

Node Intersection Logic

Kentucky Transportation Cabinet

Functional Specification

**Version 0.01**

**Document Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Update Source** | **Date** | **Description** |
| 0.02 | Joe Mendoza | 14JUL2014 | Initial draft |
|  |  |  |  |
| **For review** | | | **For approval** |
| KYTC | | |  |
| Bentley project team | | |  |

Table of Contents

[1 About This Document 4](#_Toc393096631)

[1.1 Document Purpose 4](#_Toc393096632)

[1.2 Document Terminology 4](#_Toc393096633)

[1.2.1 Abbreviations 4](#_Toc393096634)

[1.2.2 Terms and Concepts 4](#_Toc393096635)

[1.3 Related Documents 4](#_Toc393096636)

[2 Solution Overview 5](#_Toc393096637)

[2.1 Solution Objectives 5](#_Toc393096638)

[2.1.1 Intersection Identifiers 5](#_Toc393096639)

[2.1.2 Spatial Manager Execute procedure 5](#_Toc393096640)

[3 Items not addressed 8](#_Toc393096641)

[3.1 8](#_Toc393096642)

# About This Document

## Document Purpose

This document describes the systems features and behavior allowing users to verify that the system will meet their needs while also containing sufficient detail for the subsequent design.

This document describes WHAT the system should do; it makes no attempt to describe HOW this should be implemented.

Operating constraints and assumptions that may affect the final solution will also be defined.

## Document Terminology

The following abbreviations, terms and concepts are used in the document:

### Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Meaning, Definition |
| BSM | Bentley Solutions Methodology |
| KYTC | Kentucky Transportation Cabinet |
|  |  |

### Terms and Concepts

|  |  |
| --- | --- |
| Term, Concept | Meaning, Definition |
| System | Physical (hardware) and logical (software) environment, required for the solution to operate, including the solution itself |

## Related Documents

Following is the list of documents that this document refers to or that provide with additional information about this topic.

|  |  |
| --- | --- |
| # | Document, Description, Version |
|  |  |
|  |  |

# Solution Overview

## Solution Objectives

The solution can be broken into two parts: The Intersection Identifiers in the Node Usage Table and the Spatial Manager Execute procedure

### Intersection Identifiers

The purpose of this item is to be able to Identify via a reporting tool which nodes belong to a given intersection. This will be done with the use of the no\_purpose field in the nm\_nodes\_all table.

Initially this field is left null by the user as an indication that an intersection ID has not been assigned.

To manually assign a new intersection the user would type in ‘I:’ into the field this would prompt a trigger to retrieve and assign the next intersection ID.

Note: I left it as I: to minimize the typing that would have to be done but the user, however I am happy to expand that to anything else i.e. ‘INTERSECTION:’ or ‘INTERSECTION ID:’, I would like to keep the colon to make a hard character to parse off, of although since the value is fixed width, it might not be necessary.

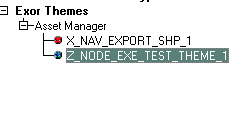
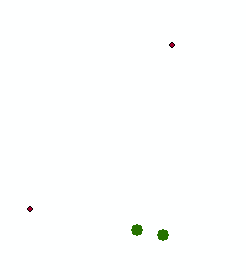
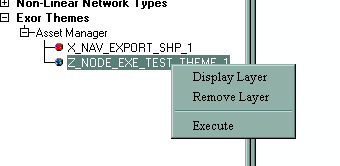
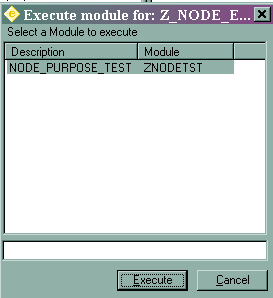
To manually assign a node to an existing intersection the user would obtain an existing intersection number let use 12345 for example and type I:[Intersection ID] so the example would be: I:12345

### Spatial Manager Execute procedure

In order for KYTC to easily be able to group and assign Intersection values in Spatial Manger an execute produce well be assigned to an Exor Theme.

The work Flow would be similar to:

The general work flow for this would be:

1. Turn on the node layer with the procedure attached
   1. In this case z\_node\_exe\_test\_theme\_1
      1. 
2. Select the Intersection Nodes Using the select Exor Features Tool
   1. 
3. On the Exor Groups tab right click on the Theme and select Execute
   1. 
4. A window will pop up allowing the user to pick a procedure
   1. 
5. Once Execute is ran, the procedure will be ran on the selected Nodes.

When the procedure is executed a few things could happen and KYTC would have to tell us which items they would like.

* Overwrite existing values
  + With this the procedure would take all the selected nodes and assign a new intersection ID regardless what the current value is. While this is the easiest to create, it is also the most dangerous, if the user accidentally selected nodes that were already assigned they would be overwriting without a rollback (unless we wanted to add a history table and increase overhead.)
* Only assign new ID to NULL no\_purpose fields
  + The procedure would assign a new ID if all of the no\_purpose fields were null, otherwise it would throw an error.
* Copy current node ID
  + If a node ID already exists on one node or the same ID exists on several nodes, then copy that ID to nodes that have a null no\_purpose otherwise throw an error
* Copy current node ID or assign new
  + This would pick the appropriate action based on the previous 2 items.

# Items not addressed

## 

This document currently does not address:

* Any Reporting Views
* How to load the initial Data from The University of Kentucky.