|  |  |
| --- | --- |
| BentleyLOGO_4C_no-tag | flyingTbanner |

***Oregon Department of Transportation***

**TransInfo Signs Asset Data Integration**

**Sign Data Access Data Functional Specification**

***Version 04.0***

***Bentley Systems 2014***

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| *Date* | *Version* | *Changed by* | *Notes* |
| November 2014 | 0.01 | JD |  |
| Dec 23, 2014 | 0.02 | JD | Sync with CD Dec 22 Draft |
| March 2, 20015 |  |  | Change to unit requirement format |

Note:

* Version numbers < 1 indicate draft
* Whole numbers indicate a finalized edition
* Version numbers between whole numbers indicate error correction, minor changes altering the substance of the report
* Version numbers with 2 decimal points indicate draft to a minor or major revision.
* Draft changes prior to a major revision should be marked as “DRAFT”
* Version does not indicate acceptance by ODOT or Bentley Systems.

Contents

[1 Synchronization Process Flow 6](#_Toc415060478)

[2 Summary of Sync Processes 6](#_Toc415060479)

[3 Sync Process Definitions 8](#_Toc415060480)

[1. Launch Sync Process 8](#_Toc415060481)

[1.1. Pre-Data Sync Process 8](#_Toc415060482)

[1.2. Post-Data Sync Process 9](#_Toc415060483)

[2. Test Connectivity 10](#_Toc415060484)

[3. Establish Sync Task Listing 11](#_Toc415060485)

[4. SFA to TI Sync 13](#_Toc415060486)

[4.1. Sync New or Deleted Installation in SFA to TI 13](#_Toc415060487)

[4.2. Sync Updated Installations in SFA to TI 16](#_Toc415060488)

[4.3. Sync Updated Supports 18](#_Toc415060489)

[4.4. Sync Maintenance Log Asset 20](#_Toc415060490)

[4.5. Sync Standard Sign Assets 23](#_Toc415060491)

[4.6. Sync Custom Sign and Legend Assets 24](#_Toc415060492)

[4.7. Notification of Sync Progress 25](#_Toc415060493)

[5. Insert/Updated LOV data in SFA 26](#_Toc415060494)

[5.1. Sync Shared Domains/LOV’s 26](#_Toc415060495)

[5.2. TI Un-located Asset LOV Sync 29](#_Toc415060496)

[6. TI to SFA Sync 30](#_Toc415060497)

[6.1. TI to SFA Installation Updated and End-date Sync 30](#_Toc415060498)

[6.2. TI to TI to SFA Maintenance History Update and End-date Sync 31](#_Toc415060499)

[6.3. TI to SFA Support Update and End-date Sync 32](#_Toc415060500)

[6.4. TI to SFA SIGN (Standard and Custom) Update and End-date 33](#_Toc415060501)

[7. Sync Status 34](#_Toc415060502)

[7.1. Create Exception Table in Oracle 34](#_Toc415060503)

[7.2. SFA to TI Sync Exceptions 35](#_Toc415060504)

[7.3. TI to SFA Sync Exceptions 36](#_Toc415060505)

[7.4. Create Sync Status Table in Oracle 37](#_Toc415060506)

[7.5. Sync Status Update 37](#_Toc415060507)

[8. Initial Data Conversion 38](#_Toc415060508)

[8.1. Initial Data Conversion 38](#_Toc415060509)

[9. Data Transformation Requirements 40](#_Toc415060510)

[9.1. Initial Data Conversion 40](#_Toc415060511)

[10. Client (Toughbook) Configuration 41](#_Toc415060512)

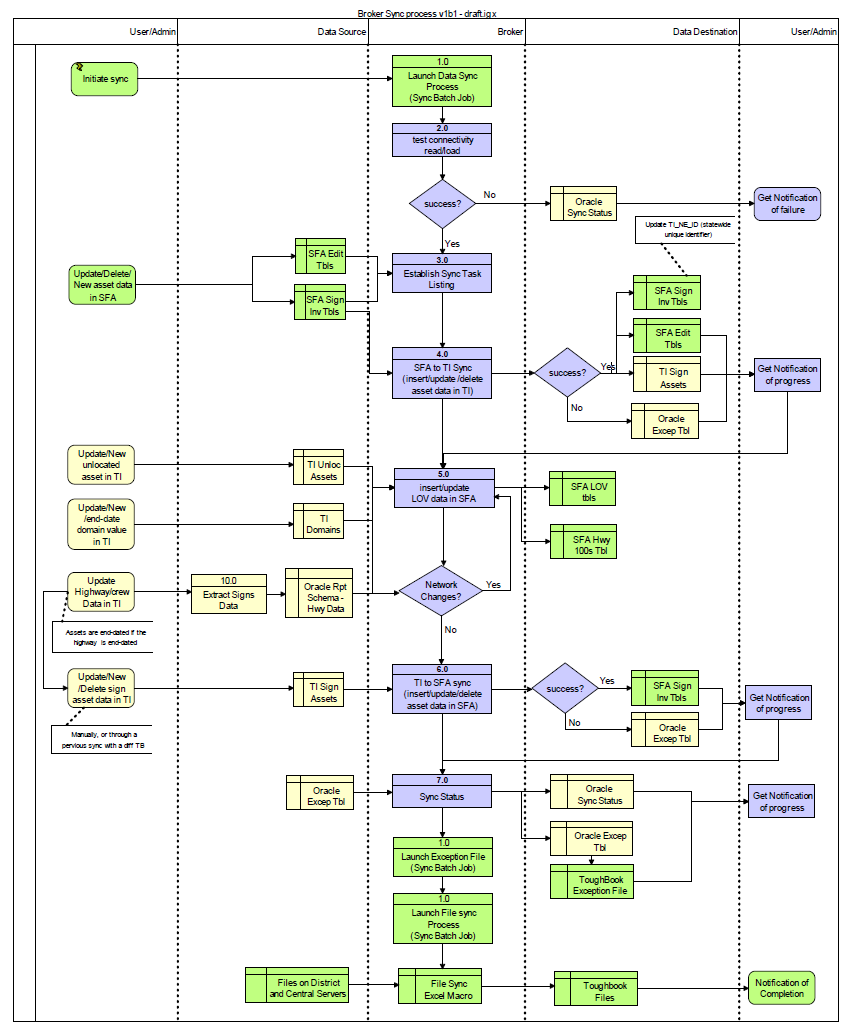
[4 Appendix 42](#_Toc415060513)

[4.1 SFA tblInstallations.fldSide to TI Installation (SNIN) XSP Transformation Crosswalk 42](#_Toc415060514)

[4.2 Sync Process Logical Data Flow Diagram 43](#_Toc415060515)

[4.3 TI to SFA, New Custom Sign Sync Diagram 44](#_Toc415060516)

# Synchronization Process Flow



# Summary of Sync Processes

1. **Launch Data Sync Process – Batch job**
   1. Pre-DataSync Process
      * Sync Broker version with central server (xcopy) and launch
   2. Post-DataSync Process
      * Launch the exception file on the toughbook
      * Launch file sync process
2. **Test Connectivity**
   * + Validate Access version is as expected? (maybe put the version in Oracle and launch an error)
     + Notification of Success/Failure
3. **Establish Sync Task Listing**
   1. Identify modified sign asset data from Access and TI
   2. Identify LOV modifications
   3. Identify Highway 100s modifications
4. **Sync Insert/update/delete assets data in TI**
   1. Sync new or deleted installation in Access to TI
   2. Sync updated installation in Access
   3. Sync new or updated\* support asset in Access and Sync to TI
   4. Sync new or updated\* maintenance log asset in Access and Sync to TI
   5. Sync new or updated standard sign panels
   6. Sync new or updated custom sign panels and legends
   7. Notification of Sync Progress
5. **Insert/Update LOV Date in SFA**
   1. Update shared domain/LOV’s Sync
   2. TI Un-located Asset LOV Sync
6. **TI to SFA Sync**
   1. TI to SFA Installation Update and End-date Sync
   2. TI to SFA Maintenance History Update and End-date Sync
   3. TI to SFA Support Update and End-date Sync
   4. TI to SFA SIGN (Standard and Custom) Update and End-date I to SFA
7. **Exceptions** 
   1. Create Exception Table in Oracle
   2. SFA to TI Sync Exceptions
   3. TI to SFA Sync Exceptions
8. **Sync Status**
   1. Create Sync Status Table in Oracle
   2. Sync Status Update
9. Initial Data Conversion
   1. Initial Data Conversion
10. **Extract Signs Data**
    1. Extract Monthly Highway Data
    2. Extract Signs Data

# Sync Process Definitions

1. Launch Sync Process
   1. Pre-Data Sync Process

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Sync Batch Job* |
| **Purpose** (Why does this task exist; what value does it add?) | *Checks that the Broker.exe on the Toughbook matches the version on the ODOT server. If not, it uses Xcopy to copy it over* |
| **Triggers** (What kicks off this task?) | *Field Maintenance launces the synchronization process on the Toughbook* |
| **Dependencies** (What has to exist before this Task can start?) | *Broker and Excel macro located on central server. Broker exe shortcut on the Toughbook to launch the synchronization process. Un/pw configured on the TB*  *Updated to Broker via Xcopy with no manual intervention on the laptop* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES** |
| 1. The field maintenance personnel launches Broker Sync re Broker UI | Manual | Check server or Oracle? |
| 1. Execute an Xcopy command that checks the date of the Toughbook Broker.exe with the date on the server Broker.exe 2. Xcopy Excel Macro if new version – check date 3. Launch Signs Broker 4. The current Broker and Access version will be stored in SignClient.tblVersionControl. The broker will compare the “major” number (see notes). If they are not equal the sync will be stopped and the user will be notified to contact an administrator. 5. Broker minor bug fixes and refinements will be tracked/accounted for in a build number and not in the version number 6. Notify user if Xcopy fails   \*Placeholder\* if the access upgrade is automate the process may occur during his task, | Sync Batch Job | Only copy the file over if it is out of date  Encrypted un/pw for oracle in xml file on each machine  The major version number is indicated in red below:  SFA Broker  TI.1.0.0 BV.1.0 |
|  |  |  |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| None |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| Broker version is out of date | *New version is downloaded and launches* | *Is user notification required* |

* 1. Post-Data Sync Process

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *At the full completion of data sync process open the exception file for the field personal to see on the Toughbook* |
| **Triggers** (What kicks off this task?) | *Successful completion of the broker/data sync process* |
| **Dependencies** (What has to exist before this Task can start?) | *Excel macro installed on TB* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES** |
| 1. Launch the Excel Macro that synchronizes files between the Toughbook and the servers 2. Open Exception file in notepad 3. Sync complete | Sync Batch Job | In the event of Excel Macro will the Excel macro issue and error – can the error be captured – Joe / Jason  <Exception filename> |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| None |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Successful TB-Server file sync 2. Manual counts - validate 3. Forced failed TB –Server Sync 4. Report Failed TB/S sync 5. Open Exception file |  | *Log file solution – count files on TB compare to server – broker incorporates into sync log* |

1. Test Connectivity

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *Verify the broker can connect to the network, to Oracle and to Access* |
| **Triggers** (What kicks off this task?) | *Broker launched* |
| **Dependencies** (What has to exist before this Task can start?) | 1. Broker has been configured with the proper username/password ? 2. No access pw required. 3. The Oracle user that the broker will use has been setup in Oracle and given the role: Broker\_Role - contact James McDonald 503 986 4499 |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. The Broker is launched | Sync Batch Job |  |
| 1. Validate Access version on TB matches the version in the Access DB – Process to be determined 2. Success    1. Continue 3. Failure    1. Stop sync and display error: “You have an outdated version of the Signs Access Database, please contact your administrator for help” | Broker | Only copy the file over if it is out of date?  Or pushed to TB by admin (Jason)  Broker.exe |
| 1. Connect to the Access Database on the Toughbook 2. Success    1. Continue 3. Failure    1. Stop sync and display error: “Sync process unable to connect to the Signs Access Database, please contact your administrator for help” | Broker |  |
| 1. Log into the Oracle Database 2. Success    1. Continue 3. Failure 4. Stop sync and display error: “Sync process unable to connect to the TI Database, please contact your administrator for help” | Broker | Pull credentials from hidden xml file |
| 1. Provide progress notification to the user showing success | Broker | Message: db connection successful? |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| NA |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| Launch Broker, Successful connection |  |  |
| Duplicated possible failure scenario | *Document possible failure reasons*   1. *Access is already open* 2. *Oracle is down* 3. *Invalid Oracle un/pw* 4. *Invalid or no broker role* 5. *Network is down* | *Set broker to single instance? – what does this mean* |
| Test error messages |  |  |

1. Establish Sync Task Listing

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *Identify data changes Installations, panels, supports, and maintenance history in SFA and TI* |
| **Triggers** (What kicks off this task?) | *Follows successful completion of connectivity tests* |
| **Dependencies** (What has to exist before this Task can start?) | *Un- synced SFA database, un synced TI Sign Assets* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** | |
| 1. Broker reads SFA asset edit tables, compiled list of unique update events 2. Broker reads TI NM\_INV\_ITEMS\_ALL and NM\_MEMBERS\_ALL establishes listing of update sign assets in TI. 3. The Broker filters TI data based to district/crew as defined in signdata.tbl.DfltDist. The broker will read the tblDistinctLkup.fldCrewOnlyflag. If check the filter will be based on the crew, if unchecked it will be based on the District 4. Broker evaluates the SFA listing and TI listing and identifies conflicts. 5. Broker reads NM\_INV\_DOMAINS\_ALL.DATE\_MODIFIED for SFA domains modification since last sync, 6. Broker reads TIODS\_RPT\_MP\_HUND, TI\_RPT\_MP\_HUND\_PRLIM,TIODS\_MAINT\_CREW\_HIER,TIODS\_RTE\_HWY\_LOC. the Broker filters this data based on district/crew as defined in signdata.tbl.DfltDist. The broker will read the tblDistinctLkup.fldCrewOnlyflag. If check the filter will be based on the crew, if unchecked it will be based on the District 7. Broker generates sign asset task listing. Compiles update asset counts to be used in UI progress feedback and sync log | There can be multiple – duplicate - records for a single installation in the SFA edit tables. The broker will retrieved the first unique installation id instance, skip the duplicates, and retrieve the next unique instance. Installation deletions will be ordered to be process first, all other instance of a deleted installation will be ignored.  If Installation is end-data in TI, no need to list end-dated child assets  Change: We cannot deal with the addition of the same sign from SFA and TI during the same sync cycle.  Change: Need to create a TI view for signs at the same location?  Not | |
|  |  |  |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table Names** |
| tblEditInstallation  ExpNewINstall  tblEditSupports  tblEditStandard  tblEditCustom  tblEditLegend  tblEditHistory |  |  | *ExpNewInstall lists new installations*  *Jason: Need SNIN\_NE\_ID in the tblEditInstallation table And tblInstallations* |

| **TI DATA** | **Data Type** | **Rules/Notes** |
| --- | --- | --- |
| **Table Names** |
| NM\_INV\_ITEMS\_ALL NM\_MEMBERS\_ALL | *Asset* | *Last mod date vs last sync date* |
| NM\_INV\_DOMAINS\_ALL.DATE\_MODIFIED | *Domains:*  *SIGN\_SHEETING*  *SIGN\_SBSTR*  *GEN\_DIR (need for new setup)* |  |
| TIODS\_MP\_HUND |  |  |
| TIODS\_MAINT\_CREW\_HIER |  |  |
| TIODS\_RTE\_HWY\_LOC\_ |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Utilizing a SFA db with all edit tables populated, perform asset validation tests – count validation 2. Identify and Duplicate failures points |  | *Requires at least 10 pre – synced SFA db’s – validate that ever possible delete/update/add is represented –*  *Compile listing of all possible edit actions* |

1. SFA to TI Sync
   1. Sync New or Deleted Installation in SFA to TI

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *Transformation of Sign data collected with the SFA into TransInfo – the ODOT enterprise asset repository* |
| **Triggers** (What kicks off this task?) | *Completion of 4.0 task* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0.* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing 2. If New:    1. Pull target installation data record from tblInstallations asset inventory data specified in task listing    2. Pass data to TI API    3. Transform side of road to XSP – based direction of travel    4. Validate fldDirection against highway route extract – throw exception if invalid    5. API validates and insert data in to NM\_INV\_ITEMS\_ALL (as SNIN asset)and NM\_MEMBERS\_ALL    6. Write SNIN\_NE\_ID back to access    7. If successful remove target record from tblEditInstallation , add children as per unit tasks below    8. If error or validation failure write data (as described in 490 section xXX) to exception table, leave records in associated edit tables, do not add child assets.    9. Proceed to next action listing 3. If Deleted:    1. pass instructions to API to end-date the target installation and associated child assets    2. If successful or doesn’t exist remove target record from tblEditInstallation    3. If there is a write lock error the record is not removed from the tblEditInstallation table |  | A new installation and it’s children are added at the same time  Us District/Crew filter for Sign asset data update on SFA  We cannot detect duplicated new installations added in the office and in the field during the same sync cycle. Exception report needed   1. There are signs located at the same milepoint and the same XSP (cannot be exclusive by XSP) – The 2. Bentley dev environment will need to be changed so that SNIN is not exclusive by XSP 3. ~~There are signs located to the thousandths –check odot exor config – is precision .00 or .00~~   tblInstallations.fldSide to XSP Tranfomations will be performed by the broker see Section 2.1  location, Direction, Side (XSP), distance, lat, long, additional notes – on change we end-date and recreate to keep history |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| tblInstallations |  |  | Jason – will check for valid mp when adding or subtracting mileage for sequencing  Jason: will validate fldDirection against (general road direction)–  Highway, prefix, mp, rdwy\_id (converted to I and D in Access) – LRS data that is the location of the asset, history kept by default  create date (start date populates it from TI to Access (if blank), |
| tblEditInstallation |  |  |  |
| ExpNewINstall |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| NM\_INV\_ITEMS\_ALL |  |  | XSIGN\_EXECPT needs to be created in ODOT dev env  Note to CD: we will send you the creation script |
| NM\_MEMBERS\_ALL |  |  |  |
| XSIGN\_EXECPT |  |  |  |

|  |  |  |
| --- | --- | --- |
| Data Transformation Rules | | |
| TI | SFA | Rule |
| XSP | Installation. | See xsp doc |
| Startdate | fldCreationDate | Ignore creation data set startdate for a new installation to sync date |
|  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validate single and multiple new installations in single sync operation 2. Validate single and multiple installation deletions/end-date in single sync operation 3. Validate Test multiple new and deleted installation from different TBs 4. Validate the same new or deleted installation on two different TB’s from the same district | **For validation use asset Items form (NM0510)** | **Is it possible to have the same new or deleted installation on two different TB’s**  **Installations are not exclusive**  **If an installation is edited twice in the same sync cycle – write exception**  location, Direction, Side (XSP), distance, lat, long, additional notes – on change we end-date and recreate to keep history |
|  |  |  |

* 1. Sync Updated Installations in SFA to TI

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *Transformation of Sign data collected with the SFA into TransInfo – the ODOT enterprise asset repository* |
| **Triggers** (What kicks off this task?) | *Completion of 4.0 task* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0.* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing 2. Pull target installation data record from tblInstallations asset inventory data specified in task listing 3. Pass data to TI API 4. Evaluate data: if change in **Location, Direction, Side (XSP**), distance, lat, long, additional notes – end-date and recreate to keep history. 5. Update – the SNIN\_NE\_ID   If change in location, just update location set location date to date of sync   1. API validates and insert data in to NM\_INV\_ITEMS\_ALL (as SNIN asset)and NM\_MEMBERS\_ALL 2. If error or validation failure write data (as described in 490 section xXX) to exception table, leave records in associated edit tables, do not add child assets. 3. Proceed to next action listing 4. Evaluate Child asset edit tables, rebuild or process as required 5. Proceed to next action listing |  | Reject bad mp values.  ~~Check if XSP retains history like MP – no it doesnt~~  **If an installation is edited twice in the same day write to oracle sync log do not remove from edit table let it sync next day – write exception**  **One update a day** |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| tblInstallations |  |  |  |
| tblEditInstallation |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| NM\_INV\_ITEMS\_ALL |  |  | Assign start date/location data as sync date |
| NM\_MEMBERS\_ALL |  |  |  |
| XSIGN\_EXECPT |  |  |  |

|  |  |  |
| --- | --- | --- |
| Data Transformation Rules | | |
| TI | SFA | Rule |
| XSP | Installation. | See xsp doc |
| Startdate | fldCreationDate | Ignore creation data set startdate for a new installation to sync date |
|  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validate single and multiple updated installations in single sync operation 2. Validate end-date update w/children 3. Validate location update 4. Validate the update of same installation in same sync, and creation of exception listing 5. Test for updated SNIN\_NE\_ID when end-date 6. Check for 2 or more updates to the same asset in a day |  |  |
|  |  |  |

* 1. Sync Updated Supports

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *Transformation of Sign data collected with the SFA into TransInfo – the ODOT enterprise asset repository* |
| **Triggers** (What kicks off this task?) | *Completion of 5.1 – 5.2 Sync Installation tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0.* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing, isolate target support installation id 2. Pull target support data record(s) from tblSupports asset inventory data 3. Pass data to TI API 4. API end-dates target installation support records (asset SNSU)–if preexisting, validates and inserts new support data records (as SNSU asset into NM\_INV\_ITEMS\_ALL and NM\_MEMBERS\_ALL 5. API builds child relationship in NM\_INV\_TYPE\_GROUPING 6. If successful remove target record from tblEditSupport 7. If error or validation failure write data (as described in 490 section xXX) to exception table 8. Proceed to next action listing |  | It’s the developers prerogative to either process all assets per installation at a time or one asset at a time |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| tblInstalledSupports |  |  |  |
| tblEditSupports |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| NM\_INV\_ITEMS\_ALL |  |  |  |
| NM\_MEMBERS\_ALL |  |  |  |
| NM\_INV\_TYPE\_GROUPING |  |  |  |
| XSIGN\_EXECPT |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validate single and multiple new and updated supports in single sync operation 2. Validate multiple new and updated supports from different TBs from the same district 3. Validate support update with a new intallation 4. Validate Installation/Support parent/child relationship | **For validation use asset Items (NM0510)** |  |

* 1. Sync Maintenance Log Asset

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *Transformation of Sign data collected with the SFA into TransInfo – the ODOT enterprise asset repository* |
| **Triggers** (What kicks off this task?) | *Completion of 5.1 – 5.2 Sync Installation tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0..* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing, isolate target Maintenance Log installation id. 2. Pull target maintenance log record(s) from tblInstallationHistory asset inventory data 3. Pass data to TI API 4. API delete update target maintenance log records –if preexisting, validate and inserts ne data records into NM\_INV\_ITEMS\_ALL and NM\_MEMBERS\_ALL 5. API builds child relationship in NM\_INV\_TYPE\_GROUPING 6. If successful remove target record from tblEditHistory 7. If error or validation failure write data (as described in 490 section xXX) to exception table 8. Proceed to next action listing |  | It’s the developers prerogative to either process all assets per installation at a time or one asset at a time |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** | |
| --- | --- | --- | --- | --- |
| **Table name** |
| tblInstallationHistory |  |  |  | |
| tblEditHistory |  |  |  | |
| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** | |
| **Table name** |
| NM\_INV\_ITEMS\_ALL |  |  |  | |
| NM\_MEMBERS\_ALL |  |  |  | |
| NM\_INV\_TYPE\_GROUPING |  |  |  | |
| XSIGN\_EXECPT |  |  |  | |
| **Test/Exception** | **Specific Instructions** | | | **Notes** |
| **Case** |
| 1. Validate single and multiple installation maintenance log updates in single sync operation 2. Validate multiple installation maintenance log updates from different TBs from the same district 3. Validate mainteacne history with new install – does this happen 4. Validate Installation/Maintenance Log parent/child relationship | **For validation use asset Items (NM0510)** | | |  |

* 1. Sync Standard Sign Assets

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *Transformation of Sign data collected with the SFA into TransInfo – the ODOT enterprise asset repository* |
| **Triggers** (What kicks off this task?) | *Completion of 5.1 – 5.2 Sync Installation tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0.* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing, isolate target standard sign installation id. 2. Pull target standard sign record(s) from tblInstallatedStandardSigns 3. Pass data to TI API 4. API deletes target sign records in SNSN –if preexisting, Set type = ‘S’, validate and inserts new standard sign data records into NM\_INV\_ITEMS\_ALL (as SNSN asset) NM\_MEMBERS\_ALL 5. API builds child relationship in NM\_INV\_TYPE\_GROUPING 6. If successful remove target record from tblEditHistory 7. If error or validation failure write data (as described in 490 section xXX) to exception table 8. Proceed to next action listing |  | It’s the developers prerogative to either process all assets per installation at a time or one asset at a time |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| tblInstalledStandardSigns |  |  |  |
| tblEditStandard |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| NM\_INV\_ITEMS\_ALL |  |  |  |
| NM\_MEMBERS\_ALL |  |  |  |
| NM\_INV\_TYPE\_GROUPING |  |  |  |
| XSIGN\_EXECPT |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validate single and multiple installation standard sign updates in single sync operation 2. Validate multiple installation standard sign from different TBs from the same district 3. Validate Installation/Standard Sign Log parent/child relationship 4. Validate insertion of standard sign with new install 5. Check multiple panels on the same install 6. Check end-dated rebuilt install with edits multiple panels | **For validation use asset Items form (NM0510)** |  |

* 1. Sync Custom Sign and Legend Assets

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *Transformation of Sign data collected with the SFA into TransInfo – the ODOT enterprise asset repository* |
| **Triggers** (What kicks off this task?) | *Completion of 5.1 – 5.2 Sync Installation tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0.* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing, isolate target custom sign installation id. 2. Pull target custom sign record(s) from tblInstallatedCoustomSigns 3. Pull target legends records from tblCustomSignLegends 4. Transform legend data for insertion into SNSN (Signs). COUSTOM\_LGND 5. Pass data to TI API 6. API deletes target sign records –if preexisting, set sign type = ‘C’; validate and inserts new Custom sign data records into NM\_INV\_ITEMS\_ALL (as SNSN asset) NM\_MEMBERS\_ALL 7. API builds child relationship in NM\_INV\_TYPE\_GROUPING 8. If successful remove target record from tblEditCustom 9. If error or validation failure write data (as described in 490 section xXX) to exception table 10. Proceed to next action listing |  | ~~CUSTOM\_LGND\_ID attribute in SNSN Asset will not be unique and therefore serves no purpose. – CD agreed~~ |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| tblInstallatedCustomSigns |  |  |  |
| tblEditCustom |  |  |  |
| tblCustomSignLegends |  |  |  |
| tblEditLegend |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| NM\_INV\_ITEMS\_ALL |  |  |  |
| NM\_MEMBERS\_ALL |  |  |  |
| NM\_INV\_TYPE\_GROUPING |  |  |  |
| XSIGN\_EXECPT |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validate single and multiple installation custom sign updates in single sync operation 2. Validate multiple installation custom sign from different TBs from the same district 3. Validate Installation/ custom sign parent/child relationship 4. Check multiple panels on the same install 5. Check end-dated rebuilt install with edits multiple panels | **For validation use asset Items form (NM0510)** |  |

* 1. Notification of Sync Progress

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *Informs user of the number of assets – lov’s involved in the sync event, and the progress as reflected in the count of synced assets as the sync process proceeds, to include exceptions as they occur.* |
| **Triggers** (What kicks off this task?) | *Task 3.0, as the sync proceeds 5.1-5.6* |
| **Dependencies** (What has to exist before this Task can start?) | *Action listing compiled in 3.0, completion of a synced asset or lov* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Read action listing, produce asset count from SFA by asset category, asset count from TI by asset category, LOV count, Hwy 100s count (number of routes) 2. Display count data in Broker UI 3. Update progress count in UI as sync process proceeds |  | . |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| tblEditInstallation  ExpNewINstall  tblEditSupports  tblEditStandard  tblEditCustom  tblEditLegend  tblEditHistory |  |  | *Add lov’s* |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table name** |
| NM\_INV\_ITEMS\_ALL  NM\_INV\_DOMAINS |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Establish object count manually 2. Intentionally end broker 3. Validate process count |  |  |

1. Insert/Updated LOV data in SFA
   1. Sync Shared Domains/LOV’s

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker/TI API* |
| **Purpose** (Why does this task exist; what value does it add?) | *To migrate domain values maintained in TI to SFA LOV tables* |
| **Triggers** (What kicks off this task?) | *Completion of 4.0 tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Updated TI domains:* SIGN\_SHEETING, SIGN\_SBSTR. Updated TI Reports: TI\_RPT\_MP\_HUND, TI\_RPT\_MAINT\_CREW\_HIER, TI\_RPT\_RTE\_HWY\_LOC |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Read action Listing    1. Pull target domain values    2. pass updated domain listing to broker for insertion into SFA LOV tables    3. In SFA delete values in target LOV table, repopulate with values from TI passed to broker in previous step.    4. In case of error? 2. If source Report/extract: API Evaluates target Report extracts for date modification sense last sync – move to task listing:    1. pass updated LOV data as specified in 490 section X.X LOV Extract Specification to broker for insertion into appropriate SFA LOV tables.    2. In SFA delete updated records in target LOV table, repopulate with updated/new values from TI passed to broker in previous step 3. tblHighwayEA , tblHighways , tblHWYLkup tblRoutesState, tblHwyGPSData will be filtered based on district/crew as defined in signdata.tbl.DfltDist. The broker will read the tblDistinctLkup.fldCrewOnlyflag. If check the filter will be based on the crew, if unchecked it will be based on the District | Use LRM key to id road  Put 100’s and 100’s prelim together  Maint Hierarchy – filter by crew type = ‘SNCW’  Any update for a given crew flush and replace for that crew.  For route tables: Select distinct I\_RTE  Distinct US\_RTE\_1 US\_RTE2  Distinct OR\_RTE\_1 OR\_RTE2  Select Distinct District  HWYLkup  LRM Key bit mapping  1-3 = Hwy #  4-5 = suffix  1-5 = SFA RD\_ID – strip 00 from suffix for mainline ?  6 = RDWY\_ID ( I or D)  7 – Mileage type  8 = Overlap code  7-8 = SFA Mileage type and SFA Mile post prefix  MileageType = 00,P0 or Z0-9 (last 2 char of the LRM)  If updated change all RD\_ID of updated recorded= (first 5 char LRM\_Key) and suffix  Test route mile post changes mod date for update purpose.  DistrictLkup, Broker will include the superset of highways under the crew/district if fldCrewOnlyFlag = F  It is added for district 7 where 7C & 7V crews will only receive the data for their crews and not the entire district.  Change: Dose DistrictLkup need to be an un-located asset?  Filter apples to highway hundredths, other highway lookups (like route, hwy), and installations/signs/supports/maintlog.  Broker will include only the highways under the crew if fldCrewOnlyFlag = T  Filter apples to highway hundredths, other highway lookups (like route, hwy), and installations/signs/supports/maintlog. |

| **Access DATA** | **Access data base** | **TI Source** |
| --- | --- | --- |
| **Table Name** |
| tblSheeting | *SignClient* | Domain: SIGN\_SHEETING |
| tblSubstrate | *SignClient* | Domain: SIGN\_SBSTR |
| tblSide | *SignClient* | XSP see notes: - what do we here – Jason up |
| tblDirections | *SignClient* | Domain: GEN\_DIR |
| tblDistricts | *SignClient* | TIODS\_MAINT\_CREW\_HIER where CREW=SNCW |
| tblHighwayEA | *SignClient* | TIODS \_MAINT\_CREW\_HIER & ODS\_RTE\_HWY\_LOC |
| tblHighways | *SignClient* | TIODS \_RTE\_HWY\_LOC\_ |
| tblHWYLkup | *SignClient* | ODS\_RTE\_HWY\_LOC & ODS\_MAINT\_ HIER |
| tblRouteIntestate | *SignClient* | TIODS \_RTE\_HWY\_LOC\_ |
| tblRoutesUS | *SignClient* | TIODS \_RTE\_HWY\_LOC\_ |
| tblRoutesState | *SignClient* | TIODS \_RTE\_HWY\_LOC\_ |
| tblMilePostPrefixes | *SignClient* | ODS\_RTE\_HWY\_LOC & Mileage Type Domain |
| tblHwyGPSData | *signdata* | ODS\_MP\_HUND & ODS\_MAINT\_ HIER |
|  |  | *See P490 Sign Asset Spec for TIG Extract to SFA LOV mapping* |
|  |  | *Y/N is a data type in Access – no need to sync YES\_NO domain* |
|  |  | *LOV that require joins, which require TIG – may require re-design* |
| LOV’s Managed in Access |  | These LOV’s will be populated from central server in the event of an update or a fresh SFA data install – i.e. on a new toughbook |
| tblMajorMinor | *SignClient* |  |
| tblSOIDefaultValues | *SignClient* |  |
| tblDistrictLkup | *SignClient* | Jason to mod, add field fldCrew\_Only\_Flag, pop yes |
| tblVersionControl | *SignClient* |  |
| tblDfltDist | *signdata* | In signdata db – 1 record one attribute |
| tblDinstricte |  |  |
| tblSignFacing | *SignClient* | Transformation ? Jason look for use of id – may get numeric – instead of alpha code. |
|  |  |  |

| **TI DATA** | **Schema/location** | **New/**  **Updated Field**  **Rules/Notes** |
| --- | --- | --- |
| **Field Name** |
| SIGN\_SHEETING | *TI Domain* |  |
| SIGN\_SBSTR | *TI Domain* |  |
| GEN\_DIR | *TI Domain* | *tblSIDE* |
| TI\_RPT\_MP\_HUND | *Report* |  |
| TI\_RPT\_MAINT\_CREW\_HIER | *Report* |  |
| TI\_RPT\_RTE\_HWY\_LOC\_ | *Report* |  |
| NM\_INV\_DOMAINS\_ALL.DATE\_MODIFIED | *TI* |  |
| Mileage type | *TI Domain* |  |

|  |  |  |
| --- | --- | --- |
| Data Transformation Rules | | |
| TI | SFA | Rule |
|  |  |  |
|  |  |  |
|  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validated LOV content 2. Create dummy domain values for all shared domains/LOV 3. Manually validate updates 4. Create dummy routes, create new EA, modify EA extent 5. Manually validate updates 6. Intentionally interrupt sync test continuation |  |  |

* 1. TI Un-located Asset LOV Sync

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To maintain complex – multi attribute – domains, and to provide TI admin rights required to maintain target LOV’s* |
| **Triggers** (What kicks off this task?) | *Completion of 4.0 tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Un-located TI assets: Actions (SNAC), Causes SNCS, Standard Sign TYPE (SIGN), Supports (SUPP), Standard Sign Graphics (SNGR), SIDES (SNSD)* |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES/Questions** |
| 1. Pre Sync: API Domain trigger updates domain on un-located asset update – except for side 2. If active is F, trigger appends “retired-” to the beginning of the meaning in the domain. 3. If the un-located asset instance is end-dated, then the trigger will end-date the domain value. 4. On sync API Evaluates target assets for modification sense last sync 5. API passes updated Asset values to broker for insertion into target SFA LOV tables 6. In SFA delete values in target LOV table, repopulate with values from TI passed to broker in previous step |  | Add attribute to standard signs, standard graphics and supports un-located asset “active” T/F – syncs with “fldshow” in the access database lookup table  If the end-dated value is used in the future, it would be rejected as an exception  Actions and causes will not be end-dated/retired  Ask Jason re support LOV |

| **Access DATA** | **Access database** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table Name** |
| tblActions | *SignClient* |  |  |
| tblCauses | *SignClient* |  |  |
| tblStandardSigns | *SignClient* |  | *Can be end-dated,* |
| tblSupport | *SignClient* |  | *Can be end-dated/add fldshow show, set to T* |
| tblStandardSignGraphics | *SignClient* |  | *Can be end-dated/add show – add fldshow, set to T* |
|  |  |  |  |

| **TI Domain** | | **TI Asset** | **Rules/Notes** |
| --- | --- | --- | --- |
|
| SIGN\_ACTN | *SNAC* | |  |
| SIGN\_CAUSE | *SNCS* | |  |
| STD\_SIGN\_TYPE | *SIGN* | | *Add ACTIVE attribute, set to T in conversion* |
| SIGN\_SUPP | *SUPP* | | *Add ACTIVE attribute, set to T in converson* |
| SIGN\_STD\_GRAF | *SNGR* | | *Add ACTIVE attribute, set to T in converson* |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Add dummy records to un- located assets 2. Manually Validate domain update trigger 3. Test sync 4. Manually validate updates |  |  |

1. TI to SFA Sync
   1. TI to SFA Installation Updated and End-date Sync

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To updates SFA assets from TI* |
| **Triggers** (What kicks off this task?) | *Completions of 5.0 tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Updated or end-dated Sign Installations in TI* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Read action listing 2. If New:    1. API pulls target installation data record from NM\_INV\_ITEMS\_ALL (SNIN)and NM\_MEMBERS\_ALL    2. Broker passed data to Broker for insertion into tblInstallations    3. Broker process child assets as prescribed in the following unit tasks 3. If location updated:    1. API pulls target installation data record from NM\_INV\_ITEMS\_ALL (SNIN)and NM\_MEMBERS\_ALL    2. Broker applies updated attributes to target installations    3. API process child assets as prescribed in the following unit tasks 4. If end-dated    1. API pulls target installation data record from NM\_INV\_ITEMS\_ALL (SNIN)and NM\_MEMBERS\_ALL    2. Broker deletes target installation and all Child asset records    3. API process child assets as prescribed in the following unit tasks | In the event of a an end-date installation because of a end-dated datum and the installation has been edited in the field:  Broker will update the end-dated asset with any changes to that asset since the last sync  Broker will delete the sign off Toughbook  If there is a failure when sending data from TI to Access, it will be logged in the oracle exception table. It will be put into as a retry entry that the broker will manage. It will appear as exception every time there sync until corrected. |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table Name** |
| tblInstallations |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| NM\_INV\_ITEMS\_ALL (SNIN)and NM\_MEMBERS\_ALL |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. ADD, updated and end-date installation in TI 2. Validated updates and deletions, including child assets post sync |  |  |

* 1. TI to TI to SFA Maintenance History Update and End-date Sync

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To updates SFA assets from TI* |
| **Triggers** (What kicks off this task?) | *Completions of 6.1 tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Updated or end-dated Supports or Maintenance History* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Read action listing 2. If New:    1. API pulls target installation data record from NM\_INV\_ITEMS\_ALL (SNML)    2. Broker establishes Parent InstalledId    3. Broker passed InstalledId with SNML data for insertion into tblInstallationHistory; 3. If end-dated (rare if ever)    1. API pulls target support data record from NM\_INV\_ITEMS\_ALL (SNML)    2. API pulls all targets support sibling data records    3. Broker deletes target Child asset records    4. Broker inserts sibling records | If there is a failure when sending data from TI to Access, it will be logged in the oracle exception table. It will be put into as a retry entry that the broker will manage. It will appear as exception every time there sync until corrected. |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table Name** |
| tblInstallationHistory |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| NM\_INV\_ITEMS\_ALL (SMML) |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. ADD, update and end-date SNML in TI 2. Validated updates and deletions |  |  |

* 1. TI to SFA Support Update and End-date Sync

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To updates SFA assets from TI* |
| **Triggers** (What kicks off this task?) | *Completions of 6.1 tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Updated or end-dated Supports or Maintenance History* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Read action listing 2. If New:    1. API pulls target Support data record from NM\_INV\_ITEMS\_ALL (SNSU)    2. Broker establishes Parent InstalledId    3. Broker passed InstalledId and support data to Broker for insertion into tblSupports; 3. If end-dated    1. API pulls all targets support sibling data records    2. Broker deletes target Child asset records    3. Broker inserts sibling records | If there is a failure when sending data from TI to Access, it will be logged in the oracle exception table. It will be put into as a retry entry that the broker will manage. It will appear as exception every time there sync until corrected. |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Table Name** |
| tblSupports |  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| NM\_INV\_ITEMS\_ALL (SNSU) |  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. ADD, update and end-date Support in TI 2. Validated updates and deletions |  |  |

* 1. TI to SFA SIGN (Standard and Custom) Update and End-date

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To updates SFA assets from TI* |
| **Triggers** (What kicks off this task?) | *Completions of 6.1 tasks* |
| **Dependencies** (What has to exist before this Task can start?) | *Updated or end-dated Sign (SNSN)* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Read action listing 2. API pulls target Support data record from NM\_INV\_ITEMS\_ALL (SNSN) 3. If New, and SIGN\_TYP=S:    1. Broker establishes Parent InstalledId    2. Broker passes InstalledId and standard sign data to Broker for insertion into tblInstalledStandardSigns; 4. If end-dated and SIGN\_TYP=S:    1. API pulls all targets support sibling data records (if any)    2. Broker deletes target Child asset records    3. Broker inserts sibling records 5. If New and SIGN\_TYP=C:    1. Broker establishes Parent InstalledId    2. Broker passes InstalledId and Custom sign data to Broker for insertion into tblInstalledCusotmSigns (minus CUSTOM\_LEG)    3. Broker pulls newCustLegnedId (this is an autonumber field populated at insertion of new custom sign in previous step) from tblInstalledCustomSigns; transforms CUSTOM\_LENG for insertions into tblCustomSignLegends 6. If end-dated and SIGN\_TYP=C:    1. API pulls all targets support sibling data records    2. Broker deletes target Child asset records Broker inserts sibling records into tblInstalledCustomSigns and tblCustomSignLegends | If there is a failure when sending data from TI to Access, it will be logged in the oracle exception table. It will be put into as a retry entry that the broker will manage. It will appear as exception every time there sync until corrected. |

| **Access DATA** | **Data Type** | **Updated Field**  **Rules/Notes** |
| --- | --- | --- |
| **Table Name** |
| tblInstalledStandardSigns |  |  |
| tblInstalledCusotmSigns |  |  |
| tblCustomSignLegends |  |  |
|  |  |  |

| **TI DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| NM\_INV\_ITEMS\_ALL (SNSN) |  |  |  |

|  |  |  |
| --- | --- | --- |
| Data Transformation Rules | | |
| TI | SFA |  |
|  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. ADD, update and end-date SNSN in TI 2. Validated updates and deletions 3. Validate Transformation of SignLegend |  |  |

1. Sync Status
   1. Create Exception Table in Oracle

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI DBA* |
| **Purpose** (Why does this task exist; what value does it add?) | *Provides data storage for sync error log* |
| **Triggers** (What kicks off this task?) |  |
| **Dependencies** (What has to exist before this Task can start?) |  |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Create XSIGN\_EXECPT table as described below | Table Location? Report Schema? |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| XSIGN EXECPT Taple | | | | | | | | |
| Column Name | SYNC\_DATE | TI\_USERNAME | NE\_ID | INSTALLATION\_ID | STATEHWY | MP | ERROR\_TYPE | SYNC\_ERROR |
| Column Type | DATE | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | Number | VARCHAR2 | VARCHAR2 |
| Size |  | 20 | 10 | 10 | 10 | NUM | 20 | 250 |

|  |  |
| --- | --- |
| Field | Description |
| SYNC\_DATE | Date of Synchronization |
| TI\_USERNAME | Oracle Username of sync initiator |
| NE\_ID | NE\_ID of Asset creating an error in the TI to SFA Sync phase |
| INSTALLATION ID | Installation ID of Asset creating an error in the SFA to TI sync phase |
| STATEHWY | State Highway of Asset |
| MP | MP location of Asset |
| ERROR TYPE | Asset type, or Domain / LOV type |
| SYNC ERROR | Sync Error Description |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| * 1. Validate table design |  |  |

* 1. SFA to TI Sync Exceptions

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To write sync failure records to* XSIGN\_EXECPT |
| **Triggers** (What kicks off this task?) | *SFA to TI Data Sync Failure* |
| **Dependencies** (What has to exist before this Task can start?) | *XSIGN\_EXCEPT table, Sign data updates in SFA, and an initiated sync process* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Establish sync error messages– created easily understood “decoded” oracle error messages, 2. Initiating action: SFA to TI Asset Sync Failure    1. Capture and decode error    2. Write Exception record to XSIGN\_EXCEPT    3. Write Exception record to text file, imbed sync date/time in file name– for the specific Session ID exclude fields Highlighted in yellow below, Write text file to: C:\work , open file on the desk top.    4. Do not include TI to SFA sync errors | If a single installation is updated centrally (or on a different Toughbook and synced) and on the Toughbook, during the same sync cycle then the second Toughbook wins but an exception is logged saying that it overwrote the data. (thus, if the date on the TI record is newer than the last sync date, then it will be overwritten but create an exception) – these conflicts are identified and the task flagged as an error in the task listing  If an installation is edited twice in the same day write to oracle sync log do not remove from edit table let it sync next day – write exception – date track error |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | XSIGN EXECPT Table | | | | | | | | |
| Column Name | SYNC\_DATE | Session ID | TI\_USERNAME | NE\_ID | INSTALLATION\_ID | STATEHWY | MP | ERROR\_TYPE | SYNC\_ERROR |
| Column Type | DATE | NUMBER | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | Number | VARCHAR2 | VARCHAR2 |
| Size |  |  | 20 | 10 | 10 | 10 | NUM | 20 | 250 |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| * 1. Determine all possible error messages   2. Create errors in test data and validate error listing |  |  |

* 1. TI to SFA Sync Exceptions

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI API / Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To wright sync failure records to* XSIGN\_EXECPT |
| **Triggers** (What kicks off this task?) | *SFA to TI Data Sync Failure* |
| **Dependencies** (What has to exist before this Task can start?) | *XSIGN\_EXCEPT table, Sign data updates in TI, and an initiated sync process* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Write Access error message to table 2. Initiating action: TI to SFA Asset Sync Failure    1. Capture and error    2. Write Exception record to a XSIGN\_EXCEPT | **Errors TBD**   1. If there is a failure when sending data from TI to Access, it will be logged in the oracle exception table. It will be put into as a retry entry – ask Joe - that the broker will manage. It will appear as exception every time there sync until corrected. |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| * 1. Determine all possible all possible error messages   2. Create errors in test data and validate error listing |  |  |

* 1. Create Sync Status Table in Oracle

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *TI DBA* |
| **Purpose** (Why does this task exist; what value does it add?) | *Provides data storage for sync error log* |
| **Triggers** (What kicks off this task?) | *End of a successful sync* |
| **Dependencies** (What has to exist before this Task can start?) |  |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Create XSIGN\_STATUS table as described below | Create in Report Schema |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SYNC STATUS Columns | | | | | | | | | | |
| Column Name | SYNC\_DATE | TI\_USERNAME | SFA\_CNT | SFA\_EXP | TI\_CNT | TI\_EXP | LOV\_CNT | LOV\_EXC | MP 100\_CNT | MP 100\_EXP |
|  |  |  |  |  |  |  |  |  |  |  |
| Column Type | DATE | VARCHAR2 | Number | Number | Number | Number | Number | Number | Number | Number |
| Size |  | 20 |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Field | Description |
| SYNC\_DATE | Date of Synchronization |
| TI\_USERNAME | Oracle Username of sync initiator |
| SFA\_CNT | Count of SFA to TI Assets Updates |
| SFA\_EXP | Count of SFA to TI Exceptions |
| TI\_CNT | Count of TI to SFA Updates |
| TI\_EXCP | Count of TI to SFA Exceptions |
| LOV\_CNT | Count of LOV table updates |
| LOV\_EXP | Count of LOV table Exceptions |
| 100\_CNT | Count of 100 table updates |
| 100\_EXP | Count of 100 table Exceptions |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| * 1. Validate table design |  | **Joe to evaluate Access Exception count** |

* 1. Sync Status Update

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Broker* |
| **Purpose** (Why does this task exist; what value does it add?) | *To wright sync status records to* XSIGN\_STATUS |
| **Triggers** (What kicks off this task?) | *Completions of sync process* |
| **Dependencies** (What has to exist before this Task can start?) | *Completion all sync process tasks* |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **NOTES/Questions** |
| 1. Write sync status data, as defined in the previous task to the XSYNC\_STATUS table. |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| * 1. Complete multiple sync updates, Validated sync status records |  |  |

1. Initial Data Conversion
   1. Initial Data Conversion

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *SFA Sign Integration TEAM* |
| **Purpose** (Why does this task exist; what value does it add?) | *1. To create a complete test database in TI for Testing*  *2. To learn from and develop final Data conversion process* |
| **Triggers** (What kicks off this task?) |  |
| **Dependencies** (What has to exist before this Task can start?) |  |

|  |  |
| --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System/actor**  **NOTES/Questions** |
| 1. Acquire SFA signdata databases from each district | Jason to prepare and provide files to Bentley |
| 1. Broker perform initial SFA to TI Load | Hieratical loaders to be provided by Bentley. |
| 1. Create CSV Loaders for TI only Sign domains – SIGN\_SHEETING, SIGN\_SBST LOV | See 5.1 |
| 1. Create loader for un-located Assets | See 5.2 |
| 1. Create new sign Domains in TI | See 5.1 / 5.2 |
| 1. Broker Populated SFA LOV’s from TI reports | See 5.1 |

| **Access DATA** | **New/**  **Updated Field**  **Rules/Notes** | |
| --- | --- | --- |
| *Asset Data* |
| tblInstallations | *Asset Data* |  |
| tblInstalledCustomSigns |
| tblInstalledStandardSigns |
| tblCustomeLegends |
| tblInstalledSuppports |
| tblInstallationHistorey |
| *Simple Domains* | | |
| tblSheeting | *LOV* | *Load into TI Domain* |
| tblSubstrate |
| *LOVs derived from TI Reports* | | |
| tblDirections | *LOV* | *Populate SFA LOV with TI Reports, see 5.1, needed for operational SFA tests* |
| tblDistricts |
| tblHighwayEA |
| tblHighways |
| tblHWYLkup |
| tblRouteIntestate |
| tblRoutesUS |
| tblRoutesState |
| tblMilePostPrefixes |
| tblHwyGPSData |
| *LOV used to pop un located assets* | | |
| tblActions | *LOV* | *Un located asset, provide trigger to populated TI Domain*  *Pre* conversion set all standard graphics and supports fldshow to T  CD to develop CSV and Load – CSV provided to Bentley |
| tblCauses |
| tblStandardSigns |
| tblSupport |
| tblStandardSignGraphics |
|  |  | |

| **TI DATA** | **Data Type** | **New/**  **Updated Field**  **Rules/Notes** |
| --- | --- | --- |
| *TI asset tables* |
| NM\_INV\_ITEMS\_ALL |  |  |
| NM\_MEMBERS\_ALL |
| *TI Assets* | | |
| SNIN - installation |  | Use start date – changed to sync date on 420 - of the highway for initially data sync |
| SNSU – supports |
| SNSN - Signs |
| SNML – Sign Maintenance Log |
| *TI Domains* | | |
| SIGN\_SHEETING |  |  |
| SIGN\_SBSTR |
| SIGN\_ACTN | *TI – UNL asset SNAC* |
| SIGN\_CAUSE | *SNCS* |
| STD\_SIGN\_TYPE | *SIGN* |
| SIGN\_SUPP | *SUPP* |

|  |  |  |
| --- | --- | --- |
| Data Transformation Rules | | |
| TI | SFA | Rule |
|  |  |  |
|  |  |  |
|  |  |  |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Validate asset data load – manually asset count, spot check assets and attribution 2. Validte TI maintained Domains – count, spot check 3. Validate un-located asset – instance count, validate domain update trigger. 4. Validated SFA LOV tables created from TI Report, record count spot check data 5. Manually validate updates |  |  |

1. Data Transformation Requirements
   1. Initial Data Conversion

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Initial data load/ Broker during Sync* |
| **Purpose** (Why does this task exist; what value does it add?) | *To list all transformation requirements in one place* |
| **Triggers** (What kicks off this task?) | *NA* |
| **Dependencies** (What has to exist before this Task can start?) |  |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System/actor** | **NOTES/Questions** |
| 1. Pre conversion: tblInstallation.RDWYID converted 1 to I, 2 to D in |  |  |
| 1. In process sync transformation: tblInstalletion.fldSide to XSP |  | Ref. 4.1, 4.4 |
| 1. In process sync: Ignore creation data set startdate for a new installation to sync date |  | Ref. 4.1 |
| 1. In process SFA to TI sync transformation: tblCustomSignLegends.fldLegend to SNSN.CUSTOM\_LGND |  | Ref. 4.6 |
| 1. In process TI to SFA sync transformation: SNSN.CUSTOM\_LGND to tblCustomSignLegends.fldLegend, and fldLindNo |  | Ref 6.4 |
| 1. Pre conversion: set tblHWYLkup. MileageType = 00,P0 or Z0-9 (last 2 char of the LRM) |  |  |

| **Data Type** | **New/**  **Updated Field** | **Rules/Notes** | | | |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | | |  |  |  | |
|  | | |  |  |  | |
|  | | |  |  |  | |

| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
| 1. Create dummy domain values for all shared domains/LOV 2. Manually validate updates |  |  |

1. Client (Toughbook) Configuration

|  |  |
| --- | --- |
|  |  |
| **Actor** (What person, job function or system performs this Task?) | *Bentley Developer will create of xml config files. Toughbook Configuration service provided by ODOT IT Staff* |
| **Purpose** (Why does this task exist; what value does it add?) | *Each Toughbook or SFA client has to have an Oracle Login and an assigned district and crew. The Login will we stored in an encrypted xml file, the District and Crew Id is stored in a standard XML file* |
| **Triggers** (What kicks off this task?) |  |
| **Dependencies** (What has to exist before this Task can start?) |  |

|  |  |  |
| --- | --- | --- |
| **STEPS** (Sequence of lower level of detail in the task) | **System** | **NOTES** |
| * 1. The password and District/Crew template xml file will be downloaded from the xcopy site along with the Broker.exe and the SFA Access sign application – the url is: ?   2. The IT Technician will acquire the assigned UN/PW from the TI DBA, and the assigned district/crew from the Districted Sign maintenance manager.   3. The TI Technician will the configure the login and district/crew xml files as documented and place them at c:\ ?   4. The TI technician will copy the SFA Access application to c:\?   5. The TI technical will then initiate the Broke |  | What is the URL for XCOPY – can an IT technician initiate a download from the site.  Joe M needs to determine the file location for the XML files  What is the file location for the SFA Access Application  Configuration documentation required |
|  |  |  |

| **Access DATA** | **Data Type** | **New/**  **Updated Field** | **Rules/Notes** |
| --- | --- | --- | --- |
| **Field Name** |
| None |  |  |  |

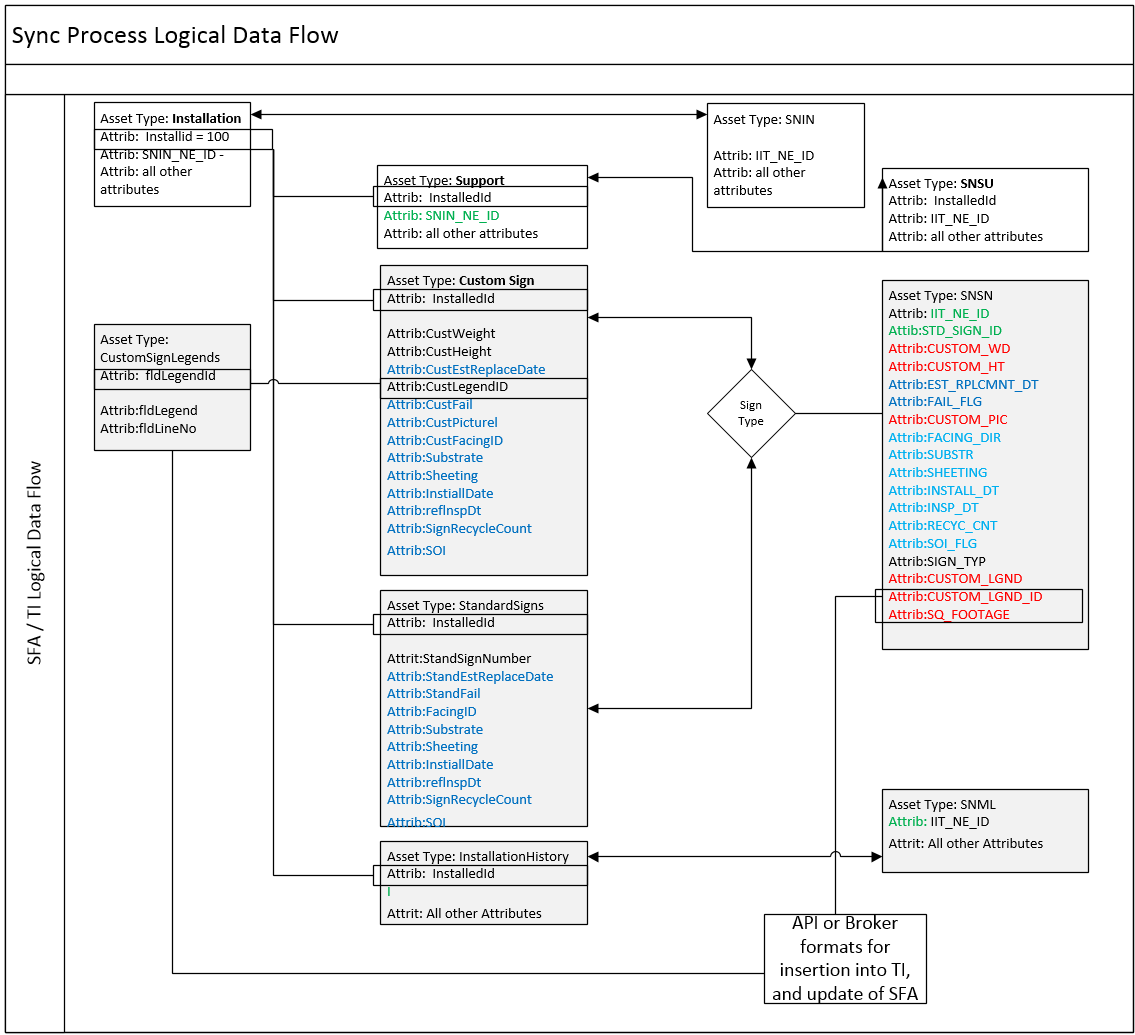
| **Test/Exception** | **Specific Instructions** | **Notes** |
| --- | --- | --- |
| **Case** |
|  |  |  |

# Appendix

## SFA tblInstallations.fldSide to TI Installation (SNIN) XSP Transformation Crosswalk

|  |  |  |  |
| --- | --- | --- | --- |
| **XSP/Value** | **tblinstallations.fldSide** | **Description** | **Transformation** |
| ITRL | L | Travel Left on increasing | where tblinstallations.flddirection = hwy.general direction |
| DTRL | L | Travel Left on decreasing | where tblinstallations.flddirection is opposite hwy.general direction |
| ITRR | R | travel right on increasing | where tblinstallations.flddirection = hwy.general direction |
| DTRR | R | travel right on decreasing | where tblinstallations.flddirection is opposite hwy.general direction |
| O | O | over | straight crosswalk |

## Sync Process Logical Data Flow Diagram



## TI to SFA, New Custom Sign Sync Diagram

