FASAMS Requirements Document

Build FASAMS Dimensional Model part 2 14777

Created by: Jesse Lindsey

Dated: 10/1/2020

Contents

[General 3](#_Toc56610565)

[Version Control 3](#_Toc56610566)

[Stakeholders 3](#_Toc56610567)

[Supporting Documents and References 3](#_Toc56610568)

[Terms and Definitions 3](#_Toc56610569)

[Scope 3](#_Toc56610570)

[Initial Request & Assumptions 3](#_Toc56610571)

[Business Statement 3](#_Toc56610572)

[Scope Definition 4](#_Toc56610573)

[Assumptions, Constraints, Dependencies 4](#_Toc56610574)

[Existing Functionality 4](#_Toc56610575)

[Proposed Functionality 4](#_Toc56610576)

[Initial Estimate/ Quote History 4](#_Toc56610577)

[Requirements 5](#_Toc56610578)

[1. Develop ETL Matching Process 5](#_Toc56610579)

[2. Develop ETL data transformation Process 8](#_Toc56610580)

# General

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author(s) | Brief Description of Change |
| 10/6/2020 | 1.0 | Jesse Lindsey | Initial document creation from Part 1 doc |
| 10/13/2020 | 1.1 | Jesse Lindsey | Update doc |
| 10/20 – 10/23 | 1.2 | Jesse Lindsey | Updates throughout week |
| 10/27 – 10/29 | 1.3 | Jesse Lindsey | Updates throughout week |
| 11/12/2929 | 1.4 | Jesse Lindsey | Updates |
| 11/18/2020 | 1.5 | Jesse Lindsey | Update doc |
|  |  |  |  |

## Stakeholders

|  |  |
| --- | --- |
| Role | Name |
| FEI Implementation Manager | Jesse Lindsey |
| FEI Account Manager | Kory Schnoor |
| FEI Product Manager | Jessica Knott |
| DCF OITS Project Director | Nathan McPherson |
| DCF SAMH Project Sponsor | Jonathan Hall |
| DCF Business Analysts | Gregory Nix / Victor Gaines |

## Supporting Documents and References

## Terms and Definitions

|  |  |
| --- | --- |
| Terms | Description |
| ODS | Operational Data Store (database tables that house the submission data from XML file submissions) |
|  |  |
|  |  |

# Scope

## Initial Request & Assumptions

### Business Statement

DCF needs a way to report on both FASAMS Version 13 and Version 14 data within the same reporting database. To achieve this, a unified reporting dimensional model database will be built which combines data from both versions of FASAMS.

### Scope Definition

The dimensional model will be developed to include all tables necessary to support all current FASAMS reports that do not need to provide immediately submitted data. The model will not support the generation of extract data (i.e. TEDS, SHR, BCI).

An automated nightly Extract Transform Load (ETL) process will be set up to populate the dimensional model.

The Dimensional Model will not include the following ODS tables:

* BCI Extract tables
* SHR Extract tables
* TEDS Extract tables
* X12 Send, receive, acknowledgement for Medicaid Eligibility and Enrollment tables
* ContractTarget table

ContractedAmount in Version 13 SubcontractService entity will be summed and used to populate the TotalSubcontractOcaAmount in Version 14 SubcontractOca entity.

### Assumptions, Constraints, Dependencies

The Dimensional model nightly load process will add all records into the model, excluding duplicate records.

## Existing Functionality

FASAMS currently does not have a dimensional model that supports multiple version of the system. FASAMS currently runs reports off of the FASAMS Version 13 Operational Data Store (ODS) database.

## Proposed Functionality

The FASAMS system will automatically populate the Dimensional model each night with data from both version 13 data and version 14 data.

## Initial Estimate/ Quote History

|  |  |  |  |
| --- | --- | --- | --- |
| Phase/ Quote Date | Item(s) | Description | Estimate/ Quote |
| 09/03/2020 | 14176 | Initial Estimate | 376 - 750 |
| 09/03/2020 | 14176 | Final Estimate | **Final Estimate:**  Analysis - 40  Dev - 184  Test - 136  Acct / Project Management - 16  **Total - 376** |

# Requirements

## Develop ETL Matching Process

| Req. ID | Requirement | Ref. ID | Dev | QA |
| --- | --- | --- | --- | --- |
| 1.0 | Create matching logic between Version 13 records and Version 14 records |  |  |  |
| 1.1 | Create ETL for DimProvider table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Providers are loaded from version 13 to the DimProvider table. From version 14 we match on SubmittingEntity and FederalTaxIdentifier to prevent duplicate DimProvider record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.2 | Create ETL for DimProviderClient table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All ProviderClients are loaded from version 13 to the DimProviderClient table. From version 14 we match on SubmittingEntity, Provider, and on following first non-null value: ClientKey, SSN, Pseudo SSN, Medicaid number to prevent duplicate DimProviderClient record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.3 | Create ETL for DimSubcontract table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Subcontracts are loaded from version 13 to the DimSubcontract table. From version 14 we match on SubmittingEntity +ContractNumber + SubcontractNumber + AmendmentNumber to prevent duplicate DimSubcontract record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.4 | Create ETL for DimProviderSite table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Provider Sites are loaded from version 13 to the DimProviderSite table. From version 14 we match on SubmittingEntity + Provider + SiteIdentifier to prevent duplicate DimProviderSite record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.5 | Create ETL for FactWaitingList table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All WaitingLists records are loaded from version 13 to the FactWaitingList table. From version 14 we match on SubmittingEntity + ProviderClient + ProviderSite + PlacementDate to prevent duplicate FactWaitingList record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.6 | Create ETL for FactProviderTreatmentEpisode table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + Second try: All Treatment Episodes are loaded from version 13 to the FactProviderTreatmentEpisode table. From version 14 we match on SubmittingEntity + ProviderClient + Provider + InitialAdmissionDate/ImmediateDischarge.EvaluationDate + ProgramAreaCode to prevent duplicate FactProviderTreatmentEpisode record creation. If a record was submitted in version 14, we will ignore any changes from version 13.   + Note: a crosswalk will be added to the ETL process for specific entities unable to use the same SRI or AdmissionDate/EvaluationDate from V13 * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.7 | Create ETL for FactImmediateDischarge table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All ImmediateDischarges are loaded from version 13 to the FactImmediateDischarge table. From version 14 we match on SubmittingEntity + Provider Treatment Episode + EvaluationDate to prevent duplicate FactImmediateDischarge record creation. If a record was submitted in version 14, we will ignore any changes from version 13. * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.8 | Create ETL for FactAdmission (Admission) table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Admission records are loaded from version 13 to the FactAdmission (Admission) table. From version 14 we match on SubmittingEntity + Provider Treatment Episode + ProgrmAreaCode + AdmissionDate to prevent duplicate FactAdmission (Admission) record creation. If a record was submitted in version 14, we will ignore any changes from version 13.   + Note: a crosswalk will be added to the ETL process for specific entities unable to use the same SRI or AdmissionDate/EvaluationDate from V13 * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.9 | Create ETL for FactAdmission (Discharge) table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Admission records are loaded from version 13 to the FactAdmission (Discharge) table. From version 14 we match on SubmittingEntity + Provider Treatment Episode + Admission + Discharge Date to prevent duplicate FactAdmission (Discharge) record creation. If a record was submitted in version 14, we will ignore any changes from version 13.   + Note: a crosswalk will be added to the ETL process for specific entities unable to use the same SRI or AdmissionDate/EvaluationDate from V13 * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.11 | Create ETL for FactEvaluation table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Evaluations are loaded from version 13 to the FactEvaluation table. From version 14 we match on SubmittingEntity + Provider Treatment Episode + Admission/Discharge + EvaluationDate + TypeCode + ToolCode to prevent duplicate FactEvaluation record creation. If a record was submitted in version 14, we will ignore any changes from version 13. * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.12 | Create ETL for FactDiagnosis table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All Diagnoses are loaded from version 13 to the FactDiagnosis table. From version 14 we match on SubmittingEntity + Provider Treatment Episode + Admission/Discharge + DiagnosisCode + StartDate to prevent duplicate FactDiagnosis record creation. If a record was submitted in version 14, we will ignore any changes from version 13. * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.13 | Create ETL for FactPerformanceOutcomeMeasure table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All POMs are loaded from version 13 to the FactPerformanceOutcomeMeasure table. From version 14 we match on SubmittingEntity + Provider Treatment Episode + Admission or Discharge + Date to prevent duplicate FactPerformanceOutcomeMeasure record creation. If a record was submitted in version 14, we will ignore any changes from version 13. * If there is no match on SRI or compsite key then a new record is inserted into the Dim Model |  |  |  |
| 1.14 | Create ETL for FactBedCapacity table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All BedCapacities are loaded from version 13 to the FactBedCapacity table. From version 14 we match on SubmittingEntity + FederalTaxIdentifier + Effective Date + Facility type + age group to prevent duplicate FactBedCapacity record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.15 | Create ETL for FactBedTypeCapacity table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All BedTypeCapacities are loaded from version 13 to the FactBedTypeCapacity table. From version 14 we match on SubmittingEntity + Provider/Contract/Subcontract + Effective Date + BedType to prevent duplicate FactBedTypeCapacity record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.16 | Create ETL for FactBedUtilization table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All BedUtilizations are loaded from version 13 to the FactBedUtilization table. From version 14 we match on SubmittingEntity + Provider/Contract/Subcontract + Effective Date + CensusDate + Facility type + age group to prevent duplicate FactBedUtilization record creation. If a record was submitted in version 14, we will ignore any changes from version 13. * Note: FactPayorClassBedOccupancy table identified in 1.22 |  |  |  |
| 1.17 | Create ETL for FactOutcomeMeasure table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All OutcomeMeasures are loaded from version 13 to the FactOutcomeMeasure table. If subcontract/contract has records submitted in v14, delete v13 and use only v14. |  |  |  |
| 1.18 | Create ETL for FactOutputMeasure table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All OutputMeasures are loaded from version 13 to the FactOutputMeasure table. If subcontract/contract has records submitted in v14, delete v13 and use only v14. |  |  |  |
| 1.19 | Create ETL for FactSubcontractOca table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All SubcontractOcas are loaded from version 13 to the FactSubcontractOca table. From version 14 we match on SubmittingEntity + Subcontract + Contract + Provider + ExpenditureOca to prevent duplicate FactSubcontractOca record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.20 | Create ETL for FactSubcontractService table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All SubcontractServices are loaded from version 13 to the FactSubcontractService table. From version 14 we match on SubmittingEntity + Contract + Subcontract + ProgramAreaCode + ExpenditureOCA + Covered Service/ProjectCode + EffectiveDate to prevent duplicate FactSubcontractService record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.21 | Create ETL for FactServiceEvent table   * All ServiceEvents are loaded from version 13 to the FactServiceEvent table. * Client Specific Service Events:   + Matching Version 13 ServiceEvents to Version 14 ServiceEvents:     - First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.       * From version 13 we match SSN/PSSN (via EpisodeSRI) + ProgramAreaCode + StaffIdentifier + CoveredServiceCode + HcpcsProcedureCode + ServiceDate + StartTime to prevent duplicate FactServiceEvent record creation. If a record was submitted in version 14, we will ignore any changes from version 13. * Non-Client Specific Service Event:   + From Version 13 we match on ContractNumber + SubcontractNumber + StaffIdentifier + CoveredServiceCode + HcpcsProcedureCode + ServiceDate + StartTime. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.22 | Create ETL for FactPayorClassBedOccupancy table   * First, the ETL process will look for a matching SRI within the Submitting Entity/Provider combo. If no SRI exists, then use compsite keys listed below.   + All FactPayorClassBedOccupancy are loaded from version 13 to the FactPayorClassBedOccupancy table. From version 14 we match on Parent Entity + PayorClassCode to prevent duplicate FactPayorClassBedOccupancy record creation. If a record was submitted in version 14, we will ignore any changes from version 13. |  |  |  |
| 1.23 | Create ETL for FactProviderLicense table   * All ProviderLicense records are loaded from the ODS Provider License table |  |  |  |
| 1.24 | Create ETL for FactMedicaidEncounter table   * All Medicaid Encounter records are loaded from the Medicaid Encounter table |  |  |  |
| 1.25 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Develop ETL data transformation Process

| Req. ID | Requirement | Ref. ID | Dev | QA |
| --- | --- | --- | --- | --- |
| 2.0 | Create transformation logic to transform Version 13 data into Version 14 format   * **Note: All V13 data will remain in tables but V14 transformed data will also be added to the table** |  |  |  |
| 2.1 | V13 Admissions (initial and transfer) to V14 Admission + PlacementRecord.StartDate and PlacementRecord.PlacementCode   * V13 Initial Admission   + Becomes V14 Admission with same AdmissionDate   + Create new PlacementRecord under V14 Admission     - V13 Admission.AdmissionDate becomes first V14 PlacementRecord.StartDate     - V13 Admission.TreatmentSetting becomes V14 PlacementRecord.PlacementCode       * Based on table in req. 3.3 * V13 Transfer Admission   + Create new PlacementRecord under Admission     - V13 Admission.AdmissionDate becomes V14 PlacementRecord.StartDate     - V13 Admission.TreatmentSetting converts to V14 PlacementRecord.PlacementCode       * Based on table in req. 3.3 |  |  |  |
| 2.2 | V13 Discharges (transfer and final) to V14 Discharge + PlacementRecord.EndDate and PlacementRecordOutcomeCode   * V13 Transfer Discharge (associated with transfer admission)   + V13 Discharge.DischargeDate becomes V14 PlacementRecord.EndDate   + V13 Discharge.DischargeReasonCode converts to V14 PlacementRecord.PlacementOutcomeCode     - Based on table in 3.4 * V13 Final Discharge (associated with initial admission)   + V13 Discharge.DischargeDate becomes     - V14 PlacementRecord.EndDate for PlacementRecords without EndDate     - V14 Discharge.DischargeDate   + V13 Discharge.DischargeReasonCode converts to V14 PlacementRecord.PlacementOutcomeCode without PlacementOutcomeCodes     - Based on table in 3.4 |  |  |  |
| 2.3 | Conversion of TreatmentSettings to PlacementCodes:   * Treatment Settings: 02   + Converts to PlacementCode: 2 * TreatmentSettings: 03   + Converts to PlacementCode: 5 * TreatmentSettings: 04 and 05   + Converts to PlacementCode: 3 * TreatmentSettings: 06, 07, and 08   + Converts to PlacementCode: 4 |  |  |  |
| 2.4 | Conversion of DischargeReasonCode to PlacementOutcomeCode:   * DischargeReasonCodes: 1, 4, 24   + Converts to PlacementOutcomeCode: 1 * DischargeReasonCodes: 2, 14   + Converts to PlacementOutcomeCode: 2 * DischargeReasonCodes: 3, 5, 6, 7, 34, 96, 97   + Converts to PlacementOutcomeCode: 3 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |