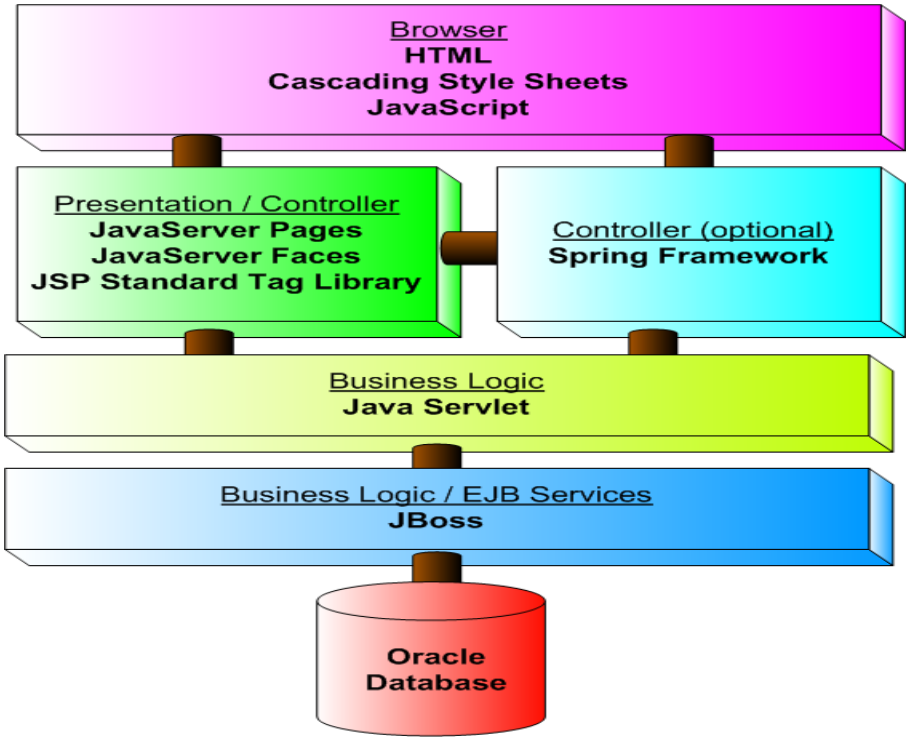
**Java Servlet Application Technology Model:**



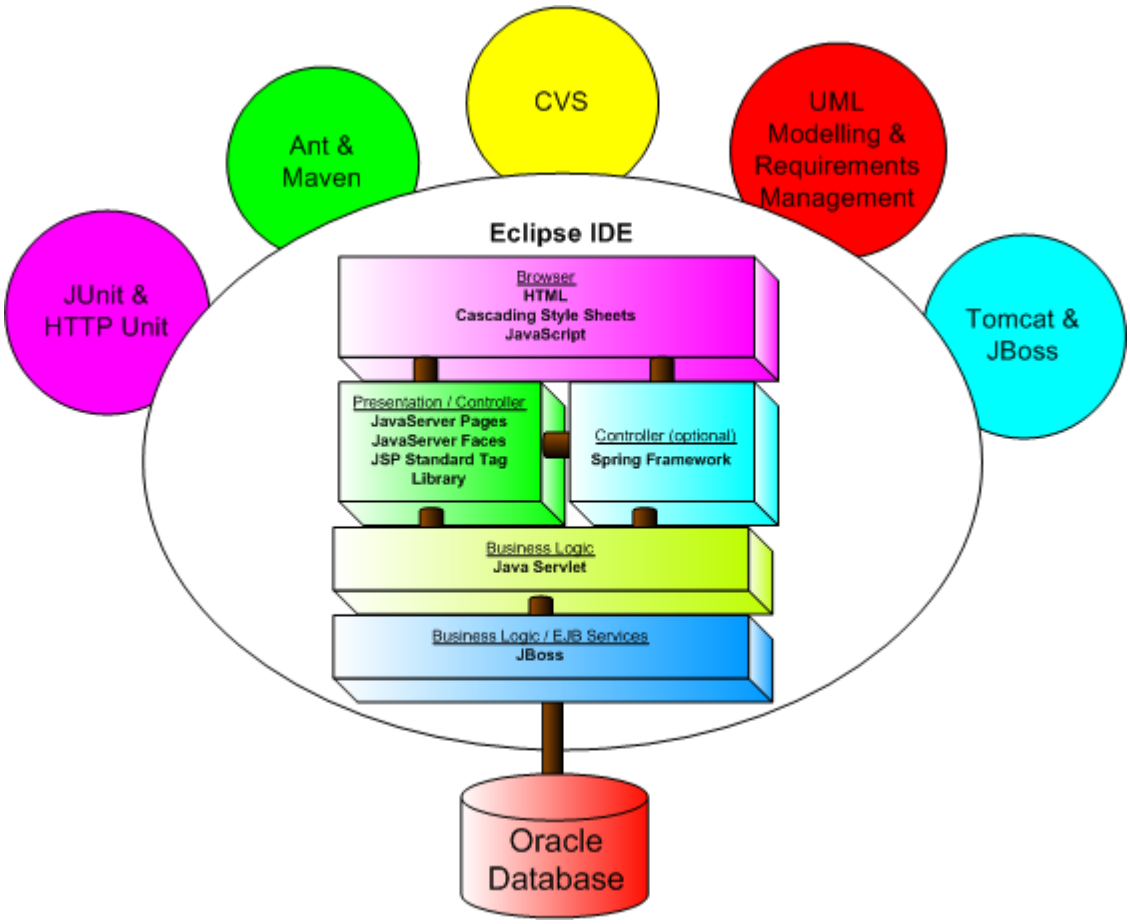
**Java EJB Application Technology Model:**




**Java EJB with Servlet Front-End Application Technology Model:**



**Development / Maintenance Environment Technology Model:**



## 3.2 Development Environment Specifications and Processes:

Mn/DOT's Information and Technology Services has specifications for a Java servlet that software developers will follow. These include:

1. **Coding Standards:**

   Developers will follow Sun Microsystems' coding standards. Located at http://java.sun.com/docs/codeconv/
   Developers will use PMD to check coding compliance to best practices and will deliver a clean PMD report at the end of each iteration. PMD is located here: http://pmd.sourceforge.net/
   There's a PMD plugin for Eclipse.

   PDM must be used by Java developers on a regular basis to identify poor coding practices.

   PMD scans Java source code and looks for potential problems like:

   - Possible bugs - empty try/catch/finally/switch statements
   - Dead code - unused local variables, parameters and private methods
   - Suboptimal code - wasteful String/StringBuffer usage
   - Overcomplicated expressions - unnecessary if statements, for loops that could be while loops
   - Duplicate code - copied/pasted code means copied/pasted bugs

   Guidelines for use:

   - Developers must run PMD at the end of each project iteration for their own benefit.
   - Developers must examine and fix all level 1 & level 2 violations, or explain why they should not be fixed.
   - Contractors must deliver a PMD report in Excel format to the project manager at the end of each project iteration. This report must be void of level 1 & level 2 violations, or an explanation should be written as to why the violation should be ignored.

## 2. Presentation (GUI) Standards:

Developers will use Apache MyFaces JavaServer Faces framework, the Apache Tomahawk library, PrimeFaces for AJAX-enhanced user interface components, and Facelets. Developers will follow Mn/DOT GUI standards for consistent "look and feel" with other Mn/DOT Web applications. See screen shots at the end of this document. Mn/DOT's Template Application includes pages that meet GUI standards. It's best to start with these pages. See "Template Application" for more detail.

Developers will follow these Screen Numbering Guidelines:

Developers should follow this example. Group the application's functionality into major categories and follow the example below. If there are more than 5 major categories, the developer can use smaller series' (2000 – 2499, 2500 – 2999, etc).

**General screens – 1000 series**
1000 - Welcome or index.html
1010 – Login
1020 – Common / Site-level search & search results

**Major Functional Category #1 – 2000 series**
2000 – Major Function General Info
2100 – View / Edit General Info
2200 – View / Edit Specific Info

**Major Functional Category #2 – 3000 series**
3000 – Major Function General Info
3100 – View / Edit General Info
3200 – View / Edit Specific Info

**Major Functional Category ...**

**Report Screens - 7000 series**
7000 - Report

**Error Screens - 8000 series**
8000 - Error

**Help & About Screens – 9000 series**
9000 – Help
9500 – About

3. **Versioning Standards:**

Developers will follow Apache Software Foundation's versioning guidelines. Located at http://apr.apache.org/versioning.html

Information and Technology Services simple description of versioning guidelines are as follows:

A <Major>.<Minor>.<Patch> numbering system is recommended to track software revisions when a new version is released.

A Major Release is a full product upgrade of the software containing significant new functionality. A Major Release is necessary if there are changes to the model or compatibility with previous versions cannot be maintained. When the Major Release version number is incremented, the Minor and Patch version numbers are reset to zero.

A Minor Release is a planned update to the existing software incorporating standard maintenance, improvements to existing features, enhancements and bug fixes. When a Minor Release number is incremented, the Major Release version number remains unchanged and the Patch Release version number is reset to zero.

A Patch Release is distributed when necessary to correct critical or significant problems that impact a customer's use of the system. When the Patch Release version number is incremented, the Major and Minor Release version numbers remain unchanged.

4. **Template Application:**

Developers will begin a project with Information and Technology Services 'template application', which consists of the technology listed in Section 3.1 and base configuration. This provides developers with a jump-start shell of an application. It helps ensure the developer starts within a Model/View/Controller framework and uses built-in patterns and best practices.
Contact Rick Meyer – Rick.Meyer@state.mn.us for a copy of the template app.

5. **Software build, dependency, and version management:**

Developers will use Ant, Maven2, and CVS for build, dependency, and version management. Developers will provide an Ant or Maven build script that builds the project without any IDE dependencies.

## 6. Source Code Maintenance:

Developers will provide and maintain application source files, configuration files, database schemas and scripts, developer documentation, and other related artifacts in Mn/DOT's CVS.

Follow these rules for naming CVS Modules:
One module (directory) per application (including all variants of that application).

The top level module name should be short (8 characters or less), all lower case letters and/or digits with no "-"(dash), and "_"(underscore), "."(period), or spaces. The module name may also be used in the JIRA project key, database schema name, and security role name.

The module name is short to facilitate naming of resources and artifacts where long user-friendly names would be problematic. For example, database table names are often prepended with the application name. Projects should determine the application's name (typically an acronym) early in the project so project artifacts and configuration files can use this name.

## 7. Documentation Tags (Javadoc):

Developers will use Javadoc tags for documentation. Please follow Sun's Javadoc reference

## 8. Object / Relational Mapping (Hibernate):

Developers will use Hibernate for Object / Relational Mapping

## 9. Code Portability:

Developers will follow practices that insure code portability. Peer reviews and documented justification is required for the use of proprietary application server extensions or libraries.

## 10. Deployment:

Developers will follow Information and Technology Services deployment process.

**Process for development server and test server application deployment:**
Use Anonymous FTP to place the deployment file (.war) at the following directory structure "/pub/incoming/apps/<appName>". The server will auto-deploy the .war within a few minutes of it being placed in its respective directory.

**Process for production server application deployment:**
1. Use Anonymous FTP to place a copy of the deployment files (.war) and any deployment instructions at the following directory structure "/pub/incoming/apps/<appName>".

2. Submit a Service Desk Ticket to the server administrator to deploy the production application.

**11. Auditing:**

Models and code will be reviewed for compliance to the specifications listed in Section 3.

**12. Controller Standards (JSF, Spring Framework)**

Developers will use Apache MyFaces JavaServer Faces framework. The use of Spring Framework is optional and should only be used if necessary to meet requirements that cannot reasonably be met with JSF and Hibernate. For example, transaction management.

**13. Java Application Server (Tomcat, JBoss)**

JBoss's embedded Tomcat is used to host Java servlets. JBoss is used to host EJB apps.

See section 3.3 of this document for details on the hosting environment.

**14. Application Security:**

Servlet security is implemented in the following manner:

Set up roles-based security.

1. Set up groups in the enterprise directory.
2. Define roles in the servlet container with 'web-sec-roles.xml'.

Example:

```
<security-role>
  <description>Application Admin Group</description>
  <role-name>Application Admin</role-name>
</security-role>

<security-role>
  <description>Application Staff Group</description>
  <role-name>Application Staff</role-name>
</security-role>
```

3. Map roles to resources (screens) with 'web-security.xml'.

Example:

```
<security-constraint>
   <web-resource-collection>
      <web-resource-name>This is the Welcome Screen</web-resource-name>
      <url-pattern>/index.jsp</url-pattern>
   </web-resource-collection>
   <auth-constraint>
      <role-name>Application Admin</role-name>
      <role-name>Application Staff</role-name>
```

```
    </auth-constraint>
</security-constraint>

<security-constraint>
    <web-resource-collection>
        <web-resource-name></web-resource-name>
        <url-pattern>/EditBudget.jsp</url-pattern>
        <url-pattern>/Import.jsp</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>Application Admin</role-name>
    </auth-constraint>
</security-constraint>

<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>application authentication</realm-name>
  <form-login-config>
    <form-login-page>/login.jsp </form-login-page>
    <form-error-page>/login_error.jsp </form-error-page>
  </form-login-config>
</login-config>
```

## 15. Scripting Standards

Developers will use JavaScript for client-side scripting, and PHP for server-side scripting.

The use of Javascript for client-side scripting should be based on the following cirteria.

1. The script is used for functionality, not styling (eye-candy) or fancy widgets.
2. The functionality cannot be provided by Apache's MyFaces, Tomahawk libraries, PrimeFaces, or the Java SDK libraries.
3. The script is not used as the only means of security or input validation.
4. The application must still provide base functionality if Javascript is disabled in the browser.
5. The script must follow coding best-practices:

    - Re-usable
    - Well commented
    - Well organized – in a separate .js file or consolidated in blocks, not sprinkled throughout and mixed with the other languages and constructs.

## 16. Naming Standards for Web applications

### User-friendly name

- The user-friendly name is the descriptive or popular name that identifies the application. The user-friendly name should be chosen by the project sponsor or application owner.
- Projects should determine the application's user-friendly name early in the project so it can be used in documentation.

### URL

- The URL is the application's Web address.
- Projects must determine the URL for the application. The URL is typically based on the user-friendly name, and must follow these examples:
    - Production URL = appname.dot.state.mn.us
    - Test URL = appnamet.dot.state.mn.us
    - Development URL = appnamed.dot.state.mn.us

### SSL for secure communications

- Projects must determine whether the application will use SSL (Secure Sockets) for secure communication early in the project because this affects the naming of server resources and configuration.

### Short-name

- The application short-name is used to name application and data resources and artifacts where long user-friendly names would be problematic. For example, database schema names are often prepended with the application short-name.
- Projects must determine the application's short-name (typically an acronym) early in the project so project artifacts and configuration files can use this name.

### 17. Issue / Bug Tracking (JIRA)

Developers will use JIRA for issue / bug tracking.

Mn/DOT's JIRA is located at http://jira.dot.state.mn.us. You must be connected to a Mn/DOT network for access.

### 18. Developer Team Collaboration (Wiki)

Information and Technology Services has a Wiki available for collaboration, documentation, and real-time information exchange. Be aware this Wiki is not a production service. Availability is not guaranteed.

Mn/DOT's Wiki is located at http://wiki.dot.state.mn.us. You must be connected to a Mn/DOT network for access.

### 19. Logging Standards

Developers must use Log4J. Developers must create application-specific log files. Do not write logs to the server's log file.

### 20. SSL for secure communications

SSL is provided by a proxy server for all external HTTP communications. There is no coding or configuration required by the developer. See "Model Java EE Web application in Mn/DOT's current infrastructure" diagram.

## 21. Reports Standards

Reports are created using Crystal Reports and hosted on the Business Objects Enterprise in a separate Web application. Report links must never be hard coded in the application. Instead, developer must use a Mn/DOT custom Tag Library included in the Template Application (archetype).

Application Dynamic Link to Reporting Environment

The Java application report's link points to the report's start page by dynamically retrieving the URL from the database. This feature has been added to the Mn/DOT tag library (mndot-taglib:reportsLink). During application compilation, the URL is queried from the database based on the application name and database information and imbedded in the application reports link.

Following these steps enable this feature:

1 - import the maven 2 dependency, or else get the appropriate jar file onto the classpath:

```
<dependency>
   <groupId>us.mn.state.dot</groupId>
   <artifactId>mndot-taglib</artifactId>
   <version>2.1.0</version>
</dependency>
```

2 - put this line of code at the top of your file:

```
<%@ taglib uri="http://dot.state.mn.us/customTags/DbStyle" prefix="mndot" %>
```

3 - put this line of code where you'd like the reports link to appear:

Internal:

```
<mndot:reportsLink dataSource="sampleDS"
appName="sample">Reports</mndot:reportsLink>
```

External:

```
<mndot:reportsLink dataSource="sampleDS" appName="sample" external="true"
>Reports</mndot:reportsLink>
```

If you are using the Mn/DOT standard applicationTemplate Version 1.3.0 or above, these steps are already included for you. However, you need to alter the datasource file name and application name specified at the step 3.

GIS applications have session reports. These reports allow the user to view the data in a session, but as soon as the session is closed the report goes away. These session reports are part of the application code and are stored in CVS. These reports are different than the reports that are used in most business applications.

The Crystal reports (Business Objects) that are developed for business applications and are binary.rpt format. These reports are stored in the Business Ojects repository.

Any reports done for GIS applications or Business applications that use the .rpt format are stored in the Business Objects repository.

## 3.3 Hosting Environment:

1. Application hosting services are typically provided by Information and Technology Services Infrastructure Section. Applications are hosted by JBoss / Tomcat on a Red Hat Enterprise Linux server. Application hosting typically resides in Mn/DOT's Network Operations Center (NOC). The customer provides a Web site to act as the customer-facing entry point for the application. This site could be an existing Web site located inside or outside Mn/DOT's NOC.
2. Database hosting services are typically provided by Information and Technology Services Infrastructure Section. Databases reside on a Dell storage area network (SAN) managed by an Oracle RDBMS running on Microsoft Windows server. Database hosting typically resides in Mn/DOT's Network Operations Center (NOC).
3. Reports hosting services are typically provided by Information and Technology Services Infrastructure Section. The reporting solution is typically Business Objects (Crystal) Enterprise. Reports hosting typically resides in Mn/DOT's Network Operations Center (NOC).

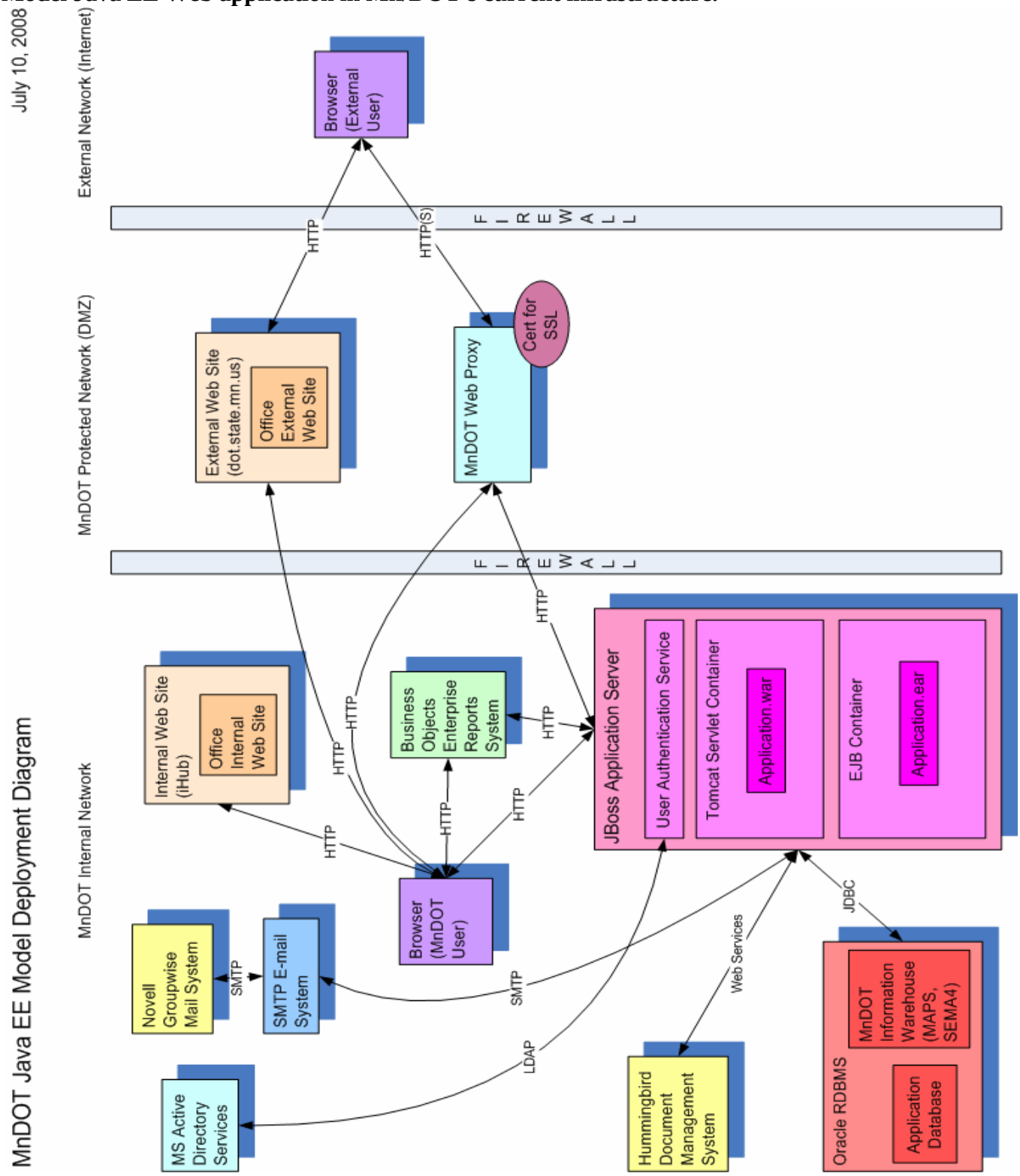
## 3.4 Development / Test Environment

Unit testing is done on development servers. Integration testing, system testing and user acceptance testing is done on test servers that reside in the same environment as the development servers.

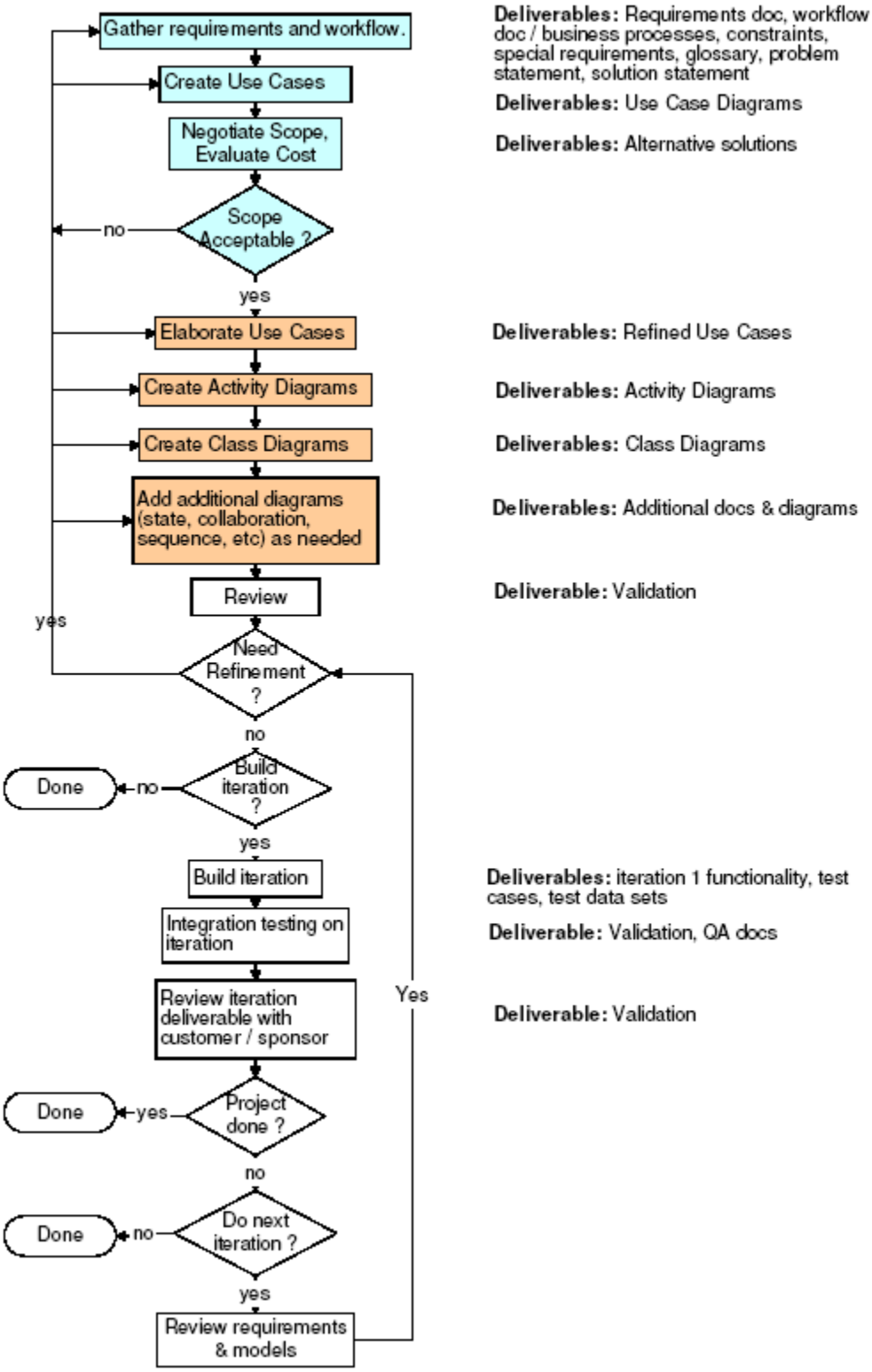
## 3.5 Production Environment

Production instances are deployed on production servers that typically reside in the same environment as the development and test servers.

**Model Java EE Web application in Mn/DOT's current infrastructure.**

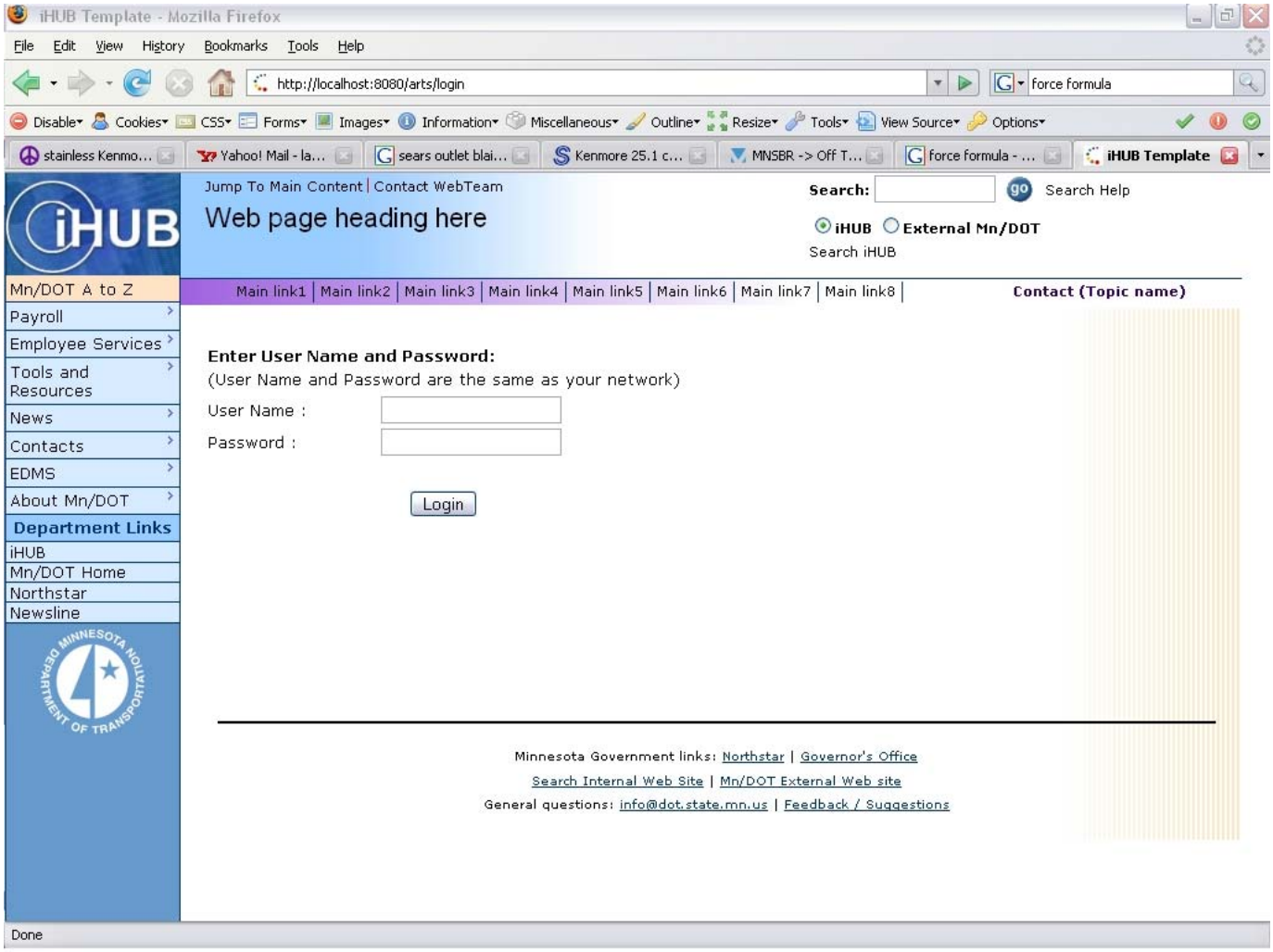


## Model iterative development cycle

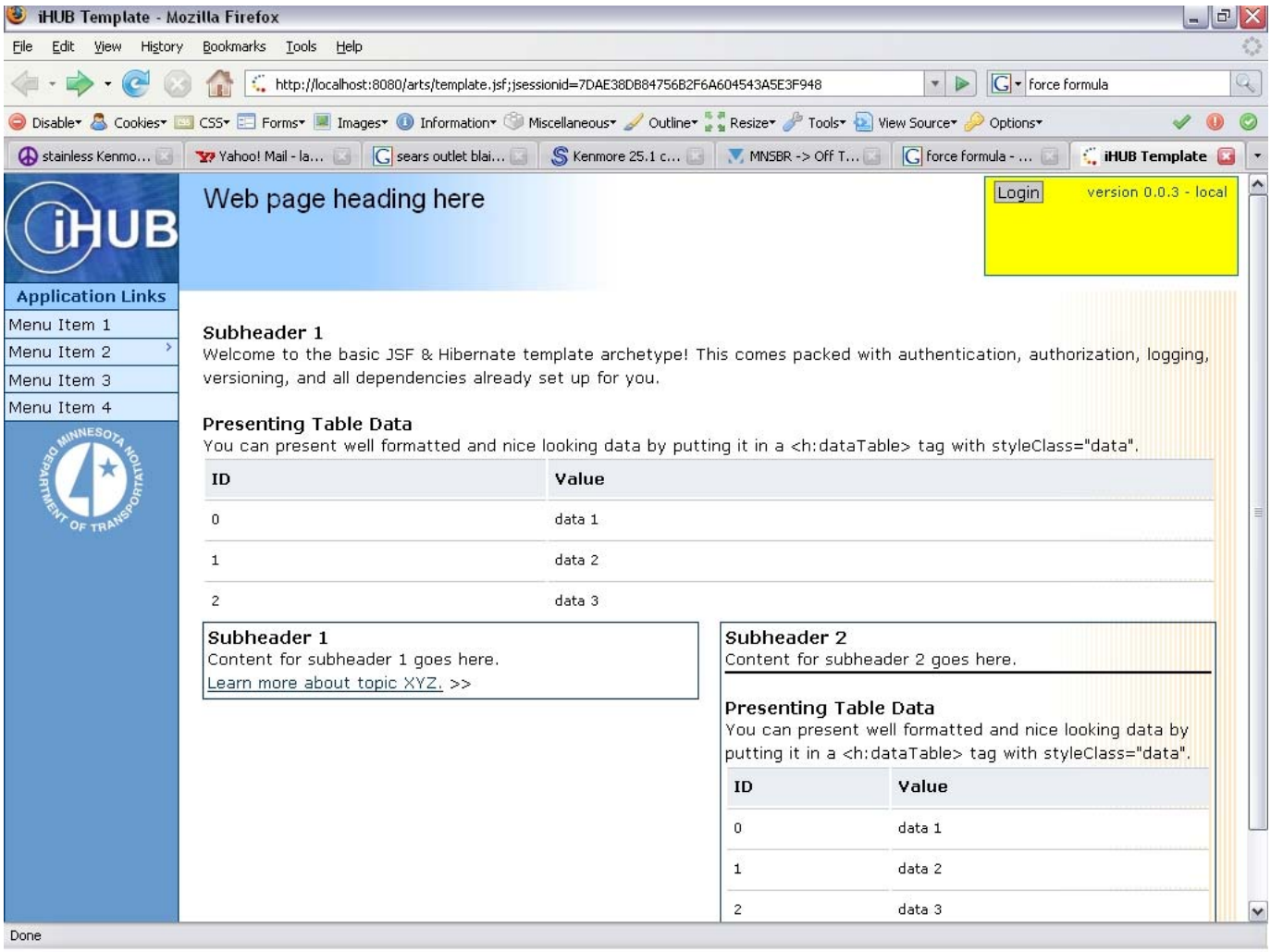


## Standard Web Application Template (screen shots)

Login page:

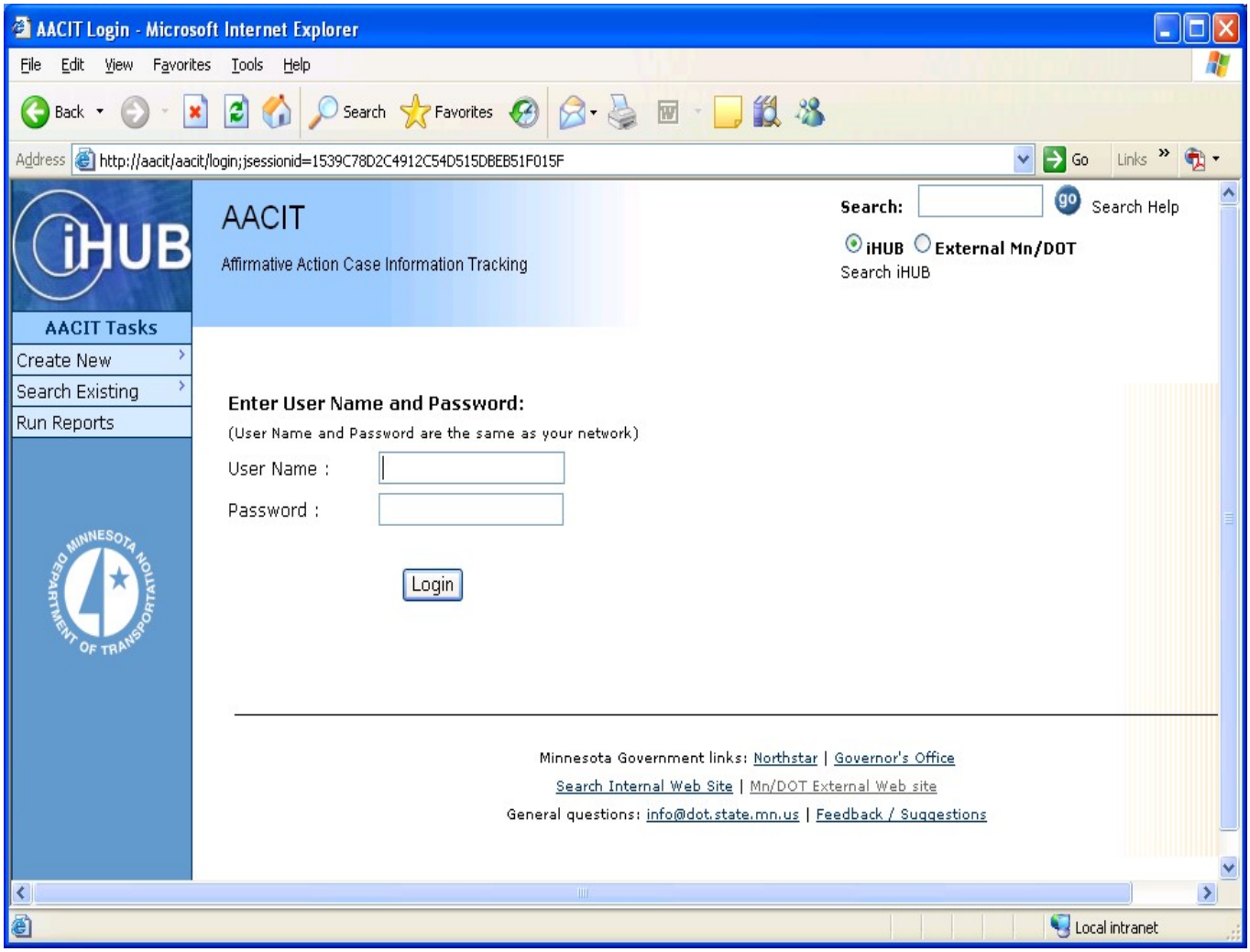


Typical application screen layout:




Note: The yellow background in the upper-right corner will turn white when the app moves into production.

Application screen showing Reports link



# *Mn/DOT*
# *Physical Data Modeling*
# *And Implementation Standards*

Mn/DOT Database Administrators Team
Version 2.0
Date: June 2010

# Version History

| Date | Version | Modified By \| Approved By | Section(s) | Comment |
|---|---|---|---|---|
| 2/11/2010 | 1.0 | | | Initial Draft taken from Mike's documentation, useful content found on the Web, and Team additions. |
| 2/22/2010 | 1.1 | | | Additions made from comments from the 2/11/2009 DBA Team Meeting. |
| 3/4/2010 | 1.2 | | | Additions made from comments from the 2/22/2010 DBA Team Meeting. |
| 3/12/2010 | 1.3 | | | Additions made from comments from the 3/11/2010 DBA Team Meeting. |
| 3/26/2010 | 1.4 | | | Additions made from comments from the 3/25/2010 DBA Team Meeting. |
| 6//2010 | 1.5 | | | Final Draft Review |
| 6/22/2010 | 2.0 | | | 1st Published Version |

# Table of Contents

PHYSICAL DATA MODELING STANDARD ..... 5
PURPOSE ..... 5
SCOPE ..... 5
STANDARD ..... 5
APPENDICES ..... 5
DEVIATION OF USE – Database Standards Exception Request ..... 6
PHYSICAL DATA MODELING and IMPLEMENTATION STANDARDS ..... 8
INTRODUCTION ..... 8
GENERAL NAMING STANDARDS ..... 9
TABLE STANDARDS ..... 10
Table Names ..... 10
Table Abbreviations ..... 10
Table Comments ..... 10
Support Tables ..... 12
COLUMNS ..... 13
Column Name ..... 13
Column Comments ..... 13
INDEXES ..... 14
Index Naming Syntax Guide ..... 14
Index Names ..... 14
INTEGRITY CONSTRAINTS ..... 16
Primary Key Constraint ..... 16
Unique Constraints ..... 17
Foreign Key Constraints ..... 18
Check Constraints ..... 18
Not Null Check Constraints ..... 19
SEQUENCES ..... 20
Sequence Names ..... 20
Column Comments - Sequences ..... 20
VIEWS ..... 21
View Names ..... 21
View Comment ..... 21
MATERIALIZED VIEWS ..... 22
Materialized View Names ..... 22
Materialized View Comment ..... 22
SYNONYMS ..... 23

Synonym Names ..... 23
Synonym Names – Security Models ..... 23
DATABASE LINKS ..... 24
PROCEDURE CODE ..... 25
Trigger ..... 28
Function ..... 28
Stored Procedure ..... 28
Package ..... 28
METADATA ..... 29
APPENDIX A: ORACLE SCHEMA OBJECT NAMES AND QUALIFIERS ..... 30
APPENDIX B: ORACLE RESERVED WORDS, KEYWORDS, AND NAMESPACES ..... 34
APPENDIX C: SQL SERVER RESERVED KEYWORDS ..... 39
APPENDIX D: MN/DOT MODIFIER WORDS AND ABBREVIATIONS ..... 47
APPENDIX E: MENU – MENU ITEMS LOOKUP TABLE DESIGN ..... 50
Menus – Menu Items Option 1 ..... 50
Menus – Menu Items Option 2 ..... 53
Menu – Menu Items Support Objects ..... 55
APPENDIX F: METADATA GUIDELINES ..... 56
Oracle Database Comment ..... 56
SQL Server Database Comment ..... 57
Sample Comments ..... 59
APPENDIX G: SECURITY MODELS ..... 62
Introduction ..... 62
Accounts ..... 62
Developer Account ..... 62
Application Account ..... 62
Abbreviation Overview ..... 62
Application Account Types ..... 63
Account Locking ..... 63
Instances ..... 64
Development Environment ..... 64
Test Environment ..... 64
Production Environment ..... 64
Roles ..... 65
Mn/DOT Security Model Configuration #1 ..... 65
Mn/DOT Security Model Configuration #2 ..... 65

# PHYSICAL DATA MODELING STANDARD

## PURPOSE

This document specifies Mn/DOT's Physical Data Modeling and Implementation Standards. The purpose of these standards is to ensure that Mn/DOT physical data models and database schema implementations have a consistent look and feel to simplify support and maintenance.

The Mn/DOT Database Administrator Team in the Office of Information and Technology Services (OI&TS) has been charged with performing data governance to insure that database schema objects are managed throughout the enterprise.

## SCOPE

This standard applies to all Oracle and SQL Server database development at Mn/DOT.

## STANDARD

The "Physical Data Modeling and Implementation Standards" Section of this document provide technical rules, guidelines, definitions, and references, including database naming standards, instructions for naming constraints, indexes, sequences, triggers and views. In addition, the physical data modeling specifications refer to the several additional supporting appendices.

When anything in the standard or supporting appendices is not explicitly stated or is unclear, always err on the side of the caution and ask the Mn/DOT Database Architect for clarification. Debate over what the standards "say" versus what they "meant" will not be entertained; the intent of the standards overrides any assumptions you may try to make. Any questions you offer are very welcome and will help to improve these standards over time.

## APPENDICES

Appendix A: Oracle Schema Object Names and Qualifiers
Appendix B: Oracle Reserved Words, Keywords and Namespaces
Appendix C: SQL Server Reserved Keywords
Appendix D: Mn/DOT Database Naming Standards Abbreviations and Acronyms List
Appendix E: MENU - MENU_ITEMS Lookup Table Design
Appendix F: Metadata Guidelines
Appendix G: Security Models

## DEVIATION OF USE – Database Standards Exception Request

Any deviation from these standards must be reviewed and approved by the Infrastructure Data Management (IDM) Supervisor. The Database Standards Exception Request Form shown on the next page can be requested from the Mn/DOT Database Administrators Team to request a deviation from the Database Standards. Granted deviations of these standards must be documented in the project plan/scope of work.

Commercial-Off-The-Shelf (COTS) or third party products will adhere to these standards. In a Request For Proposal (RFP) selection process, technical points will be deducted for not complying with these standards.

Common Database Objects (i.e. Tables, Views, etc) that are provided by Mn/DOT Database Administrators Team to support a project development may be except from these standards. These legacy database objects were constructed under different standards and requirements.

# Mn/DOT Database Standards Exception Request
## Office of Information and Technology Services (OI&TS)
## Infrastructure Data Management (IDM)

| | |
|---|---|
| Requestor Name | |
| Requestor Email | |
| Requestor Phone # | |
| Request Date | |
| Project Name | |
| Describe Exception to DB Standards | |
| **Notes (Office Use Only)** | |

Exception has been ____ Approved ____ Not Approved

IDM Supervisor signature: _____

Date: _____

# PHYSICAL DATA MODELING and IMPLEMENTATION STANDARDS

## INTRODUCTION

The physical data model is transformed into a physical implementation representing the physical tables and other database objects (views, triggers, procedures, etc.). The Mn/DOT Database Administrator Team in the Office of Information and Technology Services (OI&TS) may act as a consultant and in some cases build the physical data model, during the physical schema design.

**The Mn/DOT Database Administrators are responsible for the final review and approval of the physical data model before application code development begins for an iteration. "Deviation of Use" Exceptions will not be granted due to application code development performed before final approval of the Physical Data Model.**

The physical implementation has the following deliverables:

1. The physical data model, which is the "picture of the physical data elements" including tables, relationships, and views for the Physical Database Design.
2. Data Definition Language (DDL) for the Physical Database Objects for tables, indexes, constraints, views, triggers, procedures, packages, and functions needed to support the Physical Data Model.
3. The Metadata Dictionary, which consists of SQL comment statements for the tables, views, materialized views, and columns.

Standardizing naming conventions and abbreviations are defined in order to promote structure, consistency, readability, reusability, and maintainability. Applying these standards promotes data element definition reuse between databases in an effort to improve data sharing, data mapping, and eliminate data redundancy where possible.

## GENERAL NAMING STANDARDS

(All database objects must follow these basic rules.)

A database object are database components like a tables, columns, indexes, views, materialized views, sequences, constraints, user defined data types, triggers, stored packages/procedures, etc in a database.

The following naming conventions must be used for all database objects:

1. Only upper-case letters, [ A-Z ]; numbers, [ 0-9 ]; and underscore, [ _ ] are allowed in object names.

2. Use underscores [ _ ] as delimiters to provide a space holder between words making up object names.

3. The maximum length of an object name is limited to the length supported by the RDBMS version.

4. Reserved words can be used in an Object Name only if it is combined with other words. Refer to the Appendixes B and C.

5. Words making up object names must be spelled correctly.

6. Words used to construct object names must be descriptive, pronounceable, and related to business language.

7. Use the same object name to describe the same entity or attribute across database objects.

8. Standard prefixes and suffixes are used to identify common types of objects. Database objects must use Mn/DOT database naming standards for prefixes and suffixes.

9. Words used to define the database object name may be abbreviated, but must use the abbreviation or acronym found in Appendix D. Table Abbreviations can also be used in naming some database objects. See the Table Abbreviation Section for more detailed explanation. If the abbreviation or acronym is not found in the list, you must request an exception using the Database Standards Exception Request Form.

10. Name pattern (prefix and suffix), abbreviations and acronym usage must be consistently used throughout all database objects.

11. Abbreviations and acronyms must be spelled out in the metadata.

12. Naming conventions for objects in legacy applications must continue to follow the naming conventions that are in place. If it is unclear whether an application is considered legacy, always err on the side of the caution and ask the Mn/DOT Database Architect for clarification.

# TABLE STANDARDS

The table definition specifies what type of data the table can store and specifies referential integrity constraints used to restrict inserts, updates and deletions to the table's data. Tables in the physical data model must implement the business objects originally identified in the business model.

## Table Names

The following rules apply to table names:

1. All General Naming Standards Apply.

2. Table Names must be constructed from the name of the entity from the business or logical data model that the table is used to represent/support.

## Table Abbreviations

During database design and construction it maybe useful to define a Table Abbreviations for tables with longer names in the schema. Table Abbreviations can be used when naming Column Constraints, Indexes, Triggers, etc. Each Table Abbreviation in the schema must be unique.

Table Abbreviations can be created by using the first letter of each word that makes up the table name or an acceptable abbreviation of the words that make up the table name. Table Abbreviations can be used in object names only if it is combined with other words.

Example:

Table Name = CITY_COUNTY_DISTRICT
Table Abbreviation = CCD or CTY_CNTY_DIST

## Table Comments

Table comments must be created for each table and must be stored in the database dictionary. See Appendix F.

## Lookup (Reference) Tables (MENU – MENU ITEMS)

Mn/DOT developed a MENU – MENU ITEMS design in order to replace the small standard lookup tables with a generic set of tables to make menu maintenance easier. Typically most lookup tables are made up of a Code / ID with a Description along with supporting columns for Active/Inactive Indicator, Audit, and Sort Order used to support application menus.

The following rules apply for lookup (reference) tables:

1. The MENU – MENU ITEMS design must be used for lookup tables. If there is a need to deviate from this design, an exception must be requested and approved. See "Deviation of Use" on Page 6 for the Database Standards Exception Request Form.

2. When a table references this MENU - MENU ITEMS design, MENU_ITEMS.MENU_ITEMS_ID must be used when defining foreign key(s).

3. The column name used to reference the menu item must be named after the menu.

4. Every menu must have a recursive relationship at least to one item.

5. For directions on how to apply MENU – MENU ITEMS, see Appendix E.

6. The Mn/DOT DBA will provide the scripts to build the MENU – MENU ITEMS lookup table structure.

## Lookup (Reference) Tables (Approved by "Deviation of Use")

The following rules apply for lookup (reference) tables that were approved by "Deviation of Use":

1. The lookup table must at a minimum include the following columns:

| Column Type | Suffix | Data Type | Length |
|---|---|---|---|
| Primary Key | _ID | NUMBER | 22 |
| Name | _NAME | VARCHAR2 | 30 - 100 |
| Description | _DESC | VARCHAR2 | 100 |
| Active/Inactive Flag | _ACTIVE_FLAG | VARCHAR2 | 1 |
| Sort Order | _SEQUENCE | NUMBER | 22 |
| Audit for Record Creation | _CREATED_BY | VARCHAR2 | 50 |
| | _CREATED_DATE | DATE | |
| Audit for Record Modification | _MODIFIED_BY | VARCHAR2 | 50 |
| | _MODIFIED_DATE | DATE | |

2. Metadata table comments must clearly define the purpose and usage of these lookup (reference) tables.

3. The application must include a Maintenance Screen for each lookup table that is approved by "Deviation of Use".

## Lookup (Reference) Tables (Mn/DOT Provided)

Lookup (Reference) Tables provided by Mn/DOT are except from these standards, but must include metadata table comments that clearly define the purpose and usage of the table.

## Support Tables

Support tables provide application specific operational support in the physical database environment. The need for this type of table and the resulting naming standards associated with each support table will evolve as an application is developed.

The following rules apply for support tables:

1. Support tables must be named in a way making obvious that it is a support table.

2. Metadata table comments must clearly define the purpose and usage of these support tables.

3. Support Tables will include a Prefix of Suffix to help identify the table's usage.

| Table Type | Description | Possible Prefix | Possible Suffix |
|---|---|---|---|
| Archive | Tables used to hold archived data for historical retention that is seldom used for reporting, but must remain available. | ARCHIVE_<br>ARCH_ | _ARCHIVE<br>_ARCH |
| Conversion | Tables used to support data conversion from legacy systems to the new application tables | CONV_<br>CNV_ | _CONV<br>_CNV |
| History (Audit) | Tables used to support transaction audit history for a table | HISTORY_<br>HIST_ | _HISTORY<br>_HIST |
| Load | Tables used for staging areas for processing to support batch, upload, download, etc. where the data is temporarily stored before being loaded to permanent tables. | LOAD_ | _LOAD |
| Purge | Tables used to hold data that was Purged from the Source System, but is used frequently for reporting. | PURGE_ | _PURGE |
| Working | Tables used to support temporary processing for the application. | WORK_ | _WORK |

4. In most cases, it is more advisable to use separate specialized schemas to hold the data that is being archived, converted, loaded, or purged when the data is not an integral part of the application. Generally this data is only needed online for the period of time when the data is being processed for use in the application database. Please contact the Mn/DOT Database Administrators Team for clarification and assistance.

5.

# COLUMNS

Columns in the Physical Data Model are the transformed business elements originally identified in the business model. Columns must adhere to the following rules.

## Column Name

1. All General Naming Standards Apply.

2. Primary Key Columns must use a system generated unique number for a table. This would normally be generated by a sequence. For primary key columns, the column name will contain the [ _ID ] suffix.

3. Column ordering in tables must be ordered based most to least importance.
   Note: Mn/DOT has tools that can assist in restructuring the column order in a table.

   3.1. Within the mandatory grouping, order columns according to Primary Key, Unique Key, Foreign Key to Primary Business Tables, Business Defined columns including Lookup Foreign Key, Large Text Columns (Comments) and Audit Fields.

   3.2. Logically related columns must be grouped together whenever possible.

   For example columns used to support
   Names - First Name, Middle Name, Last Name
   Address - Street Address, City, State, Zip Code)

   3.3. Optional columns must be after mandatory ones.

   3.4. Audit Columns should be grouped together and appear at the end of the column list.

4. Column names that are used in more than one table must have the same business definition and must have the same characteristics in terms of data type, size, and business rules. This is also tru of columns used to link to other Application Systems.

5. Column names that are used to support a Foreign Key Relationship in a Detail Table must be the same name as the column(s) from the Master Table.

   Exception to this rule would be for Support/Lookup Tables where multiple foreign key constraints may exist to constrain several separate columns that have different business uses. For rules applying to Lookup Tables, including MENU and MENU_ITEMS, see Appendix E.

6. Modifier words or their approved abbreviations must always be associated with another word to be descriptive. A list of modifier words and abbreviations can be found in Appendix D.

## Column Comments

Column comments must be created for each column in a table or view and must be stored in the database dictionary. See Appendix F.

# INDEXES

An index is a data structure that takes the value of one or more columns of a table (the key) and returns all rows (or the requested columns in a row) with that value of the column quickly. Indexes are used to support database constraints and to improve query performance in an application database.

## Index Naming Syntax Guide

| Format Values | Format Description |
|---|---|
| {Table Name} | Table name. |
| {Table Abbreviation} | Abbreviated table name. |
| {Field Name} | Field or fields in the table that the index is created. |
| <id> | Represents the type of index.<br><br>General indexes: AK – Access Key<br><br>Use the following constraints: PK – Primary Key<br>UK – Unique Key<br>FK – Foreign Key |
| Nn | Optional Integer incremented for each index of the same type placed on a table Primary Key indexes do not have an integer suffix because a table can have only one primary key. The numbering is optional if there is only one index of a certain type. |

## Index Names

The following rules apply to Indexes:

1. Index names must be of the following format

   {Table Name} _<id>nn
   {Table Abbreviation}_<id>nn
   {Table Name}_{Field Name}_I
   {Table Abbreviation}_{Field Name}_I

   Examples: MNDOT_EMP_PK, GIS_HWY_UK01,

2. Specifying the tablespace in which the index will be created is required. Most applications have one or more tablespaces that are dedicated to storing indexes.

   Exception is the use of spatial index which requires that the index in the same tablespace as the table.

3. Indexes must be used to enforce Constraints and improve performance.

4. Not more than seven (7) indexes can be created per table.

5. Unique and Primary Key Indexes. Typically created when the constraint is created. Use the same name that was used for the Unique or Primary Constraint.

6. Foreign Key Indexes

6.1. Foreign Keys index support on detail tables is always recommended to support Master - Detail Queries.

6.2. Foreign Keys on columns supported by look up tables are optional and not recommended unless they will benefit the performance of the application.

## Example Index Names

| # | Table name | Primary Key Column name | Table Name or Table Abbreviation | Primary Key Constraint Name |
|---|---|---|---|---|
| 1 | MNDOT_EMPLOYEE | EMPL_WORK_LOCATION | MNDOT_EMP | MNDOT_EMP_AK1 |
| 2 | MNDOT_EMPLOYEE | USER_ID | MNDOT_EMP | MNDOT_EMP_UK1 |
| 3 | GIS_HWY | GISHWY_LOCATION | GH | GH_AK1 |

## Or

| # | Table name | Primary Key Column name | Table Name or Table Abbreviation | Primary Key Constraint Name |
|---|---|---|---|---|
| 1 | MNDOT_EMPLOYEE | EMPL_WORK_LOCATION | MNDOT_EMP | MNDOT_EMP_WORK_ LOCATION |
| 2 | MNDOT_EMPLOYEE | USER_ID | MNDOT_EMP | MNDOT_EMP_USER_ID |
| 3 | GIS_HWY | GISHWY_LOCATION | GH | GH_LOCATION |

# INTEGRITY CONSTRAINTS

Integrity constraints are used to ensure accuracy and consistency of data in a relational database. Data integrity is handled in a relational database through the concept of referential integrity. There are many types of integrity constraints that play a role in referential integrity.

**Constraint Naming Syntax Guide**

| Format Values | Format Description |
|---|---|
| {Table Name} | Table name. |
| {Table Abbreviation} | Abbreviated table name. |
| {Reference Table Name} | Table name of the table referenced in the Foreign Key reference. |
| {Reference Table Abbreviation} | Abbreviated table name of the table referenced in the Foreign Key reference. |
| nn | Optional Integer incremented for each constraint of the same type placed on a table. The numbering is optional if there is only one constraint of a certain type on a table. |

## Primary Key Constraint

The column, or set of columns, specified as part of the constraint definition, is known as the primary key. A primary key ensures each table row is uniquely identified and no duplicate rows are entered in that table. Primary key constraints are implemented through unique indexes, as are unique key constraints.

The following rules apply to primary keys:

1. All General Naming Standards Apply.

2. The column(s) in a Primary Key Constraint are required and will have a NOT NULL Constraint.

3. The Primary Key Constraint must be based on the required column or columns in a table that are used to uniquely identify a row in the table. See Column Naming Standards.

4. Primary key constraint names must be of the format:

   {Table Name}_PK or {Table Abbreviation}_PK.

5. Intersecting tables that are used to resolve many-to-many relationships between application tables must have a primary key based either of the following:

   5.1. Single primary key column which is system generated unique number for each row in the table.

   5.2. The combination of all the primary key columns from the intersecting tables.

6. Exceptions to Standards:

   6.1. Spatial Feature Classes do not have a primary key.

   6.2. Intersecting tables in a spatial implementation must use the combination of all the primary key columns from the intersecting tables.

### Example Primary Key Constraint and Column Names

| # | Table name | Primary Key Column name | Table Name or Table Abbreviation | Primary Key Constraint Name |
|---|---|---|---|---|
| 1 | MNDOT_EMPLOYEE | EMPL_NBR | MNDOT_EMP | MNDOT_EMP_PK |
| 2 | GIS_HWY | GISHWY_ID | GH | GH_PK |
| 3 | EMPLOYEE_WORK_ LOCATION | EMPL_NBR_ LOCATIONS | EMPL_WOR_LOC | EMPL_WOR_LOC_ PK |

## Unique Constraints

Use a unique key constraint to specify that no two rows of a table can have duplicate values in a specified column or set of columns. A table may have zero, one, or many unique key constraints.

1. All General Naming Standards Apply.

2. The column(s) in a Unique Constraint can be optional and allow Null Values.

3. Unique constraint names must be of the format:

   {Table Name}_UKnn or {Table Abbreviation}_UKnn

### Example Unique Key Constraint and Column Names

| # | Table name | Unique Key Column name | Table Name or Table Abbreviation | Unique Key Constraint Name |
|---|---|---|---|---|
| 1 | MNDOT_EMPLOYEE | EMPL_NBR | MNDOT_EMP | MNDOT_EMP_UK01 |
| 2 | GIS_HWY | GISHWY_ID | GH | GH_UK01 |

## Foreign Key Constraints

A foreign key constraint, also referred to as a referential integrity constraint, ensures that the values in the foreign key correspond to values of a primary key. Foreign key constraints are used so that records are not orphaned, provide ease of generating new models and provide ease of reverse engineering existing models.

1. All General Naming Standards Apply.

2. Foreign Key Constraints must be implemented and enforced.

   Exceptions to this standard are legacy databases where it is not possible to enforce foreign key constraints. If it is unclear whether an application is considered legacy, always err on the side of the cautious and ask a Mn/DOT Database Architect for clarification.

3. The Foreign Key Constraint name will be made up of the Table Name or Table Abbreviations of the table and the table that is being referenced. Use the following format:

   {Table Name or Table Abbreviation}_{Parent Table Name or Parent Table Abbreviation}_FKnn

## Check Constraints

Check constraints enforce logical expressions on columns, which must evaluate to true for every row in the table.

The following rules apply to Check Constraints:

1. All General Naming Standards Apply.

2. Check constraint names must be of the format:

   {Table Name or Table Abbreviation}_CKnn

3. Check constraint names are mandatory.

4. Check constraints must be used in coordination with the application developer to ensure that the database and application code don't contradict each other.

### Example Check Constraint and Column Names

| # | Table name | Column name | Table Name or Table Abbreviation | Check Constraint Name |
|---|---|---|---|---|
| 1 | MNDOT_EMPLOYEE | EMPL_NBR | MNDOT_EMP | MNDOT_EMP_CK01 |
| 2 | GIS_HWY | GISHWY_ID | GH | GH_CK02 |

## Not Null Check Constraints

NOT NULL constraints enforce columns that can't be allowed to be left null, with no values.

1. All General Naming Standards Apply.

2. When a column in a CREATE TABLE statement is specified to be NOT NULL, Oracle implements a CHECK constraint on the column to enforce the "not null" rule. These automatically generated constraints are named SYS%_Cxxx, where xxx is the system generated number.

   It is not necessary to rename "NOT NULL" constraints.

   If you choose to name Not Null constraints, use the following format:

   {Table Name}_NNnn or {Abbreviated Table Name}_NNnn

3. If a column is identified as a mandatory column, the definition NOT NULL must be added to the column definition in the Table DDL.

4. Not Null constraints must be used in coordination with the application developer to ensure that the database and application code do not contradict each other.

# SEQUENCES

Sequences are database objects used for generating unique integers for input into a table. They are useful for providing generated primary key values and for input of number type columns where the input must be unique and in some form of numerical sequence.

## Sequence Names

The following rules apply when creating Sequences:

1. All General Naming Standards Apply.

2. A sequence name must contain the [ _SEQ ] suffix.

3. Sequence Name may optionally include the Table Name/Abbreviated Table Name, but must include the column name that the sequence is supporting. In the case of columns that include the suffix "_ID", the "_ID" suffix is optional.

   For Example:

   Table CASE_TYPE (Table Abbreviated Name = CT) has a Primary Key Column CASE_TYPE_ID

| Format Options | Sequence Name |
|---|---|
| {Column Name}_SEQ | CASE_TYPE_ID_SEQ<br>CASE_TYPE_SEQ |
| {Table Short Name}_{Column_Name}_SEQ | CT_CASE_TYPE_ID_SEQ<br>CT_CASE_TYPE_SEQ |
| {Table Name}_{Column Name}_SEQ | CASE_TYPE_CASE_TYPE_ID_SEQ<br>CASE_TYPE_CASE_TYPE_SEQ |

4. Sequences must be used to generate primary key values for a table.

5. Every primary key must be supported by a separate, dedicated sequence.

6. The NOCYCLE parameter must be used.

7. The CACHE must be set to 0 (Zero) when the record creation transaction volume is low.

## Column Comments - Sequences

Sequences must be documented in the Column Comments for the Column using the sequence. See Appendix F.

# VIEWS

A view is a stored query that creates a dynamic, virtual (temporary) table that automatically refreshed when it is accessed by a user.

## View Names

The following rules apply when creating views:

1. All General Naming Standards Apply.

2. As with table names, view names must be descriptive, pronounceable, related to the business language, and indicate what the view contains. Views have two primary purposes:

   2.1. A view that pre-joins multiple tables is used to simplify a complex join for less sophisticated users or reporting tools.

   2.2. A security view is used to limit access to data inside a table. The Oracle accounts that need access to the view would be granted SELECT privileges on the view, but not on the base table.

3. View names must contain the [ _V } suffix.

### Example View Names

| # | View Type | Table name or Table Abbreviation #1 | Table name or Table Abbreviation #2 | View Name |
|---|---|---|---|---|
| 1 | Pre-join multiple tables | ABB_OFFICE | ABB_SEMA4_DEPT | ABB_OFFICE_TO_SEMA4_DEPT_V |
| 2 | Security View | MNDOT_EMPL_CONFID | | BLT_MNDOT_EMPL_V |

### Or sometimes there are multiple views that contain the same table joins. In this case, a view numbering increment must be used.

| # | Table name or Table Abbreviation #1 | Table name or Table Abbreviation #2 | View Name |
|---|---|---|---|
| 1 | ABB_OFFICE | ABB_SEMA4_DEPT | ABB_OFFICE_TO_SEMA4_DEPT_V01 |
| 2 | ABB_OFFICE | ABB_SEMA4_DEPT | ABB_OFFICE_TO_SEMA4_DEPT_V02 |

## View Comment

View comments must be created for each view and must be stored in the database. Metadata for the view name must list all tables/views involved in the view. See Appendix F.

# MATERIALIZED VIEWS

A Materialized view (also known as snapshot) is a stored query that creates a static table that contains the results of the query. The data in the materialized view can be updated as needed and the underlying table that supports the materialized view can be indexed in the same way as a table to provide better performance.

## Materialized View Names

When creating materialized views, apply the following rules:

1. All General Naming Standards Apply.

2. As with table names, materialized view names must be descriptive, pronounceable, related to the business language, and indicate what the materialized view contains. Materialized Views have two primary purposes:

   2.1. To create a refreshable snapshot from another database in the local database (replication).

   2.2. To create summary tables to improve query performance on very large tables through Oracle's query rewrite mechanism.

3. For a replication snapshot, name the materialized view {Application Name}_{Table Name}_MV, where {Application Name} indicates the external data source and {Table Name} indicates what the snapshot contains. For example, a snapshot containing buildings from the ITAMS schema might be named ITAMS_BUILDINGS_MV.

4. For a summary materialized view, name the materialized view {Application Name}_{Table Name}_MVnn, where nn is a sequential number and {Table Name} is the base table being summarized. This approach takes into account that summaries are often generated at multiple levels, and are not usually accessed directly; they are accessed transparently through query rewrite, so knowing the exact name of a specific summary table is not required. For example, a set of summary tables on table SIGNTRACK_FIELD_INVENTORY would be named SIGNTRACK_FIELD_INVENTORY_MV01, SIGNTRACK_FIELD_INVENTORY_MV02, etc.

5. As with tables, specifying a tablespace for the materialized view and any indexes created on the materialized view is required.

   Exception is a spatial index which needs to have the index in the same tablespace as the table.

## Materialized View Comment

Materialized view comments must be created for each materialized view and must be stored in the database. Metadata for the Materialized View name must list all tables/views involved in the Materialized View. See Appendix F.

# SYNONYMS

Synonyms are aliases for objects in the database, and they are used to make it easy for users to access database objects owned by other users, and for security purposes. Synonyms hide the underlying object's identity and can be either private or public.

## Synonym Names

When creating synonyms, apply the following rules:

1. All General Naming Standards Apply.

2. Public and Private synonyms must be used so that application code does not prefix objects with the schema owner name. In other words, for portability purposes of an application, application code must not use fully qualified object names.

3. Any request for a public synonym must be approved and created by a Mn/DOT Database Administrator.

   Exceptions to this rule are synonyms automatically generated by GIS tools, such as ArcSDE, SDO, etc.

4. You may create private synonyms to refer to the same table by different aliases in different schemas.

## Synonym Names – Security Models

Mn/DOT follows 2 basic security models for application deployments. Please refer to Appendix G for further details.

**Mn/DOT Security Model Configuration #1** is used primarily for Web Based Applications where the application generally uses a single sign user account to the database and user authentication occurs externally using a directory service or from databases tables in the application schema. This model will use private synonyms in the majority of cases.

**Mn/DOT Security Model Configuration #2** is used primarily for Client Server/Web Based Applications where the application has individual user accounts on the database that are used for authentication. This model will use public synonyms in the majority of cases.

There are cases in our environment where shared applications may reside on the same database and the database will have both Security Models present.

## DATABASE LINKS

A database link allows access to a different database through a remote database user account.

When creating database links, apply the following rules:

1. All General Naming Standards Apply.
2. Any request for a private database link must be approved and created by a Mn/DOT Database Administrator.
3. If there is a need for a public database link, an exception must be requested.
4. Database link names must contain the [ _LINK.WORLD } suffix.
5. Database Links must use separate user accounts on the target database that is secure and limited on access only to the objects needed by the application.

## PROCEDURE CODE

1. All General Naming Standards Apply.

2. All PLSQL Code must be formatted for readability.

3. General Layout

   3.1. The same layout and formatting style will be used throughout all scripts, procedures and packages for an Application Development Project.

   3.2. Vertical alignment will be used to make code more readable.

   3.3. Indenting will be used to assist in identifying the flow of control, blocks of code, and to make all scripts more readable.

   3.4. Spaces should be used for indentation and for all other white space.

   3.5. Tab characters should not be used for indentation.

   3.6. When indenting code a minimum of two (2) spaces will be used per each logical layer of code indentation.

   3.7. The elements within a line of PL/SQL and SQL commands should be separated by a minimum of a single space.

   3.8. A newline will be used to separate logical groupings of SQL Commands and PL/SQL Code.

4. Case Keywords and User Defined Objects

   4.1. All SQL and PL/SQL keywords shall be in Upper Case.

   4.2. The names of all built in functions, procedures and packages shalll be in Upper Case.

   4.3. All other identifiers (Such as table names, view names, column names, user defined package , procedures, functions, and variables names, etc.) shall be in lower case.

5. Variable/Parameter Lists and Logical Operators

   5.1. Variable and parameter names should be meaningful, descriptive, pronounceable, and be related to their use within the code.

5.2. Variable and parameter names should be distinguished from other parts of the code by either a prefix or suffix.

| Variable Type | Description | Possible Prefix | Possible Suffix |
|---|---|---|---|
| Local Variables | Local variables used in a section of code. | lv_<br>v_ | _lv<br>_v |
| Passed Parameters | Variable names used as parameters to call to a Function, Procedure, or Package. | pv_<br>p_ | _pv<br>_p |
| Global Variables | Variables assigned that are used globally by a number of Functions, Procedures, and Packages. | gv_<br>g_ | _gv<br>_g |

5.3. All variable/parameter declarations including optional assignments shall be stacked on their own line and be left vertically aligned.

5.4. All cursors/variable/parameter declarations should be grouped together by type in the declaration section of the PL/SQL Code.

5.5. Parameters list for procedure and function calls must be consistently formatted either on the same line or stacked with one parameter per line to provide the best readability of the code.

5.6. The placement of commas for any stacked list of parameter declarations, parameters, columns, list of values, etc. shall be placed consistently in the code.

5.7. All AND – OR conditions shall be consistently formatted and placed to make the logic easy to follow in the code.

6. Specific Statements

6.1. The keywords for all Select, Insert, Update, and Delete statements shall be on their own line and be vertically aligned.

6.2. The column list for all statements shall be stacked on their own line and be left vertically aligned.

6.3. All cursor declarations shall be on the same line with the Select portion after the Cursor declaration and aligned in the same way as other Select statements.

6.4. All IF - ELSIF – THEN –ELSE statements shall have the keywords on separate lines and left vertically aligned. All code within each IF - ELSIF – THEN –ELSE statement shall be on its own line and indented for ease of readability.

6.5. All LOOP statements shall be on its own line with the corresponding END statement left vertically aligned. All code within each LOOP statement shall be indented for ease of readability.

7. Comments

7.1. Comment Blocks should be aligned with the left margin unless the comment is for a specific block or line of code in which case it should vertically aligned with that code.

7.2. All PLSQL Code must include a Author Section that specifies:

- Program Name.
- Description/Purpose of the code.
- Person who created the code.
- Date when code was created or changed.
- Description or notes describing the change.

```
*****************************************************************************************-
--*  PROGRAM  NAME:
--*  DESCRIPTION  :
--*
--*  HISTORY
--*  Author   Date         Description
--*  ------   ----------   -----------
--*
--*****************************************************************************************
```

8. Must include comments to explain variable usage, processing sections, and error exception handling.

9. Use the following naming conventions Triggers, Functions, Procedures, and Packages :

**Procedure Naming Syntax Guide**

| Format Values | Format Description |
|---|---|
| {Table Name} | Table name. |
| {Table Abbreviation} | Abbreviated table name. |
| {Descriptive Name} | A descriptive Name that describes the purpose of the code. |
| {A/B} | Indicates a<br>A – Trigger fires After Action on Table<br>B – Trigger fires Before Action on Table |
| {IUD} | Indicates I – Insert; U – Update; D – Delete |

### Trigger

Trigger Names must be of the format:

TRG_{Table Name}_{A/B}{IUD}

{Table Name}_{A/B}{IUD}_TRG

TRG_{Table Abbreviation}_{A/B}{IUD}

{Table Abbreviation}_{A/B}{IUD}_TRG

### Function

Function names must be of the format:

FUNC_{Descriptive Name}

{Descriptive Name}_FUNC

## Stored Procedure

Stored Procedure names must be of the format:

PROC_{Descriptive Name}

{Descriptive Name}_PROC

## Package

Package names must be of the format

PKG_{Descriptive Name}

{Descriptive Name}_PKG

# METADATA

Metadata is used to document the business and technical usage of a Table, View, Materialized View, and associated Columns in an application database schema. This information is useful in the construction phase of an application to provide a common understanding of the database objects for all business users and developers involved in the construction phase. Metadata also provides valuable information to the support staff involved in the ongoing maintenance of the application database schema after the construction has been completed. See Appendix F.

When creating metadata, apply the following rules:

1. Metadata is required for all Tables, Views, Materialized Views, and associated Columns for an application database schema.

2. Metadata for Tables, Views, Materialized Views, and associated Columns will be stored in the database as a Database Comments.

3. Database Comments for all Tables, Views, Materialized Views, and associated Columns for an application database schema must be delivered as part of any deployment scripts.

4. Database Table, View, Materialized View, and Column comments must be formatted to include a "Business Name" followed by a "Description Section" .

   The "Business Name" of a table/view/materialized view must be spelled out with no abbreviations and put in Business Terms.

   The "Business Name" of a column from the table/view/materialized view is spelled out with no abbreviations. Business Names are universal and are intended to define columns in terms of the business across multiple systems even if the column names themselves are spelled or abbreviated differently.

   The "Description Section" consist of 1 - 3 sentences describing in common business or technical language the purpose and usage of the Table, View, Materialized View or associated Columns.

   Format of a Comment: [*Business Name*] – *Description Section*.

5. The Business Name must not contain any abbreviations. The only exception is the Word "ID".

6. The Business Names may contain Acronyms used by the business, but these Acronyms must be spelled out in the Description Section of the comment.

7. The Description Section must follow the Metadata Guidelines outlined in Appendix F This appendix outlines the format, requirements, and Standard Business Names already defined by Mn/DOT for the different types of Tables, Views, Materialized Views, and associated columns.

# APPENDIX A: ORACLE SCHEMA OBJECT NAMES AND QUALIFIERS

Oracle® Database SQL Language Reference
11*g* Release 1 (11.1)
Schema Object Names and Qualifiers

Some schema objects are made up of parts that you can or must name, such as the columns in a table or view, index and table partitions and subpartitions, integrity constraints on a table, and objects that are stored within a package, including procedures and stored functions. This section provides:

1. Rules for naming schema objects and schema object location qualifiers
2. Guidelines for naming schema objects and qualifiers

**Note:**
Oracle uses system-generated names beginning with "SYS_" for implicitly generated schema objects and subobjects, and names beginning with "ORA_" for some Oracle-supplied objects. Oracle discourages you from using these prefixes in the names you explicitly provide to your schema objects and subobjects to avoid possible conflict in name resolution.

**Schema Object Naming Rules**

Every database object has a name. In a SQL statement, you represent the name of an object with a **quoted identifier** or a **non-quoted identifier**.

- A quoted identifier begins and ends with double quotation marks ("). If you name a schema object using a quoted identifier, then you must use the double quotation marks whenever you refer to that object.
- A non-quoted identifier is not surrounded by any punctuation.

You can use either quoted or non-quoted identifiers to name any database object. However, database names, global database names, and database link names are always case insensitive and are stored as uppercase. If you specify such names as quoted identifiers, then the quotation marks are silently ignored. Refer to CREATE USER for additional rules for naming users and passwords.

The following list of rules applies to both quoted and non-quoted identifiers unless otherwise indicated:

1. Names must be from 1 to 30 bytes long with these exceptions:
    - Names of databases are limited to 8 bytes.
    - Names of database links can be as long as 128 bytes.

    If an identifier includes multiple parts separated by periods, then each attribute can be up to 30 bytes long. Each period separator, as well as any surrounding double quotation marks, counts as one byte. For example, suppose you identify a column like this:

    "*schema*"."*table*"."*column*"

    The schema name can be 30 bytes, the table name can by 30 bytes, and the column name can be 30 bytes. Each of the quotation marks and periods is a single-byte character, so the total length of the identifier in this example can be up to 98 bytes.

2. Non-quoted identifiers cannot be Oracle Database reserved words. Quoted identifiers can be reserved words, although this is not recommended.

   Depending on the Oracle product you plan to use to access a database object, names might be further restricted by other product-specific reserved words.

   **Note:**

   The reserved word ROWID is an exception to this rule. You cannot use the uppercase word ROWID, either quoted or non-quoted, as a column name. However, you can use the uppercase word as a quoted identifier that is not a column name, and you can use the word with one or more lowercase letters (for example, "Rowid" or "rowid") as any quoted identifier, including a column name.

   **See Also:**

   - Appendix B, "Oracle Database Reserved Words" for a listing of all Oracle Database reserved words
   - The manual for a specific product, such as *Oracle Database PL/SQL Language Reference*, for a list of the reserved words of that product

3. The Oracle SQL language contains other words that have special meanings. These words include datatypes, schema names, function names, the dummy system table DUAL, and keywords (the uppercase words in SQL statements, such as DIMENSION, SEGMENT, ALLOCATE, DISABLE, and so forth). These words are not reserved. However, Oracle uses them internally in specific ways. Therefore, if you use these words as names for objects and object parts, then your SQL statements may be more difficult to read and may lead to unpredictable results.

   In particular, do not use words beginning with SYS_ or ORA_ as schema object names, and do not use the names of SQL built-in functions for the names of schema objects or user-defined functions.

   **See Also:**

   "Datatypes", "About SQL Functions", and "Selecting from the DUAL Table"

4. You should use ASCII characters in database names, global database names, and database link names, because ASCII characters provide optimal compatibility across different platforms and operating systems.
5. Non-quoted identifiers must begin with an alphabetic character from your database character set. Quoted identifiers can begin with any character.
6. Non-quoted identifiers can contain only alphanumeric characters from your database character set and the underscore (_), dollar sign ($), and pound sign (#). Database links can also contain periods (.) and "at" signs (@). Oracle strongly discourages you from using $ and # in non-quoted identifiers.

Quoted identifiers can contain any characters and punctuations marks as well as spaces. However, neither quoted nor non-quoted identifiers can contain double quotation marks or the null character (\0).

7. Within a namespace, no two objects can have the same name.

   The following schema objects share one namespace:

   - Tables
   - Views
   - Sequences
   - Private synonyms
   - Stand-alone procedures
   - Stand-alone stored functions
   - Packages
   - Materialized views
   - User-defined types

   Each of the following schema objects has its own namespace:

   - Indexes
   - Constraints
   - Clusters
   - Database triggers
   - Private database links
   - Dimensions

   Because tables and views are in the same namespace, a table and a view in the same schema cannot have the same name. However, tables and indexes are in different namespaces. Therefore, a table and an index in the same schema can have the same name.

   Each schema in the database has its own namespaces for the objects it contains. This means, for example, that two tables in different schemas are in different namespaces and can have the same name.

   Each of the following non-schema objects also has its own namespace:

   - User roles
   - Public synonyms
   - Public database links
   - Tablespaces
   - Profiles
   - Parameter files (PFILEs) and server parameter files (SPFILEs)

   Because the objects in these namespaces are not contained in schemas, these namespaces span the entire database.

8. Non-quoted identifiers are not case sensitive. Oracle interprets them as uppercase. Quoted identifiers are case sensitive.

   By enclosing names in double quotation marks, you can give the following names to different objects in the same namespace:

employees
"employees"
"Employees"
"EMPLOYEES"

Note that Oracle interprets the following names the same, so they cannot be used for different objects in the same namespace:

employees
EMPLOYEES
"EMPLOYEES"

9. When Oracle stores or compares identifiers in uppercase, the uppercase form of each character in the identifiers is determined by applying the uppercasing rules of the database character set. Language-specific rules determined by the session setting NLS_SORT are not considered. This behavior corresponds to applying the SQL function UPPER to the identifier rather than the function NLS_UPPER.

   The database character set uppercasing rules can yield results that are incorrect when viewed as being in a certain natural language. For example, small letter sharp s ("ß"), used in German, does not have an uppercase form according to the database character set uppercasing rules. It is not modified when an identifier is converted into uppercase, while the expected uppercase form in German is the sequence of two characters capital letter S ("SS"). Similarly, the uppercase form of small letter i, according to the database character set uppercasing rules, is capital letter I. However, the expected uppercase form in Turkish and Azerbaijani is capital letter I with dot above.

   The database character set uppercasing rules ensure that identifiers are interpreted the same in any linguistic configuration of a session. If you want an identifier to look correctly in a certain natural language, then you can quote it to preserve the lowercase form or you can use the linguistically correct uppercase form whenever you use that identifier.

   Columns in the same table or view cannot have the same name. However, columns in different tables or views can have the same name.

10. Procedures or functions contained in the same package can have the same name, if their arguments are not of the same number and datatypes. Creating multiple procedures or functions with the same name in the same package with different arguments is called **overloading** the procedure or function.

# APPENDIX B: ORACLE RESERVED WORDS, KEYWORDS, AND NAMESPACES

http://oracle.su/appdev.111/b31231/appb.htm

This appendix contains the following sections:

Oracle Reserved Words
Oracle Keywords
PL/SQL Reserved Words
Oracle Reserved Namespaces

This appendix lists words that have a special meaning to Oracle. Each word plays a specific role in the context in which it appears. For example, in an INSERT statement, the reserved word INTO introduces the tables to which rows will be added. But, in a FETCH or SELECT statement, the reserved word INTO introduces the output host variables to which column values will be assigned.

**Oracle Reserved Words**

The following words are reserved by Oracle. That is, they have a special meaning to Oracle and so cannot be redefined. For this reason, you cannot use them to name database objects such as columns, tables, or indexes.

| **Oracle Reserved Words** | | | |
|---|---|---|---|
| ACCESS | ELSE | MODIFY | START |
| ADD | EXCLUSIVE | NOAUDIT | SELECT |
| ALL | EXISTS | NOCOMPRESS | SESSION |
| ALTER | FILE | NOT | SET |
| AND | FLOAT | NOTFOUND | SHARE |
| ANY | FOR | NOWAIT | SIZE |
| ARRAYLEN | FROM | NULL | SMALLINT |
| AS | GRANT | NUMBER | SQLBUF |
| ASC | GROUP | OF | SUCCESSFUL |
| AUDIT | HAVING | OFFLINE | SYNONYM |
| BETWEEN | IDENTIFIED | ON | SYSDATE |
| BY | IMMEDIATE | ONLINE | TABLE |
| CHAR | IN | OPTION | THEN |
| CHECK | INCREMENT | OR | TO |
| CLUSTER | INDEX | ORDER | TRIGGER |
| COLUMN | INITIAL | PCTFREE | UID |

| Oracle Reserved Words | | | |
|---|---|---|---|
| COMMENT | INSERT | PRIOR | UNION |
| COMPRESS | INTEGER | PRIVILEGES | UNIQUE |
| CONNECT | INTERSECT | PUBLIC | UPDATE |
| CREATE | INTO | RAW | USER |
| CURRENT | IS | RENAME | VALIDATE |
| DATE | LEVEL | RESOURCE | VALUES |
| DECIMAL | LIKE | REVOKE | VARCHAR |
| DEFAULT | LOCK | ROW | VARCHAR2 |
| DELETE | LONG | ROWID | VIEW |
| DESC | MAXEXTENTS | ROWLABEL | WHENEVER |
| DISTINCT | MINUS | ROWNUM | WHERE |
| DROP | MODE | ROWS | WITH |

**Oracle Keywords**

The following words also have a special meaning to Oracle but are not reserved words and so can be redefined. However, some might eventually become reserved words.

| Oracle Keywords | | | |
|---|---|---|---|
| ADMIN | CURSOR | FOUND | MOUNT |
| AFTER | CYCLE | FUNCTION | NEXT |
| ALLOCATE | DATABASE | GO | NEW |
| ANALYZE | DATAFILE | GOTO | NOARCHIVELOG |
| ARCHIVE | DBA | GROUPS | NOCACHE |
| ARCHIVELOG | DEC | INCLUDING | NOCYCLE |
| AUTHORIZATION | DECLARE | INDICATOR | NOMAXVALUE |
| AVG | DISABLE | INITRANS | NOMINVALUE |
| BACKUP | DISMOUNT | INSTANCE | NONE |
| BEGIN | DOUBLE | INT | NOORDER |
| BECOME | DUMP | KEY | NORESETLOGS |
| BEFORE | EACH | LANGUAGE | NORMAL |
| BLOCK | ENABLE | LAYER | NOSORT |
| BODY | END | LINK | NUMERIC |
| CACHE | ESCAPE | LISTS | OFF |
| CANCEL | EVENTS | LOGFILE | OLD |
| CASCADE | EXCEPT | MANAGE | ONLY |
| CHANGE | EXCEPTIONS | MANUAL | OPEN |

| Oracle Keywords | | | |
|---|---|---|---|
| CHARACTER | EXEC | MAX | OPTIMAL |
| CHECKPOINT | EXPLAIN | MAXDATAFILES | OWN |
| CLOSE | EXECUTE | MAXINSTANCES | PACKAGE |
| COBOL | EXTENT | MAXLOGFILES | PARALLEL |
| COMMIT | EXTERNALLY | MAXLOGHISTORY | PCTINCREASE |
| COMPILE | FETCH | MAXLOGMEMBERS | PCTUSED |
| CONSTRAINT | FLUSH | MAXTRANS | PLAN |
| CONSTRAINTS | FREELIST | MAXVALUE | PLI |
| CONTENTS | FREELISTS | MIN | PRECISION |
| CONTINUE | FORCE | MINEXTENTS | PRIMARY |
| CONTROLFILE | FOREIGN | MINVALUE | PRIVATE |
| COUNT | FORTRAN | MODULE | PROCEDURE |
| PROFILE | SAVEPOINT | SQLSTATE | TRACING |
| QUOTA | SCHEMA | STATEMENT_ID | TRANSACTION |
| READ | SCN | STATISTICS | TRIGGERS |
| REAL | SECTION | STOP | TRUNCATE |
| RECOVER | SEGMENT | STORAGE | UNDER |
| REFERENCES | SEQUENCE | SUM | UNLIMITED |
| REFERENCING | SHARED | SWITCH | UNTIL |
| RESETLOGS | SNAPSHOT | SYSTEM | USE |
| RESTRICTED | SOME | TABLES | USING |
| REUSE | SORT | TABLESPACE | WHEN |
| ROLE | SQL | TEMPORARY | WRITE |
| ROLES | SQLCODE | THREAD | WORK |
| ROLLBACK | SQLERROR | TIME | |

**PL/SQL Reserved Words**

The following PL/SQL keywords may require special treatment when used in embedded SQL statements.

| PL/SQL Reserved Words | | | |
|---|---|---|---|
| ABORT | BETWEEN | CRASH | DIGITS |
| ACCEPT | BINARY_INTEGER | CREATE | DISPOSE |
| ACCESS | BODY | CURRENT | DISTINCT |
| ADD | BOOLEAN | CURRVAL | DO |
| ALL | BY | CURSOR | DROP |

| PL/SQL Reserved Words | | | |
|---|---|---|---|
| ALTER | CASE | DATABASE | ELSE |
| AND | CHAR | DATA_BASE | ELSIF |
| ANY | CHAR_BASE | DATE | END |
| ARRAY | CHECK | DBA | ENTRY |
| ARRAYLEN | CLOSE | DEBUGOFF | EXCEPTION |
| AS | CLUSTER | DEBUGON | EXCEPTION_INIT |
| ASC | CLUSTERS | DECLARE | EXISTS |
| ASSERT | COLAUTH | DECIMAL | EXIT |
| ASSIGN | COLUMNS | DEFAULT | FALSE |
| AT | COMMIT | DEFINITION | FETCH |
| AUTHORIZATION | COMPRESS | DELAY | FLOAT |
| AVG | CONNECT | DELETE | FOR |
| BASE_TABLE | CONSTANT | DELTA | FORM |
| BEGIN | COUNT | DESC | FROM |
| FUNCTION | NEW | RELEASE | SUM |
| GENERIC | NEXTVAL | REMR | TABAUTH |
| GOTO | NOCOMPRESS | RENAME | TABLE |
| GRANT | NOT | RESOURCE | TABLES |
| GROUP | NULL | RETURN | TASK |
| HAVING | NUMBER | REVERSE | TERMINATE |
| IDENTIFIED | NUMBER_BASE | REVOKE | THEN |
| IF | OF | ROLLBACK | TO |
| IN | ON | ROWID | TRUE |
| INDEX | OPEN | ROWLABEL | TYPE |
| INDEXES | OPTION | ROWNUM | UNION |
| INDICATOR | OR | ROWTYPE | UNIQUE |
| INSERT | ORDER | RUN | UPDATE |
| INTEGER | OTHERS | SAVEPOINT | USE |
| INTERSECT | OUT | SCHEMA | VALUES |
| INTO | PACKAGE | SELECT | VARCHAR |
| IS | PARTITION | SEPARATE | VARCHAR2 |
| LEVEL | PCTFREE | SET | VARIANCE |
| LIKE | POSITIVE | SIZE | VIEW |
| LIMITED | PRAGMA | SMALLINT | VIEWS |
| LOOP | PRIOR | SPACE | WHEN |

| PL/SQL Reserved Words | | | |
|---|---|---|---|
| MAX | PRIVATE | SQL | WHERE |
| MIN | PROCEDURE | SQLCODE | WHILE |
| MINUS | PUBLIC | SQLERRM | WITH |
| MLSLABEL | RAISE | START | WORK |
| MOD | RANGE | STATEMENT | XOR |
| MODE | REAL | STDDEV | |
| NATURAL | RECORD | SUBTYPE | |

**Oracle Reserved Namespaces**

The table below contains a list of namespaces that are reserved by Oracle. The initial characters of function names in Oracle libraries are restricted to the character strings in this list. Because of potential name conflicts, use function names that do not begin with these characters.

For example, the SQL*Net Transparent Network Service functions all begin with the characters "NS," so you need to avoid naming functions that begin with "NS."

Oracle Reserved Namespaces

| Namespace | Library |
|---|---|
| O | OCI functions |
| S | function names from SQLLIB and system-dependent libraries |
| XA | external functions for XA applications only |
| GEN KP L NA NC ND NL NM NR NS NT NZ TTC UPI | Internal functions |

# APPENDIX C: SQL SERVER RESERVED KEYWORDS

SQL Server 2008 Books Online (November 2009)
**Reserved Keywords (Transact-SQL)**

**http://msdn.microsoft.com/en-us/library/ms189822.aspx**

Microsoft SQL Server uses reserved keywords for defining, manipulating, and accessing databases. Reserved keywords are part of the grammar of the Transact-SQL language that is used by SQL Server to parse and understand Transact-SQL statements and batches. Although it is syntactically possible to use SQL Server reserved keywords as identifiers and object names in Transact-SQL scripts, you can do this only by using delimited identifiers.

**The following table lists SQL Server reserved keywords.**

| | | |
|---|---|---|
| ADD | EXISTS | PRECISION |
| ALL | EXIT | PRIMARY |
| ALTER | EXTERNAL | PRINT |
| AND | FETCH | PROC |
| ANY | FILE | PROCEDURE |
| AS | FILLFACTOR | PUBLIC |
| ASC | FOR | RAISERROR |
| AUTHORIZATION | FOREIGN | READ |
| BACKUP | FREETEXT | READTEXT |
| BEGIN | FREETEXTTABLE | RECONFIGURE |
| BETWEEN | FROM | REFERENCES |
| BREAK | FULL | REPLICATION |
| BROWSE | FUNCTION | RESTORE |
| BULK | GOTO | RESTRICT |
| BY | GRANT | RETURN |
| CASCADE | GROUP | REVERT |
| CASE | HAVING | REVOKE |
| CHECK | HOLDLOCK | RIGHT |
| CHECKPOINT | IDENTITY | ROLLBACK |
| CLOSE | IDENTITY_INSERT | ROWCOUNT |
| CLUSTERED | IDENTITYCOL | ROWGUIDCOL |
| COALESCE | IF | RULE |
| COLLATE | IN | SAVE |
| COLUMN | INDEX | SCHEMA |
| COMMIT | INNER | SECURITYAUDIT |

| | | |
|---|---|---|
| COMPUTE | INSERT | SELECT |
| CONSTRAINT | INTERSECT | SESSION_USER |
| CONTAINS | INTO | SET |
| CONTAINSTABLE | IS | SETUSER |
| CONTINUE | JOIN | SHUTDOWN |
| CONVERT | KEY | SOME |
| CREATE | KILL | STATISTICS |
| CROSS | LEFT | SYSTEM_USER |
| CURRENT | LIKE | TABLE |
| CURRENT_DATE | LINENO | TABLESAMPLE |
| CURRENT_TIME | LOAD | TEXTSIZE |
| CURRENT_TIMESTAMP | MERGE | THEN |
| CURRENT_USER | NATIONAL | TO |
| CURSOR | NOCHECK | TOP |
| DATABASE | NONCLUSTERED | TRAN |
| DBCC | NOT | TRANSACTION |
| DEALLOCATE | NULL | TRIGGER |
| DECLARE | NULLIF | TRUNCATE |
| DEFAULT | OF | TSEQUAL |
| DELETE | OFF | UNION |
| DENY | OFFSETS | UNIQUE |
| DESC | ON | UNPIVOT |
| DISK | OPEN | UPDATE |
| DISTINCT | OPENDATASOURCE | UPDATETEXT |
| DISTRIBUTED | OPENQUERY | USE |
| DOUBLE | OPENROWSET | USER |
| DROP | OPENXML | VALUES |
| DUMP | OPTION | VARYING |
| ELSE | OR | VIEW |
| END | ORDER | WAITFOR |
| ERRLVL | OUTER | WHEN |
| ESCAPE | OVER | WHERE |
| EXCEPT | PERCENT | WHILE |
| EXEC | PIVOT | WITH |
| EXECUTE | PLAN | WRITETEXT |

Additionally, the ISO standard defines a list of reserved keywords. Avoid using ISO reserved keywords for object names and identifiers. The ODBC reserved keyword list, shown in the following table, is the same as the ISO reserved keyword list.

**Note:**

The ISO standards reserved keywords list sometimes can be more restrictive than SQL Server and at other times less restrictive. For example, the ISO reserved keywords list contains **INT**. SQL Server does not have to distinguish this as a reserved keyword.

Transact-SQL reserved keywords can be used as identifiers or names of databases or database objects, such as tables, columns, views, and so on. Use either quoted identifiers or delimited identifiers. Using reserved keywords as the names of variables and stored procedure parameters is not restricted. For more information, see Using Identifiers As Object Names.

ODBC Reserved Keywords

The following words are reserved for use in ODBC function calls. These words do not constrain the minimum SQL grammar; however, to ensure compatibility with drivers that support the core SQL grammar, applications should avoid using these keywords.

**This is the current list of ODBC reserved keywords.**

| | | |
|---|---|---|
| ABSOLUTE | EXEC | OVERLAPS |
| ACTION | EXECUTE | PAD |
| ADA | EXISTS | PARTIAL |
| ADD | EXTERNAL | PASCAL |
| ALL | EXTRACT | POSITION |
| ALLOCATE | FALSE | PRECISION |
| ALTER | FETCH | PREPARE |
| AND | FIRST | PRESERVE |
| ANY | FLOAT | PRIMARY |
| ARE | FOR | PRIOR |
| AS | FOREIGN | PRIVILEGES |
| ASC | FORTRAN | PROCEDURE |
| ASSERTION | FOUND | PUBLIC |
| AT | FROM | READ |
| AUTHORIZATION | FULL | REAL |
| AVG | GET | REFERENCES |
| BEGIN | GLOBAL | RELATIVE |
| BETWEEN | GO | RESTRICT |
| BIT | GOTO | REVOKE |
| BIT_LENGTH | GRANT | RIGHT |
| BOTH | GROUP | ROLLBACK |
| BY | HAVING | ROWS |

| | | |
|---|---|---|
| CASCADE | HOUR | SCHEMA |
| CASCADED | IDENTITY | SCROLL |
| CASE | IMMEDIATE | SECOND |
| CAST | IN | SECTION |
| CATALOG | INCLUDE | SELECT |
| CHAR | INDEX | SESSION |
| CHAR_LENGTH | INDICATOR | SESSION_USER |
| CHARACTER | INITIALLY | SET |
| CHARACTER_LENGTH | INNER | SIZE |
| CHECK | INPUT | SMALLINT |
| CLOSE | INSENSITIVE | SOME |
| COALESCE | INSERT | SPACE |
| COLLATE | INT | SQL |
| COLLATION | INTEGER | SQLCA |
| COLUMN | INTERSECT | SQLCODE |
| COMMIT | INTERVAL | SQLERROR |
| CONNECT | INTO | SQLSTATE |
| CONNECTION | IS | SQLWARNING |
| CONSTRAINT | ISOLATION | SUBSTRING |
| CONSTRAINTS | JOIN | SUM |
| CONTINUE | KEY | SYSTEM_USER |
| CONVERT | LANGUAGE | TABLE |
| CORRESPONDING | LAST | TEMPORARY |
| COUNT | LEADING | THEN |
| CREATE | LEFT | TIME |
| CROSS | LEVEL | TIMESTAMP |
| CURRENT | LIKE | TIMEZONE_HOUR |
| CURRENT_DATE | LOCAL | TIMEZONE_MINUTE |
| CURRENT_TIME | LOWER | TO |
| CURRENT_TIMESTAMP | MATCH | TRAILING |
| CURRENT_USER | MAX | TRANSACTION |
| CURSOR | MIN | TRANSLATE |
| DATE | MINUTE | TRANSLATION |
| DAY | MODULE | TRIM |
| DEALLOCATE | MONTH | TRUE |
| DEC | NAMES | UNION |

| | | |
|---|---|---|
| DECIMAL | NATIONAL | UNIQUE |
| DECLARE | NATURAL | UNKNOWN |
| DEFAULT | NCHAR | UPDATE |
| DEFERRABLE | NEXT | UPPER |
| DEFERRED | NO | USAGE |
| DELETE | NONE | USER |
| DESC | NOT | USING |
| DESCRIBE | NULL | VALUE |
| DESCRIPTOR | NULLIF | VALUES |
| DIAGNOSTICS | NUMERIC | VARCHAR |
| DISCONNECT | OCTET_LENGTH | VARYING |
| DISTINCT | OF | VIEW |
| DOMAIN | ON | WHEN |
| DOUBLE | ONLY | WHENEVER |
| DROP | OPEN | WHERE |
| ELSE | OPTION | WITH |
| END | OR | WORK |
| END-EXEC | ORDER | WRITE |
| ESCAPE | OUTER | YEAR |
| EXCEPT | OUTPUT | ZONE |
| EXCEPTION | | |

**Future Keywords**
The following keywords could be reserved in future releases of SQL Server as new features are implemented. Consider avoiding the use of these words as identifiers.

| | | |
|---|---|---|
| ABSOLUTE | HOST | RELATIVE |
| ACTION | HOUR | RELEASE |
| ADMIN | IGNORE | RESULT |
| AFTER | IMMEDIATE | RETURNS |
| AGGREGATE | INDICATOR | ROLE |
| ALIAS | INITIALIZE | ROLLUP |
| ALLOCATE | INITIALLY | ROUTINE |
| ARE | INOUT | ROW |
| ARRAY | INPUT | ROWS |
| ASENSITIVE | INT | SAVEPOINT |
| ASSERTION | INTEGER | SCROLL |

| | | |
|---|---|---|
| ASYMMETRIC | INTERSECTION | SCOPE |
| AT | INTERVAL | SEARCH |
| ATOMIC | ISOLATION | SECOND |
| BEFORE | ITERATE | SECTION |
| BINARY | LANGUAGE | SENSITIVE |
| BIT | LARGE | SEQUENCE |
| BLOB | LAST | SESSION |
| BOOLEAN | LATERAL | SETS |
| BOTH | LEADING | SIMILAR |
| BREADTH | LESS | SIZE |
| CALL | LEVEL | SMALLINT |
| CALLED | LIKE_REGEX | SPACE |
| CARDINALITY | LIMIT | SPECIFIC |
| CASCADED | LN | SPECIFICTYPE |
| CAST | LOCAL | SQL |
| CATALOG | LOCALTIME | SQLEXCEPTION |
| CHAR | LOCALTIMESTAMP | SQLSTATE |
| CHARACTER | LOCATOR | SQLWARNING |
| CLASS | MAP | START |
| CLOB | MATCH | STATE |
| COLLATION | MEMBER | STATEMENT |
| COLLECT | METHOD | STATIC |
| COMPLETION | MINUTE | STDDEV_POP |
| CONDITION | MOD | STDDEV_SAMP |
| CONNECT | MODIFIES | STRUCTURE |
| CONNECTION | MODIFY | SUBMULTISET |
| CONSTRAINTS | MODULE | SUBSTRING_REGEX |
| CONSTRUCTOR | MONTH | SYMMETRIC |
| CORR | MULTISET | SYSTEM |
| CORRESPONDING | NAMES | TEMPORARY |
| COVAR_POP | NATURAL | TERMINATE |
| COVAR_SAMP | NCHAR | THAN |
| CUBE | NCLOB | TIME |
| CUME_DIST | NEW | TIMESTAMP |
| CURRENT_CATALOG | NEXT | TIMEZONE_HOUR |
| CURRENT_DEFAULT_TRANSFORM_GROUP | NO | TIMEZONE_MINUTE |

| | | |
|---|---|---|
| CURRENT_PATH | NONE | TRAILING |
| CURRENT_ROLE | NORMALIZE | TRANSLATE_REGEX |
| CURRENT_SCHEMA | NUMERIC | TRANSLATION |
| CURRENT_TRANSFORM_GROUP_FOR_TYPE | OBJECT | TREAT |
| CYCLE | OCCURRENCES_REGEX | TRUE |
| DATA | OLD | UESCAPE |
| DATE | ONLY | UNDER |
| DAY | OPERATION | UNKNOWN |
| DEC | ORDINALITY | UNNEST |
| DECIMAL | OUT | USAGE |
| DEFERRABLE | OVERLAY | USING |
| DEFERRED | OUTPUT | VALUE |
| DEPTH | PAD | VAR_POP |
| DEREF | PARAMETER | VAR_SAMP |
| DESCRIBE | PARAMETERS | VARCHAR |
| DESCRIPTOR | PARTIAL | VARIABLE |
| DESTROY | PARTITION | WHENEVER |
| DESTRUCTOR | PATH | WIDTH_BUCKET |
| DETERMINISTIC | POSTFIX | WITHOUT |
| DICTIONARY | PREFIX | WINDOW |
| DIAGNOSTICS | PREORDER | WITHIN |
| DISCONNECT | PREPARE | WORK |
| DOMAIN | PERCENT_RANK | WRITE |
| DYNAMIC | PERCENTILE_CONT | XMLAGG |
| EACH | PERCENTILE_DISC | XMLATTRIBUTES |
| ELEMENT | POSITION_REGEX | XMLBINARY |
| END-EXEC | PRESERVE | XMLCAST |
| EQUALS | PRIOR | XMLCOMMENT |
| EVERY | PRIVILEGES | XMLCONCAT |
| EXCEPTION | RANGE | XMLDOCUMENT |
| FALSE | READS | XMLELEMENT |
| FILTER | REAL | XMLEXISTS |
| FIRST | RECURSIVE | XMLFOREST |
| FLOAT | REF | XMLITERATE |
| FOUND | REFERENCING | XMLNAMESPACES |
| FREE | REGR_AVGX | XMLPARSE |

| | | |
|---|---|---|
| FULLTEXTTABLE | REGR_AVGY | XMLPI |
| FUSION | REGR_COUNT | XMLQUERY |
| GENERAL | REGR_INTERCEPT | XMLSERIALIZE |
| GET | REGR_R2 | XMLTABLE |
| GLOBAL | REGR_SLOPE | XMLTEXT |
| GO | REGR_SXX | XMLVALIDATE |
| GROUPING | REGR_SXY | YEAR |
| HOLD | REGR_SYY | ZONE |

# APPENDIX D: MN/DOT MODIFIER WORDS AND ABBREVIATIONS

| Modifier Words | Acronym/ Abbreviation | Definition | Recommended Data Type |
|---|---|---|---|
| Above | ABV | | |
| Accident | ACC | | |
| Action | ACT | | |
| Address | ADDR | | |
| Agency | AGCY | | |
| Agreement | AGR / AGRMT | | |
| Amount | AMT | A monetary value. | |
| Audit | AUD | | |
| Average | AVG | | |
| Block | BLK | | |
| Category | CAT | A specifically defined division or subset in a system of classification in which all items share the same concept of taxonomy. | |
| Check | CHK | | |
| Class | CLS | | |
| Code | CD | A combination of one or more numbers, letters, or special characters which is submitted for a specific meaning (includes Status and Abbreviation). | |
| Comment | CMT | A explanatory/illustrative/critical note, remark, or observation. | |
| Count | CNT | | |
| County | CNTY | | |
| City | CTY | | |
| Daily | DLY | | |
| Date | DT | The notion of a specific period of time. | |
| Day | DAY | The notion of a specific period of time. | |
| Description | DESC / DSC | A word or phrase which interprets an abbreviation, code or other shortened term so as to make it more understandable to users (i.e., Minnesota is the description of the abbreviation MN). | |
| District | DIST / DSTR | | |
| Division | DIV | | |

| | | | |
|---|---|---|---|
| Error | ERR | | |
| Flag | FLG | An alphanumeric "switch" with a maximum of two values representing the opposite of each other. | |
| History | HIST / HST | | |
| Hours | HR / HRS | The notion of a specific period of time. | |
| Identifier | ID | A system generated unique number used internally within the database and not usually known by the user. This is usually generated by a sequence. | |
| Indicator | IND / INDC | An identifier which has a domain of only 2 states, such as: Y/N, ON/OFF,TRUE/FALSE, 0/1, etc. | |
| Leave | LV | | |
| License | LIC | | |
| Location | LOC | | |
| Manager | MGR | | |
| Name | NM | The designation of an object/entity expressed in a word or phrase. | |
| Month | MTH / MNTH | Calendar Month. | |
| Number | NBR / NUM / NMBR | A value which is not for the purpose of measuring a quantity or serving as a counter, but which is usually a numeric value (it may contain non-numeric characters, such as a Product Number, PO Number, etc.). It is distinguishable in that arithmetic operations are not usually carried out on it. | |
| Object | OBJ | | |
| Paid | PD | | |
| Quantity | QTY | A non-monetary value. (Includes: Average, Balance, Deviation, Factor, Index, Level, Mean, Median) | |
| Rate | RT | A quantity, amount or degree of something in relation to units of something else. (Includes: Factor, Frequency, Percent, Scale) | |
| Report | RPT | | |
| Round | RND | | |
| Square | SQ | | |
| Text | TXT | An unformatted character string, generally in the form of words. (Includes: Abbreviation, Comments) | |
| Time | TIME | A notion of a specified chronological point within a period. | |
| Total | TOT | | |
| Type | TYP / TP | | |
| Volume | VOL | | |
| Week | WK | The notion of a specific period of time. | |
| Value | VAL | | |

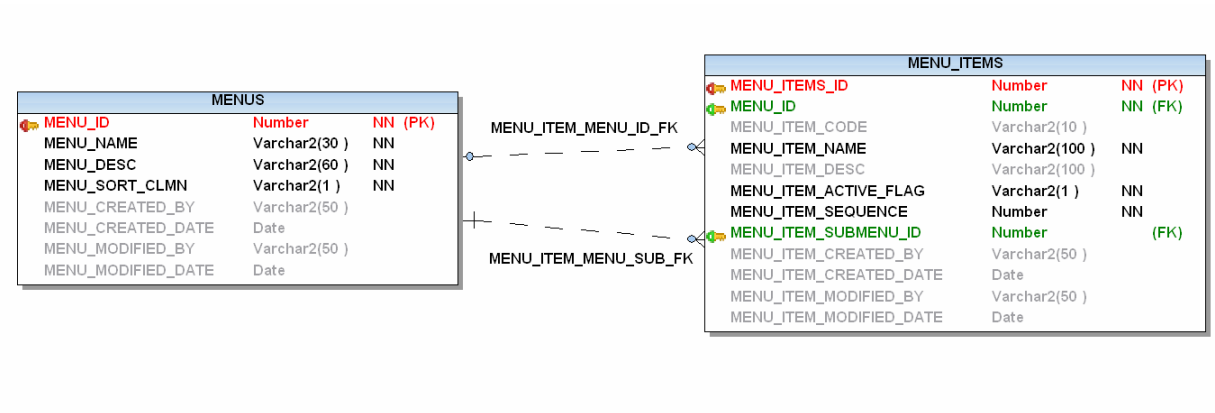
| | | | |
|---|---|---|---|
| Year | YR | The notion of a specific period of time. | |

# APPENDIX E: MENU – MENU ITEMS LOOKUP TABLE DESIGN

Menu – Menu Items was designed to replace many of the small standard lookup (Menu) tables with a generic set of tables to make Menu Maintenance easier. Typically most lookup tables were made up of a Code / ID and a Description.

## Menus – Menu Items Option 1

This basic design of the MENUS and MENU_ITEMS Table is simple one-to-many relationship between MENUS and MENU_ITEMS. The design also allows Menus to be linked to other menus allowing for a hierarchal menu design using the MENU_ITEM_SUBMENU_ID which can be linked to another MENU_ID.




**MENUS** Table holds the names of the separate menus.

**MENU_ITEMS** Table holds the records that make up the list of items for the Menu.

**FUNC_GET_MENU_OPTION** function can be used as a generic call to retrieve the Menu Items Name to avoid having to join tables to MENU_ITEMS Table for queries, views, and/or reports.

### Foreign Key Column Naming Conventions using Menu – Menu Items

We use the Menu Name (Spaces replaced by "_" + "_ID" to name the columns in a table that are referencing the Menu-Menu Items Tables.

**Example Foreign Key Naming**

For example we have a MENU Named “COLOR” with a list of varies Colors in the MENU_ITEMS Table. In the WIDGETS Table we would like to have a Color Identifier. The Column in the WIDGETS Table would be named “COLOR_ID” which would have a Foreign Key Relationship to the Primary Key Column MENU_ITEMS_ID in the MENU_ITEMS Table.

As another example we have a MENU Named “WARRANTY PERIOD”, the column name would be “WARRANTY_PERIOD_ID”.

Note: It may be necessary to use appropriate abbreviations of the Column Name to conform to Oracle’s Column Naming Length Requirements.

WIDGET Table

| Column Name | Data Type |
|---|---|
| WIDGET_ID | NUMBER |
| … | |
| COLOR_ID | NUMBER (Foreign Key to MENU_ITEMS. MENU_ITEMS_ID |
| … | |
| WARRANTY_PERIOD_ID | NUMBER (Foreign Key to MENU_ITEMS. MENU_ITEMS_ID |

### MENUS Table Column Descriptions

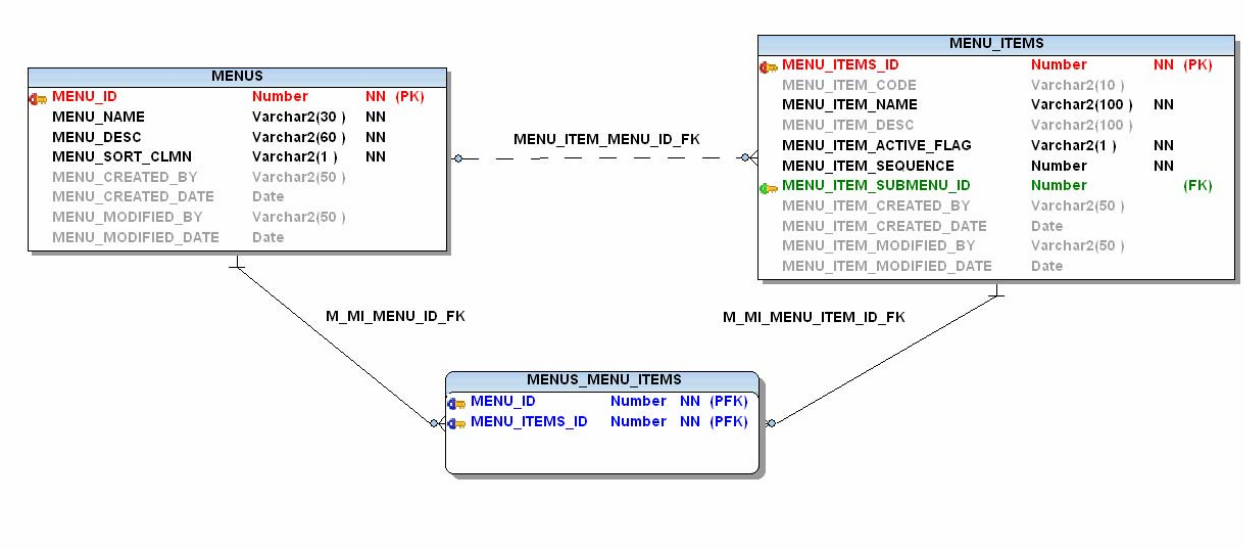
| COLUMN_NAME | DATA_TYPE | LENGTH | BUSINESS_NAME | BUSINESS_DESCRIPTION |
|---|---|---|---|---|
| MENU_ID | NUMBER | 22 | Menu ID | Primary Key of or Foreign Key to MENU Table - User Defined Number. |
| MENU_NAME | VARCHAR2 | 30 | Menu Name | Name of the Menu. |
| MENU_DESC | VARCHAR2 | 60 | Menu Description | Short description of Menu. |
| MENU_SORT_CLMN | VARCHAR2 | 1 | Menu Sort Column | A one character indication of which column of the MENU_ITEMS table to sort on. N for MENU_ITEMS_NAME, S for MENU_ITEMS_SEQUENCE, C for MENU_ITEMS_CODE |
| MENU_CREATED_BY | VARCHAR2 | 50 | Created By | User ID of the person who created the Record. Standard Record Audit Field. |
| MENU_CREATED_DATE | DATE | 7 | Created Date | Date of when Record was created. Standard Record Audit Field. |
| MENU_MODIFIED_BY | VARCHAR2 | 50 | Modified By | User ID of the last person to modify the Record. Standard Record Audit Field. |
| MENU_MODIFIED_DATE | DATE | 7 | Modified Date | Last date Record was changed. Standard Record Audit Field. |

### MENU_ITEMS Table Column Descriptions

| COLUMN_NAME | DATA_TYPE | LENGTH | BUSINESS_NAME | BUSINESS_DESCRIPTION |
|---|---|---|---|---|
| MENU_ITEMS_ID | NUMBER | 22 | Menu Items ID | Primary key of MENU_ITEMS Table - Numeric Sequence MENU_ITEMS_ID_SEQ. |
| MENU_ID | VARCHAR2 | 4 | Menu ID | Primary Key of or Foreign Key to MENU Table - User Defined Number. |
| MENU_ITEM_CODE | VARCHAR2 | 10 | Menu Item Code | Common Code Name for Menu Item |
| MENU_ITEM_NAME | VARCHAR2 | 100 | Menu Item Name | Name of the Menu Item. |
| MENU_ITEM_DESC | VARCHAR2 | 100 | Menu Item Description | Short description of Menu Item |
| MENU_ITEM_ACTIVE_FLAG | VARCHAR2 | 1 | Menu Item Active Flag | Y/N value used to activate/deactivate menu items from display. |
| MENU_ITEM_SEQUENCE | NUMBER | 22 | Menu Item Sequence | Sequential numbering field for menu items to be used to provide a sort order for the Menu Items. |
| MENU_ITEM_SUBMENU_ID | VARCHAR2 | 22 | Menu Item Submenu ID | Primary Key of or Foreign Key to MENU Table - User Defined Number. Allows for the linkage of a Menu Item to another Menu. |
| MENU_ITEM_CREATED_BY | VARCHAR2 | 50 | Created By | User Id of the person who created the Record. Standard Record Audit Field. |
| MENU_ITEM_CREATED_DATE | DATE | 7 | Created Date | Date of when Record was created. Standard Record Audit Field. |
| MENU_ITEM_MODIFIED_BY | VARCHAR2 | 50 | Modified By | User ID of the last person to modify the Record. Standard Record Audit Field. |
| MENU_ITEM_MODIFIED_DATE | DATE | 7 | Modified Date | Last date Record was changed. Standard Record Audit Field. |

## Menus – Menu Items Option 2

This model is designed to allow the MENUS and MENU_ITEMS Table to have a many-to-many Relationship between them using the MENUS_MENU_ITEMS intersection table. This allows reuse of Menus and can be used for more complex menu structures.




**MENUS** Table holds the names of the separate menus.

**MENU_ITEMS** Table holds the records that make up the list of items for the Menu. One difference in this design the MENU_ID Column has been removed from the table so that Menu Items can be used in Multiple Menus.

**MENUS_MENU_ITEMS** Table holds the intersection Primary Key Columns between the MENUS and MENU_ITEMS Table.

**FUNC_GET_MENU_OPTION** function can be used as a generic call to retrieve the Menu Items Name to avoid having to join tables to MENU_ITEMS Table for queries, views, and/or reports.

**MENUS_MENU_ITEMS_VW** view combines the MENUS_MENU_ITEMS to the MENU_ITEMS you can produce the same feel as the original design where MENU_ITEMS records were associated to only one MENUS record.

**<u>Foreign Key Column Naming Conventions using Menu – Menu Items</u>**

Use the same convention from Option 1.

### MENUS Table Column Descriptions

| COLUMN_NAME | DATA_TYPE | LENGTH | BUSINESS_NAME | BUSINESS_DESCRIPTION |
|---|---|---|---|---|
| MENU_ID | NUMBER | 22 | Menu ID | Primary Key of or Foreign Key to MENU Table - User Defined Number. |
| MENU_NAME | VARCHAR2 | 30 | Menu Name | Name of the Menu. |
| MENU_DESC | VARCHAR2 | 60 | Menu Description | Short description of Menu. |
| MENU_SORT_CLMN | VARCHAR2 | 1 | Menu Sort Column | A one character indication of which column of the MENU_ITEMS table to sort on. N for MENU_ITEMS_NAME, S for MENU_ITEMS_SEQUENCE, C for MENU_ITEMS_CODE |
| MENU_CREATED_BY | VARCHAR2 | 50 | Created By | User ID of the person who created the Record. Standard Record Audit Field. |
| MENU_CREATED_DATE | DATE | 7 | Created Date | Date of when Record was created. Standard Record Audit Field. |
| MENU_MODIFIED_BY | VARCHAR2 | 50 | Modified By | User ID of the last person to modify the Record. Standard Record Audit Field. |
| MENU_MODIFIED_DATE | DATE | 7 | Modified Date | Last date Record was changed. Standard Record Audit Field. |

### MENU_ITEMS Table Column Descriptions

| COLUMN_NAME | DATA_TYPE | LENGTH | BUSINESS_NAME | BUSINESS_DESCRIPTION |
|---|---|---|---|---|
| MENU_ITEMS_ID | NUMBER | 22 | Menu Items ID | Primary key of MENU_ITEMS Table - Numeric Sequence MENU_ITEMS_ID_SEQ. |
| MENU_ITEM_CODE | VARCHAR2 | 10 | Menu Item Code | Common Code Name for Menu Item |
| MENU_ITEM_NAME | VARCHAR2 | 100 | Menu Item Name | Name of the Menu Item. |
| MENU_ITEM_DESC | VARCHAR2 | 100 | Menu Item Description | Short description of Menu Item |
| MENU_ITEM_ACTIVE_FLAG | VARCHAR2 | 1 | Menu Item Active Flag | Y/N value used to activate/deactivate menu items from display. |
| MENU_ITEM_SEQUENCE | NUMBER | 22 | Menu Item Sequence | Sequential numbering field for menu items to be used to provide a sort order for the Menu Items. |
| MENU_ITEM_SUBMENU_ID | NUMBER | 22 | Menu Item Submenu ID | Primary Key of or Foreign Key to MENU Table - User Defined Number. Allows for the linkage of a Menu Item to another Menu. |
| MENU_ITEM_CREATED_BY | VARCHAR2 | 50 | Created By | User Id of the person who created the Record. Standard Record Audit Field. |
| MENU_ITEM_CREATED_DATE | DATE | 7 | Created Date | Date of when Record was created. Standard Record Audit Field. |
| MENU_ITEM_MODIFIED_BY | VARCHAR2 | 50 | Modified By | User ID of the last person to modify the Record. Standard Record Audit Field. |
| MENU_ITEM_MODIFIED_DATE | DATE | 7 | Modified Date | Last date Record was changed. Standard Record Audit Field. |

### MENU_ITEMS Table Column Descriptions

| COLUMN_NAME | DATA_TYPE | LENGTH | BUSINESS_NAME | BUSINESS_DESCRIPTION |
|---|---|---|---|---|
| MENU_ID | NUMBER | 22 | Menu ID | Primary Key of or Foreign Key to MENU Table - User Defined Number. |
| MENU_ITEMS_ID | NUMBER | 22 | Menu Items ID | Primary key of MENU_ITEMS Table - Numeric Sequence MENU_ITEMS_ID_SEQ. |

## Menu – Menu Items Support Objects

### FUNC_GET_MENU_OPTION Function SQL

```
CREATE OR REPLACE FUNCTION func_get_menu_option (p_input_menu_items_id IN
NUMBER)
RETURN VARCHAR2 IS

  v_menu_option   menu_items.menu_item_name%TYPE;

BEGIN

  IF p_input_menu_items_id IS NULL THEN

    v_menu_option    := NULL;

  ELSE

    SELECT  menu_item_name
    INTO    v_menu_option
    FROM    menu_items
    WHERE   menu_items_id = p_input_menu_items_id;

  END IF;

  RETURN v_menu_option;

END func_get_menu_option;
/
```

### MENUS_MENU_ITEMS_VW View SQL

```
CREATE OR REPLACE FORCE VIEW MENUS_MENU_ITEMS_VW
AS
SELECT   mmi.menu id,
         mi.menu items id,
         mi.menu item code,
         mi.menu item name,
         mi.menu item desc,
         mi.menu item active flag,
         mi.menu item sequence,
         mi.menu item submenu id,
         mi.menu item created by,
         mi.menu item created date,
         mi.menu item modified by,
         mi.menu item modified_date
FROM     menus menu items mmi,
         menu items       mi
WHERE    mmi.menu_items_id = mi.menu_items_id;
```

# APPENDIX F: METADATA GUIDELINES

## Oracle Database Comment

Mn/DOT uses the Database Comment Command to document Tables, Materialized Views, Views, and their associated Columns in the database. These comments are portable and can be included as part of the deployment scripts for an application. Comments can be accessed using the Database Tools that are provided by Mn/DOT or through specialized tables, views, scripts that are also provide to assist Application end users, application administrators, developers, DBA's, and contractors with Metadata to aid in the use and support of an application.

Oracle 10g/11g allows comments to have a maximum length of 4000 Characters.

### COMMENT COMMAND SYNTAX

| Format Values | Format Description |
|---|---|
| {Owner) | Schema Owner that owns the Table Materialized View, or View. |
| {Object_Name} | Table, Materialized View, or View Name. |
| {Column Name} | Name of the column. |
| {Business Name} | Business Name of the Table, Materialized View, View or Associated Column Name spelled out with no abbreviations. |
| {Description Section} | A description of the Table, Materialized View, View or Associated Column Name and its purpose defined in technical and business terms. |

Tables and Views use the following Syntax:

COMMENT ON TABLE **{Owner}**.**{Object_Name}** IS '[**{Business Name}**] – **{Descriptive Section}**.';

Materialized Views use the following Syntax:

COMMENT ON MATERILIZED VIEW **{Owner}**.**{Object_Name}** IS '[**{Business Name}**] – **{Descriptive Section}**.';

Columns use the following syntax:

COMMENT ON COLUMN **{Owner}**.**{Object_Name}**.**{Column_Name}** IS '[**{Business Name}**] – **{Descriptive Section }**. ';

# SQL Server Database Comment

SQL Server supports a three-part naming convention when you refer to the current server. The ISO standard also supports a three-part naming convention. However, the names used in both naming conventions are different. The information schema views are defined in a special schema named **INFORMATION_SCHEMA**. This schema is contained in each database. Each information schema view contains metadata for all data objects stored in that particular database.

Adding metadata to a table, column or view is accomplished by using the Extended Properties in SQL Server. If you enter text in the 'Description' property of a column using the Management Studio, you are creating an Extended Property called 'MS_Description'. The GUI interface to extended properties is somewhat cumbersome if you are adding metadata to several tables, columns or views.

Extended Properties (Comments) can be accessed using the Database Tools that are provided by Mn/DOT or through specialized tables, views, scripts that are also provide to assist Application end users, application administrators, developers, DBA's, and contractors with Metadata to aid in the use and support of an application.

## Adding metadata using the GUI tools:

Using the Management Studio GUI, you can highlight the database, table, view, etc. then right click and select properties, and then extended properties. This will allow you to create or modify metadata on the object you have selected.

## Adding metadata using T-SQL:

Using T-SQL to script the metadata entries you can add or replace an extended property value.

### EXTENDED PROPERTY (COMMENT) COMMAND SYNTAX

**Add Extended Property to a database:**

```
USE "DATABASE NAME"
GO
EXEC sp_addextendedproperty
  @name=N'MS_Description',
  @value=N"[{Business Name}] – {Descriptive Section }..'
GO
```

**Add an Extended Property to a Table/Column:**

```
USE “DATABASE NAME”
GO
exec sp_addextendedproperty 'MS_Description',
'[{Business Name}] – {Descriptive Section }.',
'schema',
'dbo',
'table',
'table name',
'column',
'column name'
GO
```

## Sample Comments

[{Business Name}] – {Descriptive Section }.

| |
|---|
| **Audit Columns - c**olumns used to track who and when records were changed, created, or deleted. |
| [Created By] - User Id of the person who created the Record. Standard Record Audit Field. |
| [Created Date] - Date of when Record was created. Standard Record Audit Field. |
| [Deleted By] - User ID of the person who deleted the record. Standard Record Audit Field. |
| [Deleted Date] - Date Record was deleted. Standard Record Audit Field. |
| [Modified By] - User ID of the last person to modify the Record. Standard Record Audit Field. |
| [Modified Date] - Last date Record was changed. Standard Record Audit Field. |
| |
| **Financial** |
| [Appropriation Unit Number] - Number which identifies the unit of the related appropriation account. |
| [Fiscal Year] - State fiscal year. A reference to the fiscal year may be to the year of accounting significance, budgetary significance, or the year that the chart of accounts was valid, depending upon the table context. |
| [Fund Code] - MAPS - Code which represents a specific fund. |
| [Organization Number] - Number which identifies the subordinate unit of an agency. All documents processed in MAPS must have an organization code. |
| |
| **Miscellaneous** |
| [Active Flag] - Indicates if a record is active (T) or not (F). |
| [Active Flag] - Indicates if a record is active (Y) or not (N). |
| [Comment] - Text Body of the Note or Comment related to the record. |
| [Title] - Short Description/Title of the used to describe the record. |
| |
| **Person or Employee Name, Address, Phone, etc** |
| [Employee Number] - Number which identifies an Employee. A unique SEMA4-assigned number used to identify an employee throughout his or her state career. Numbers from the old system are carried over into SEMA4. |
| [Last Name] - Last Name of a Person. |

| |
|---|
| [First Name] - First Name of a Person. |
| [Middle Name] - Middle Name of a Person. |
| [Email Address] - External or Internet Email Address for Business or Person that they receive electronic mail. |
| [Office Section Name] - Name of the Office/Business and Specific Section/Unit. |
| [Phone Number] - Phone Number of a Company or Person. |
| [Fax Phone Number] - Fax Phone Number of a Company or Person. |
| [Street Address] - Address which locates a Business, house or apartment number and street or route of a person. |
| [Mail Stop] - Internal Address which locates the mail drop for the Business or Person. |
| [City ID] - Numeric equivalent of the CITY_CODE. |
| [City Code] - Primary Key of or Foreign Key to CITIES Table - 4 Character String. |
| [City Name] - Name of the city of the employees address or of the location code for a state office. |
| [State Name] - Code or Name which represents the U.S. Postal Service two-character state identifier/name for a Business or Person. |
| [Zip Code] - Code which represents the U.S. Postal Service zip code designation for a state office or an employee`s address. |
| [County ID] - Numeric equivalent of the COUNTY_CODE. |
| [County Code] - Primary Key of or Foreign Key to COUNTIES Table - 1-2 Character String. |
| [County Name] - Name of the County. |
| [Country Name] - Country Name where the Person or Company is located. i.e. Canada, USA, |

**Menu – Menu Items**

[{Column  Business Name}] - MENU / MENU_ITEMS Tables.  MENU_ID = {Menu ID} - {Description}.

**Primary Key -** Columns that are referenced in Foreign Keys can use the same comment as the Primary Key that they are referencing.

Exception see Menu – Menu Items is actually replacing a lookup table so you would not use the Primary Key Comment in this case.

[{Column  Business Name}] - Primary Key of or Foreign Key to {Table_Name} Table – Numeric Sequence {Sequence_Name}.';

[{Column  Business Name}] - Primary Key of or Foreign Key to {Table_Name} Table - User Defined Character String.';

[{Column  Business Name}] - Primary Key of or Foreign Key to {Table_Name} Table – Calculated by Application.';

# APPENDIX G: SECURITY MODELS

## Introduction

The purpose of this document is to describe the managed-database policy for the Oracle database environment at Mn/DOT, along with naming convention requirements and process controls for Programmers and Contractors. It is designed to clarify the policies and procedures for database development.

## Accounts

Oracle accounts are classified into three categories:

- Application account
- GIS Analyst account
- Developer account
- GIS Editor account
- Oracle Forms user account
- Reports account
- User account

### Developer Account

Developer Accounts are assigned to specific individuals. These accounts are used to experiment with Oracle. Individual accounts should never be used to hold project data. Generally, individual accounts are set up to have privileges to connect to Oracle and create objects in their own schema.

### Application Account

Application accounts are assigned to a project. All Oracle objects created for an application should be owned by the application account. These accounts are generally given a name that identifies a project.

### Abbreviation Overview

| | | |
|---|---|---|
| {app_alias} | - | Refers to the acronym for the application. |
| ### | - | 1-5 Letter/Number Code |
| ??? | - | 1-5 Letter/Number Code |

## Application Account Types

AGS_{app_alias}

IMS_{app_alias}

{app_alias}_ADMIN

- Account that owns Application Database Objects (schema owner).
- Account will be locked or unlocked based on Environment.

{app_alias}_USER

- Account used for User Access by Application Server
- Account will have required CRUD Data Access.

{app_alias}_REPORTS

- Account used for Crystal Report Access.
- Account should have only SELECT (Read) Data Access.

{app_alias}_LINK (Optional Account for Database Links)

- Account used for database links by other databases.
- Account should have the minimal CRUD Data Access required to support the Database Applications using the Database Link for Data Access.
- Grants can be through Database Role or Direct for Schema Admin Account.

{app_alias}_WEB (Optional Account for Secure Web Reporting)

- Account used for Business Objects Enterprise Web Interface reference table look-up.
- Account should have only SELECT (Read) Data Access.

## Account Locking

An account may become locked for one of several reasons: the Oracle DBA has locked the account to prevent access or more than 3 consecutive unsuccessful attempts have been made to access an Oracle account. Oracle will not permit connections to an Oracle instance using a locked account. In most cases, the only way to unlock an account is to contact the Oracle DBA.

## Instances

Autonomous database environments which can be managed completely independently from one another. Each instance fulfills a specific role in the development of software for Oracle databases.

### Development Environment

- This instance is used for developing databases.
- The purpose is to provide a place where developers can build and experiment with the database and application.
- Developers are granted the most privileges in this instance.
- This instance should never be used as a repository for production data.
- Developers will have access to all accounts.
- Admin Account will be locked during fixed testing periods.
- All Database modifications will be reviewed by DBA.
- DBA/Developers must review and create change package from Development to be applied to Test.
- Passwords for accounts will be provided by the Mn/DOT DBA. Stricter password rules will be followed in all environments when the data is considered non-public.

### Test Environment

- This instance is used for testing application code and database changes against a clean version of the database.
- This instance is used to simulate the real production environment.
- This instance should never be used as a repository for production data.
- Developer access will be limited to ensure proper testing by customers.
- Admin Account will be locked during fixed testing periods.
- DBA/Developers must supply database change package from Test Environment to be deployed to Production Environment.
- Passwords for accounts will be provided by the Mn/DOT DBA as needed to application support staff. Stricter password rules will be followed in all environments when the data is considered non-public.

### Production Environment

- This instance is the only instance that should be used to hold production data and be accessed via applications by end-users.
- This instance is the most stable database instance.
- Any changes made to the production database should go through the development and test databases first.
- Developer access will be limited to data maintenance.
- Admin Account will always be locked except for Application Database Maintenance Periods.

- Database account will be backed up prior to deployment of changes and optionally backed up after the change was deployed in the event the data will be used to refresh Development or Test Environments.
- Passwords for accounts will be provided by the Mn/DOT DBA as needed to application support staff. Stricter password rules will be followed in all environments when the data is considered non-public.

## Roles

Oracle provides for easy and controlled privilege management through roles. Roles are named groups of related database privileges that are granted to Oracle user ids or other roles. Each application schema receives two basic roles by default in its creation. These roles contain the database access privileges to an application schema's objects.

## Mn/DOT Security Model Configuration #1

Characteristics of this model:

1. Requires 3-5 specific fixed application specific user accounts for application and/or report access.
2. Deployed to a shared database environment with other similar types of applications.
3. Uses Private Synonyms since the number accounts remains fixed.
4. User authentication comes from a directory service or application specific user management tables.

## Mn/DOT Security Model Configuration #2

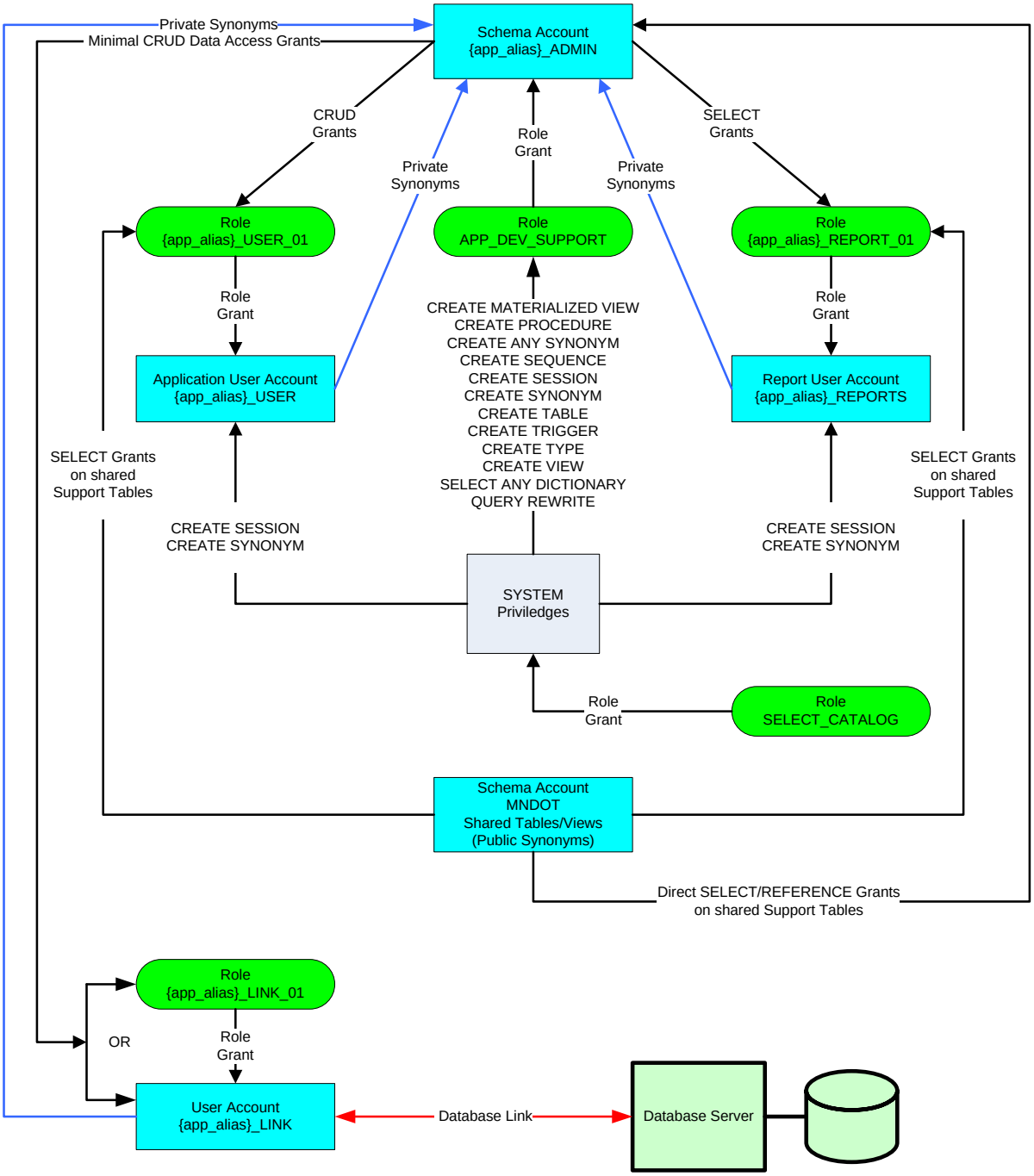
Characteristics of this model

1. Individual database user accounts are setup for each user.
2. Application database is generally dedicated to its own database instance.
3. Uses Public Synonyms since the number of accounts changes overtime.
4. User authentication comes from the database instance.

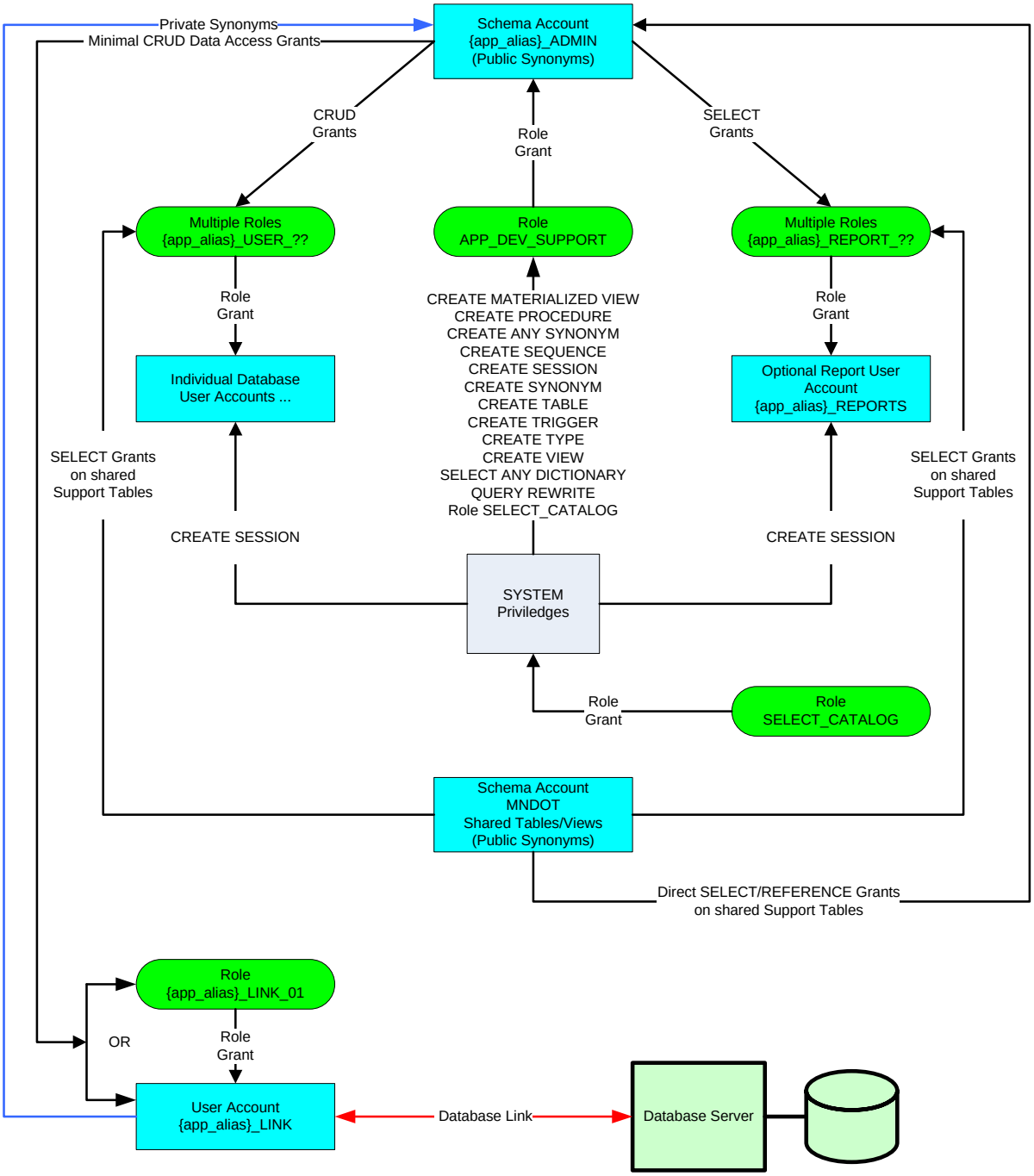
Last Updated: May 2010
Rev. 1.1

## Mn/DOT Security Model Configuration #1

Note: Used by Java, .NET, Access Applications where the application connects using a application specific user account.
Generally private synonyms are used in this type of configuration.



Last Updated: May 2010
Rev. 1.1

## Mn/DOT Security Model Configuration #2

Note: Used by Applications where users have individual user accounts on the database.
Generally Public Synonyms are used in this case.