****

**Specification**

**for**

**9/28/09**

IderaLogo

Proprietary and Confidential to BBS Technologies, Inc.

**© 2009 BBS Technologies, Inc.; all rights reserved.**

# Revision history:

|  |  |  |
| --- | --- | --- |
| **When?** | **Who?** | **What?** |
| 7/27/2009 | Barry | Initail draft |
| 8/11/2009 | Barry | Revision 1 |
| 8/12/2009 | Barry | Revision 2 |
| 8/25/2009 | Barry | Changes as per spec review |
| 9/28/2009 | Barry | Added open issues |

# Table of Contents

1. Revision history: ii

2. Table of Contents iii

3. Requirements 1

3.1. Overview/Purpose 1

3.1.1. Related Customer Requests 1

3.2. Target Users 2

3.3. Use Cases 2

3.4. Feature/Function Market Requirements 3

3.4.1. Required Functions 3

3.4.2. Non-Supported Functions 3

3.5. FAQ 3

3.6. Open Issues 3

4. Functional Design 3

4.1. User Interfaces 3

4.1.1. Replication Topology Grid: 4

4.1.2. Details Tab 4

4.1.3. Screen Shots of view layout 6

4.1.4. New Distribution Latency Alert 7

4.2. Installation and Upgrade 8

4.3. Permissions and other Required Configuration 8

4.4. Licensing Issues 8

4.5. Dependencies 8

4.5.1. Dependencies on Idera Software 8

4.5.2. Third-Party Software Required on the Customer Machine 8

4.5.3. Third-Party Software Required Internally 8

5. Internal Design 9

5.1. Architecture 9

5.1.1. Scheduled refesh 9

5.1.2. Saving the replication data 10

5.1.3. The Replication View 12

5.2. Installation Issues 12

5.3. Schedule 12

5.3.1. Work Breakdown and Sizings 12

5.3.2. Areas of Risk 13

6. Quality Assurance Considerations 13

6.1.1. Overview 13

6.1.2. Developer-Created Unit Tests 13

7. Documentation Considerations 13

8. Bibliography 14

# Requirements

## Overview/Purpose

Replication functionality in SQLdm is somewhat limited and the time has come to address these issues.

We currently do not alert on subscription latency if the distributor is not installed on the publisher. This needs to be addressed such that we deal with a remote publisher and connect accordingly.

We only show information when connected to the publisher. The replication view shows nothing when connected to the distributor or subscriber. The view should probe the replication topology and connect accordingly, ideally showing the topology as well as whatever information can be gleaned from the monitored servers.

We show the distribution (publisher) and subscription (distributor) queues in the view. This is not very useful. What would be more useful would be to see the topology and key metrics detailing the current state of replication on all machines that are related to replication on the monitored server.

### Related Customer Requests

I have highlighted the items that have not been addressed in this spec.

* We have had a number of requests from customers, the most detailed being a call with Hilary Cotter. His key points were that we should:
* Display topology.
* Display subscription latency.
* Display number of unsubscribed transactions.
* Display subscription rate.
* Display unsubscribed\subscription rate = estimated time to apply queue.
* This is just a guess of course. I will put it in and we can gauge the efficacy of this metric. It is likely that many folks will always sit with a estimated catch-up time that they will never reach because the queue is always getting longer.
* Display the distribution latency.
* **2014540 – After migrating the existing database, replication is no longer working.**
* The customer has replied that he can’t configure the distribution database. This means that replication is rather broken. I am not sure if we will fix him but being able to view replication from the perspective of the distributor will certainly be a good start.
* 2011938 – Provide real-time information even if log reader s in use.
* We do not rely on the log reader now that we do not try and show the publication queue.
* 2011863 – Show replication from the subscribers perspective
* We now cater for showing replication from all 3 servers perspectives, as long as they are all being monitored. If only the subscriber is being monitored there is literally nothing to show.
* 2011860 – Publisher queue filter returns too many rows.
* Publisher queue has been removed.
* 2011324 – The message that is displayed when there is no data in the queues is not descriptive enough. It just says “there are no items to show”.
* We will now show a lot more information rather than 2 empty grids.
* 2010988 – FRQ: Alert on “replication configuration latency”. I take this to mean distribution latency.
* This alert has been added in the spec.
* 2010546 – View does not show merge replication.
* Merge replication is very different that transactional and almost warrants a separate view. I will include the ability to view merge as well as transactional replication in the topology, but not details.
* 2010525 – All replication alerts should be based on the publisher and not on the distributor, especially since you cannot even view replication on the publisher.
* This is addressed by the fact that we show replication on all participants. We also show the user which entity the alert belongs to on the view.
* 2010498 – command text gives less info now than in 4.1.
* I will attempt to improve the data returned by the non-subscribed queue when I get deep in the code.
* **205228 – Replication alerts, independent of job alerting.**
* We are not looking at adding monitoring of replication specific jobs for this release. This functionality has changed significantly since this was requested. I recommend this be closed.

## Target Users

Users of any version of SQL Server since and including SQL2000. A large number of replication implementations would seem to be SQL 2000, so it is very important that we do not get caught up in the new functions, procedures and dmv’s that have been added for monitoring replication on SQL 2008.

## Use Cases

* DBA alerted to excessive subscription latency
* DBA alerted to excessive distribution latency.
* DBA alerted to excessive non-subscribed transactions.
* DBA alerted to excessive non-distributed transactions.
* DBA views replication status on the publisher.
* DBA views replication status on the distributor.
* DBA views replication status on the subscriber in a pull configuration (agents on subscriber).
* DBA views replication status on the subscriber in a push configuration (agents on distributor).
* DBA needs to see the replication topology and the server to which he is connected highlighted to show how it participates in the replication session.
* DBA could be viewing a subscriber that has subscriptions to multiple distributors. He should see all subscriptions on the server that he has selected, as well as the distribution and publication information that we can get to (whatever he is monitoring).
* DBA view replication topology on server that is participating in a merge replication session.

## Feature/Function Market Requirements

This feature is primarily focused on displaying transactional replication topology when connected to any of the monitored replication participants. We need to provide the user with a one-stop overview of replicated sessions on their system and key metrics that display the health of their replication.

### Required Functions

The primary purpose of this feature is to deal with displaying comprehensive transactional replication data.

Alert the user to excessive non-subscribed transactions regardless of network topology. The only requirement is that the user is monitoring the publisher and distributor. Alert the user to excessive non-distributed transactions, non-distributed queue, non-subscribed queue.

Populate the replication view when any of the participating servers are selected, without being blocked by replication itself.

### Non-Supported Functions

If the user is not monitoring the publisher and the distributor, we cannot get to any meaningful replication information.

We do not show the distribution queue. Use of sp\_replcommands is not recommended because it causes a lock on the log reader or it fails if replication itself has locked the log reader. The value of seeing the distribution queue is debatable in any case.

We do not support snapshot replication. That is to say it is not the focus of this feature. Snapshot replication has almost no real-time diagnostics because it is more of a job scheduling exercise.

The history browser in not supported in this feature.

## FAQ

This section should present a high level FAQ that is used to answer common questions from others as they read the spec and also that customers using the feature may ask. This should also provide a start for “He said, She said” documents for sales.

## Open Issues

* Subscribed transactions is an ever increasing counter. It is actually of limited value and the rate at which it is increasing is of more value. I hae remove subscribed from the graph as it does not make sense next to non-subscribed which is the number of non-subscribed transactions at a point in time.
* Show graphs of all metrics that we alert on. Subscription and distribution latency, subscription and distribution queue lengths.
* The graphs do not fill with data in the context of the selected grid row.. If we are connected to a server(serverA) and it is a subscriber in a session we will show it under the replication topology. Clicking on the row will result in a connection to the publisher and distributor of the session as well as to serverA to get the subscriber details. We do not have a connection to server that can just get the sum of the distribution and subscription counters to be added to the charts. What we would have to do is create a specific probe which can go and fetch these specific counters, summed for this server and add the results to the chart. This will mean creating an additional probe that is much like the scheduled refresh probe that returns these counter values.
* Is this worth the overhead.
* So clicking on the row would cause the following:
* Connection to the publisher to get the non-distributed metrics for the selected publication.
* Connection to the distributor to get the non-subscribed metrics for the selected publication.
* Connection to the subscriber to get the subscriber (ServerA)metrics.
* Connection to ServerA to get the chart metrics for the server as a whole.
* Consider a customizable replication batch timeout for customers with extreme replication conditions (Jason VD)
* Only graph those metrics that are alerted on.

# Functional Design

## User Interfaces

When the user opens the view they should be presented with an overview of the replication topology for all replication relationships that the selected server is participating in. This comes from our repository and provides the view with connection information to the replication participant instances.

### Replication Topology Grid:

Show only topological information that is gathered from the Replication Topology table.

#### Columns

* Number of Articles
* Publisher (server..database)
* Publisher Servers status – Icon showing the status of the connection to this server in SQLdm.
* Distributor (server..database)
* Distributor Server Status
* Subscriber (server..database)
* Subscriber server status
* Publication name
* Replication Type

##### Navigation

Right - clicking on a row offers a pop-up menu that allows the user to open the view from the perspective of the other replication participants that are in the selected row.

* Navigate to Publisher
* Navigate to Distributor
* Navigate to Subscriber

The menus are only enabled if the target server is monitored and the user has view rights to it.

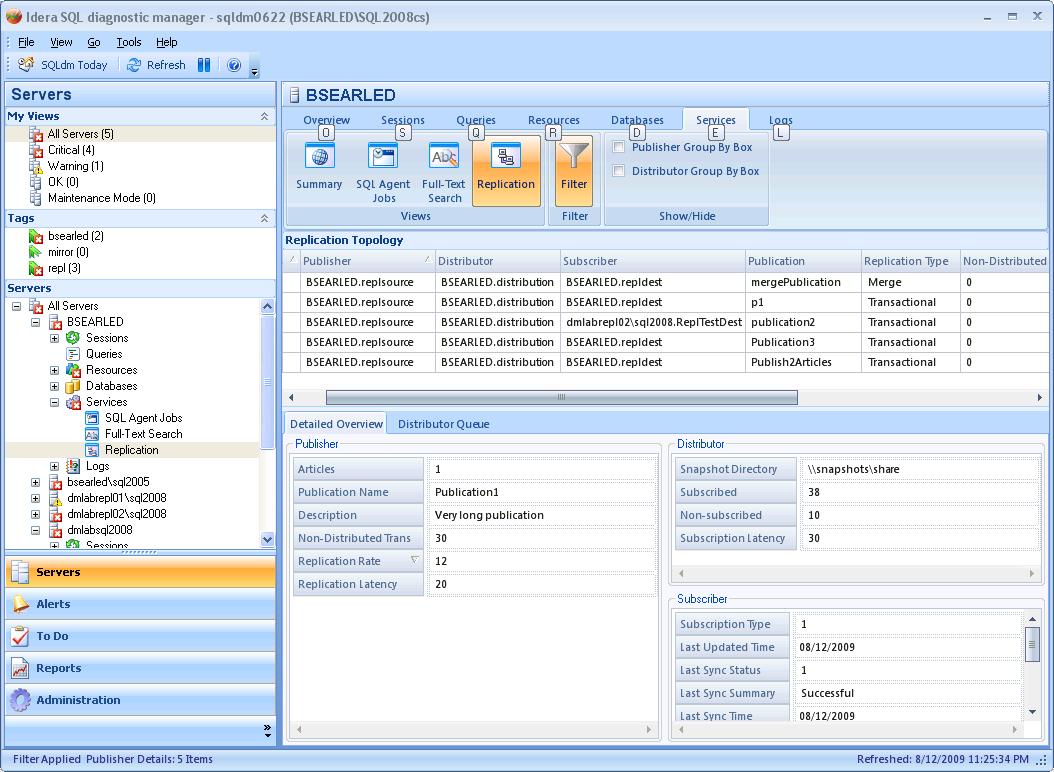
### Details Tab

There are 2 tab pages, the first containing a detailed overview of transactional replication and merge replication and the second page for the distribution queue.

* Details of Transactional replication
* Overview
* Publisher Frame on the detailed overview tab
  + - Number of replicated articles in this publication.
    - Publication Name
    - Publication Description
    - Non-distributed transactions (marked for distribution in the publishers log) \* alert
    - Replication Rate (rate at which distribution database is taking transactions)
    - Replication Latency (lag of publisher behind distributor) \*alert
    - Distributor frame on the detailed overview tab
    - Snapshot directory
    - Subscribed - from distributor (number of commands that have been successfully subscribed).
    - Non-Subscribed - from distributor (in distributor bit not subscribed) \*alert
    - Subscription Latency- from distributor (lag of subscriber behind distributor) \*alert
    - Approximate time till synchronized. This might prove to be meaningless.
* Subscriber frame on the detailed overview tab
* Subscription Type {push, pull, anonymous}
* Last Updated Time (for this publication)
* Last Sync Status
* Last Sync Summary
* Last Sync Time
* Details of merge replication
* Publisher
* Subscriber
* Distributor
* Publication
* Agent Name
* Last action description
* Action time
* Start time
* Duration – of current sync if synchronizing otherwise total run time of agent.
* Delivery Rate
* Publisher
* Insert count
* update count
* delete count
* conflicts
* Subscriber
* insert count
* update count
* delete count
* conflicts
* Distributor Queue
  + - Queue contents
    - Entry Date
    - Subscription Database (there can be multiple of these per distributor)
    - Wait Time
    - Command

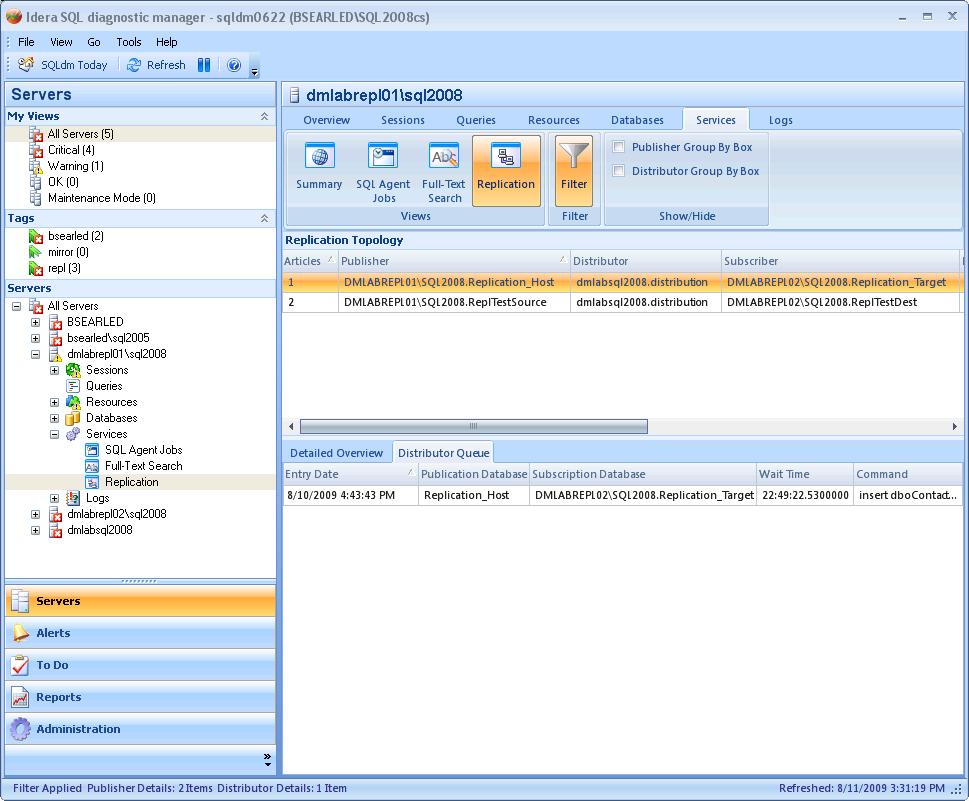
### Screen Shots of view layout

#### Detailed Overview



* Approximate time till synchronized will be in the distributor frame.
* The Filter button will be removed.
* The Publisher group by box will become the Topology group by box.
* The frst tab will display merge details if a mere session is selected.
* Show graphs of alert metrics on the detailed overview tab.
* Non-distributed queue
* Non-distributed latency
* Non-subscribed queue
* Non-subscribed latency
* When replication has been disabled show a warning “Replication has been disabled”. The probe will not try and gather data from a server that has disabled replication.

#### Distribution Queue



### New Distribution Latency Alert

We did not previously alert for high distribution latency but it is as important as subscription latency.

Measured in seconds, this alert has the same default thresholds and the existing unsubscribed transactions alert.

OK – 0 to 30 seconds

Warning - 30 to 60 seconds

Critical – 60 to xxx seconds. xxx is the max that we allow on other alerts.

Alert suppression allows the user to set up a suppression window wherein this alert will not be raised.

## Installation and Upgrade

Ww will need to add the creation of a number of objects to the database creation script.

* p\_UpdateReplicationTopology is the stored procedure that will save the topology to a table.
* p\_UpdateReplicationTopologySubscriber – Updates subscriber specific data
* p\_UpdateReplicationTopologyDistributor – Updates distributor specific data
* GroomReplicationTopology
* ReplicationTopology table in the repository.

## Permissions and other Required Configuration

This view is entirely read-only and as such, there are no special privileges required to access the functionality.

## Licensing Issues

## Dependencies

### Dependencies on Idera Software

### Third-Party Software Required on the Customer Machine

### Third-Party Software Required Internally

# Internal Design

## Architecture

* Multiple distributors can exist on one server.
* A publisher instance can have only one distributor.
* A subscriber can subscribe to many distributors.
* The accessible key information on all participants is the publisher instance, publisher database and the publication name.

### Scheduled refresh

Issues

* If the distributor is not on the publisher we do not run the sql to get the subscription latency alert.
* The query that gives us the non-subscribed queue length is incorrectly joined such that the queue length is lengthened by a factor of the number of published articles.
* We do not save any topology data to assist the view in knowing which servers are replication participants so it shows nothing when viewing the subscriber and the distributor (if the distributor is not on the publisher). By saving the data the view can ascertain which servers to connect to.

We have three objects that are populated during the scheduled refresh depending on what roles the machine is fulfilling.

* Published Database
  + Number of articles – dist and pub
  + Instance – dist and pub
  + db name – dist and pub
  + subscriber instance – dist and pub
  + subscriber db – dist and pub
  + Status – pub - tinyint
  + Sync\_type – pub - tinyint
  + Subscription\_type – pub - int
  + Replicated\_trans (non-distributed) - pub
  + Replicated trans\sec - pub
  + Replication latency (sec) - pub
  + Distributor Instance - pub
  + Distributor DB – pub
  + publication name - dist
  + publication description - dist
  + subscribed transactions - dist
  + non-subscribed transactions - dist
  + Sqldm database id
* Distributor – details gathered from publisher by running sp\_helpdistributor
  + Instance
  + DB
  + Directory
  + Account
  + Min distribution retention
  + Max distribution retention
  + History retention
* Distributor – details gathered from distributor
  + Max subscription latency
* subscriberDBs - sp\_MSenumsubscriptions
  + publisher
  + publisherdb
  + subscriber
  + subscriber db
  + publication name
  + replication\_type
  + subscription type
  + last\_updated
  + last\_sync\_status
  + last\_sync\_summary
  + last\_sync\_time

#### Scheduled Refresh Probe

The ReplicationCheck script that was previously used by the replication collector will be modified to iterate through publisher, distributor and subscriber databases on the server and return the required data.

### Saving the replication data

The non-distributed queue, subscribed trans and non-susbscribed queues must be saved to the repository.

Replication data should be saved so that the new replication view can populate with data from our repository, showing the topology of the monitored replication environment.

p\_UpdateReplicationTopology is the stored procedure that will save the topology to a table called ReplicationTopology in the repository.

ReplicationTopology Table

|  |  |  |
| --- | --- | --- |
| Column Name | Type | Source |
| LastSnapshotUpdate | DateTime | Snapshot update time |
| Publisher Instance | Nvarchar(128) | Publisher.syssubscriptions |
| PublisherDB | Nvarchar(128) | Publisher.syssubscriptions |
| publisherDBID | Int | sqldmRepository.p\_InsertDatabaseName |
| Distributor Instance | Nvarchar(128) | Publisher.sp\_helpdistributor |
| DistributorDb | Nvarchar(128) | Publisher.sp\_helpdistributor |
| distributorDBID | Int | sqldmRepository.p\_InsertDatabaseName |
| Subscriber Instance | Nvarchar(128) | Publisher.syssubscriptions |
| SubscriberDB | Nvarchar(128) | Publisher.syssubscriptions |
| subscriberDBID | Int | sqldmRepository.p\_InsertDatabaseName |
| Subscribed transactions | Long | Distributor.MSdistribution\_status |
| Non-subscribed transactions | Long | Distributor.MSdistribution\_status |
| Non-distributed transaction | Long | Publisher.Msrepl\_counters |
| Replication latency (secs) | Double | Publisher.Ms\_repl\_counters |
| Max Subscription Latency (secs) | Long | Distributor.MSrepl\_transactions |
| Replication Type | Byte {transactional, snapshot, merge} | Subscriber.MSenumsubscriptions |
| Subscription Type | Byte {push, pull, anonymous} | Subscriber.MSenumsubscriptions |
| Last Updated – the time of the last update of the subscriber by the distribution db. | Datetime | Subscriber.MSenumsubscriptions get it from MSreplication\_subscriptions |
| last\_sync\_status {\*} | Int | Subscriber.MSenumsubscriptions |
| last\_sync\_summary | Nvarchar(128) | Subscriber.MSenumsubscriptions |
| last\_sync\_time | Datetime | Subscriber.MSenumsubscriptions |
| subscriptionStatus | Byte {Inactive, subscribed, active} | Publisher.syssubscriptions |
| Publication | Nvarchar(128) | Distributor.MSpublications |
| Publication Description | Nvarchar(255) | Distributor.MSpublications |
| Number of articles | Int | Distributor.MSSubscriptions |

last\_sync\_status **0** = All jobs are waiting to start

**1** = One or more jobs are starting

**2** = All jobs have executed successfully

**3** = At least one job is executing

**4** = All jobs are scheduled and idle

**5** = At least one job is attempting to execute after a previous failure

**6** = At least one job has failed to execute successfully

Last\_sync\_time

Time the subscription information was updated. This is a UNICODE string of ISO date (114) + ODBC time (121). The format is yyyymmdd hh:mi:sss.mmm where 'yyyy' is year, 'mm' is month, 'dd' is day, 'hh' is hour, 'mi' is minute, 'sss' is seconds, and 'mmm' is milliseconds

#### Grooming

Delete the replication topology data after a scheduled refresh if we find that replication has been deleted. In the table have a bit for “Session not found” for each of the participants. If the bit is set for all 3 participants then delete the session.

As a fallback, for servers that were removed we have the last updated time. This table needs to be groomed of all sessions with participants that have not been updated in the last x time. X is the same period as we use on our history data.

### The Replication View

Issues with the existing view:

* Does not show anything for subscriber and for distributors that are not on the publisher.
* Shows only the distribution queue using msrepl\_commands (which is dangerous) and only shows data when viewing a publisher.
* Does not show any of the key metrics such as:
  + Max Subscription Latency – time that a transaction has been distributed but not been subscribed for.
  + Max Distribution Latency – Time a transaction has been marked for distribution but is still not distributed.
  + Non-subscribed queue length.
  + Non-distributed queue length

#### Probes

##### Distributor Details

##### PublisherDetails

##### Subscriber Details

## Installation Issues

* New ReplicationTopology table needs to be added to the repository creation script.
* New UpdateReplicationTopology Stored Procedure needs to be added to the script.
* New GroomReplicationTopology stored procedure needs to be added.

## Schedule

### Work Breakdown and Sizings

Insert areas that put the implementation or schedule at risk – Assumptions made in design, new areas that have a learning curve etc. Break the project into logical components and add time for design, coding, integration, unit testing. These estimates should not take into account outside forces like maintenance work. These will be applied in building the overall project schedule. A general rule of thumb is to break any unit of work greater then 1 week into smaller pieces to ensure accuracy and to allow measurement of progress during the project.

|  |  |  |
| --- | --- | --- |
| Component | Who | Sizing |
| R&D | Barry | 2 weeks |
| Code Piece #1 |  |  |
| Code Piece #2 |  |  |
| Porting |  |  |
| Unit and Integration Testing |  |  |
| **Total** |  | **2 weeks** |

### Areas of Risk

Complex but somewhat loosely bound nature of the relationships between the replication participants could pose some risk.

Replication participants share no common GUIDs so all joins are on strings.

# Quality Assurance Considerations

### Overview

* The 3 existing alerts need to function as they did before.
* The new replication latency alert needs to work the same as subscription latency.
* The view should never get blocked as it did previously.

### Developer-Created Unit Tests

Automated, Developer-Created Unit Tests which will be Created (and Passed to QA)

Developer Tests which will be Manually Performed

# Documentation Considerations

# Bibliography