

Table of Contents

[New Features in BimlFlex 2018.04](#)

[Release Notes](#)

New Features in BimlFlex 2018.04

Azure Blob Storage-based staging and persisting

BimlFlex now has full support for Azure Blob Storage as a staging and file persistence for both Azure SQL Data Warehouse and SQL Server 2016+ scenarios. The blob-based solution architecture allows data extracts to be compressed into flat files and transferred to Azure based blob storage. This is accessed through External Table definitions and loaded in to the Data Warehouse using ELT based processes. The blob-based files are available to use in other processes such as Azure Data Lake Analytics and Machine Learning scenarios where the tool can access blob storage-based datasets directly.

Full support for SQL Server CDC Sources

BimlFlex has added full SQL Server CDC source component support.

For sources using Microsoft SQL Server CDC, Change Data Capture, to derive deltas BimlFlex is now able to directly read only changed data.

Metadata modelling uses the base tables for metadata ingestion and modelling. For initial loads BimlFlex will read current state data from the base tables. For incremental loads BimlFlex will automatically derive deltas from the CDC tables.

Data Type Mappings

The Data Type Mappings function allows for expansion of data types. This allows the source data types to be converted to a more accommodating type on load. This in turn allows for changes in the source datatypes without affecting the data warehouse. As an example, a source with a varchar(10) column can update that column to a longer string or it can update to nvarchar to support Unicode characters. either change would require a rebuild of the load process to accommodate the new data type. A Data Type Mapping to nvarchar(100) would negate that need and allow the changed data type to be loaded with little or no impact.

The Data Type Mappings behavior has a set of default expansion rules. These can be customized in the **DataTypeMappings** Sheet in the BimlFlex settings.

Global Extension Points

Project Parameters might be used across multiple SSIS projects. This global target for parameters allows a single definition to be injected into all projects.

Use Extension Point **ProjectParameter** with target **@@global**:

```
<#@ extension bundle="BimlFlex.bimlb" extensionpoint="ProjectParameter" target="@@global" #>
<Parameter Name="YourParameterName" DataType="String" IsRequired="true">DefaultParameterValue</Parameter>
```

More information on this is available in this video:

Visual Studio SQL Server Data Tools, SSDT, projects

All SQL artefacts are now added to a separate Visual Studio SQL Server Data Tools, SSDT, project. This project can be used to source control and maintain the databases and SQL artefacts that are part of the Data Warehouse. This can be used for deploying to databases through Visual Studio, to create dacpacs for scripted deployments or as part of a Continuous Integration, Continuous Delivery pipeline.

A configuration setting has been added to the Bundle options that controls the generation of the SSDT Project.

BimlC.exe, command line Biml Compiler

BimlFlex projects can now be built using the BimlC Biml Compiler with settings files to control scope of build. This can be used for scripted deployments or as part of a Continuous Integration, Continuous Delivery. As an example it is commonly used to create SQL artefacts and deploying SQL databases and tables before SSIS packages are created. A BimlFlex Build will include a reference to the `bimlc.exe` build command line expression and response file for scenarios where this should be automated at a later stage.

```
To replicate this build with bimlc.exe. Use the bimlc.resp file:  
bimlc.exe @"C:\Path\Project\output\ProjectName.mst.bimlc.resp"
```

BimlFlex support for SQL Server 2017

BimlFlex and BimlStudio now support SQL Server 2017 as a destination for both Data Warehouse and SSIS. BimlFlex custom SSIS components have been updated to support SSIS 2017.

Metadata Snapshot management

BimlFlex Metadata management now has support to save and restore Snapshots of the current metadata. This is useful for versioning as well as rollback for rapid, agile development processes.

Data Vault Modelling

BimlFlex full support for insert only Data Vault implementations

BimlFlex Data Vault patterns now have full, end to end support for insert only Data Warehouses. This removes effective to dates, is current flags as well as the end dating logic to optimise for data warehouse load performance. As a query support construct, Point In Time tables can be used to derive full timelines when necessary. To exclude end dating code, use the setting `EnableEndDateRdv`. Set to `"N"` for insert only scenarios.

Data Vault Acceleration

Controlled acceleration of Link Satellites

A setting has been added that control if the Data Vault Accelerator should create Link Satellites for generated Links. For some modelling approaches the effectiveness of a relationship is tracked in a Hub rather than a Link Satellite, in other scenarios the link satellite adds limited functionality. This setting controls the automatic acceleration of the Link Satellites.

The setting `DvAccelerateLinkSatellite` has been added to the `Settings` sheet. This controls if Link Satellites should be accelerated. This feature has a default value of `"N"`, meaning no Link Satellites will be accelerated unless it is updated to `"Y"`. Overrides can be created using attribute `SettingValue` definitions for specific source tables that should have a different setting to the defined default.

Accelerating Same As Links

Added support for `Same As Links` in the Data Vault Accelerator

Data Modelling

BimlFlex support for `SelectBySql`

Metadata support for `SelectBySql` for `DISTINCT, TOP n` scenarios

BimlFlex support for `OrderBySql`

Metadata support for `OrderBySql` for `ORDER BY Column` scenarios

Performance Optimization

BimlFlex build support for `AutoAdjustBufferSize`

BimlFlex now has support for conditional setting of `AutoAdjustBufferSize` to `true` or `false` for built SSIS Packages in supported SSIS versions

BimlFlex updates to local caching and minification

BimlFlex now implements local caching and file minification to optimize performance

Table Create script updates

Create table scripts now use `Page` compression as default Create database scripts now create databases using simple recovery model

Additional Features

PSA table velocity profiling

BimlFlex now has, optional, added velocity profiling for PSA tables to track data change velocity.

Default updates to Business Keys

The metadata import dialog now has support to add the connections Record Source to the creation of the Business Key as well as changing references from source constraints to Business Key defined relationships. This allows a more rapid and agile Data vault modelling approach. Any derived key is by default set to `IsNotPersistent` meaning they are not stored in PSA unless specifically configured for persistence.

Parameters

Standard Parameter management through metadata has been updated to support parameter queries on source as well as additional columnar expressions. This is configured through metadata through `ParameterColumnExpression` and `ExecuteSqlOnSource` etc.

Initial load logic

BimlFlex will automatically check if the load is an initial load by querying the PSA or destination for existing records. This removes the `IsInitialLoad` project parameter dependency. `IsInitialLoad` is now automatically set by the load process.

Key hashing Option for SQL compatible Hashing

The BimlFlex custom SSIS Component used for in-stream hashing of keys has been updated to provide a configurable encoding. Configure the setting `UseSqlCompatibleHash` as "Y" to use hashing compatible with Azure SQL Data Warehouse and other SQL based CHECKSUMS and HASHBYTES functions using the `SHA1` hashing algorithm.

SSIS Optimization

Added `ApplyLookupFilterRdv` to filter SSIS lookups by joining to the Staging layer. This minimizes memory usage for lookup components. This optional cross database join functionality requires the databases to be co-located or the tables to be in the same database

Auditing and Logging

Added `RowCountSum` to the Metadata model and Add-in drop down for `CustomAttributes`

Metadata management

Updated BimlFlex metadata configuration, BimlFlex now has separate management of configurations and settings in two separate sheets within the BimlFlex Excel based metadata editor. The BimlFlex Excel metadata management plugin is now more lightweight with a performance optimized footprint. Schema preview is available in BimlStudio.

Custom SQL Objects and queries

Added `CreateSql` and `OverrideSql` functionality to both Metadata and Extension Points to support custom SQL DDL object creation and custom SQL source queries

Quick Parse

A `Quick Parse` option for BimlStudio has been added as an options to only load placeholder objects from metadata when modelling. This allows faster modelling in BimlFlex and uses less resources for representing packages in BimlStudio. Once Packages needs to be built, disable the `Quick Parse` setting and refresh the metadata to have BimlStudio populated with full fidelity objects.

Split `Use My ConnectionStrings` and `Use My Exclusions`

In BimlStudio and BimlFlex Excel Metadata editor it is now possible to split `Use My ConnectionStrings` and `Use My Exclusions`. This allows a more granular control over user based settings in team based development scenarios.

Added SsisAnnotations

All generated SSIS packages now have added `SsisAnnotations`

Release Notes

BimlStudio 2018.1 Note

Warning

BimlStudio 2018 can deploy and update BimlFlex and BimlCatalog databases for SQL Server up to 2016. For customers with databases on SQL Server 2017 we currently recommend deploying and upgrading through the BimlFlex Utility Application. Please send an email to bimlflex-support@varigence.com to get a copy.

Bundle 62925

- Add: PSA Merge SQL Statement now raises an error on transaction rollback for audit and logging
- Update: Lookup join from DV Satellites and Link Satellites now filter better on existing records
- Add: SSIS Delay Validation on Connections

Bundle 62907

- Added support for `ParentProject` targets in `CustomAttributes`
- Updated BimlFlex Utility to update new local application data folder location for template Bundle
- Added Batch for re-initialization of all PSA to Staging packages
- Added Validator for Self-referencing Business Keys
- Added validator for BimlCatalog connection for `ADONET` and `MSSQL/SQL Server` properties
- Added support for SQL Server 2017 for custom Script components
- Added initial support for integrated source Delete detection

Bundle 62901

- Added support for `ExcludeFromBuild` objects when using the Data Vault Accelerator
- Updated, BimlFlex now exclude creation of external tables when `ExcludeFromBuild` is set to `Y`
- Added `ObjectInherit` support for BatchVariable Extension Point
- Added `LookupJoin` Extension Point for `WHERE` clause inclusion

Bundle 62823

- Added logic to convert `TEXT` and `NTEXT` Source columns to corresponding `VARCHAR` for Blob storage loads to accommodate PolyBase
- Added Validator to ensure Target Dimensions has a separate Business Key and Primary Key defined
- Added a `CDATA` wrapper to comments and descriptions to accommodate xml control characters. In certain scenarios some characters would introduce issues in the BimlStudio parsing of contents.

Bundle 62817

- Updates to Satellite record compression to correctly identify initial rows. In certain scenarios an additional row was being created in Satellite loads. Record compression now works across these scenarios.
- Updated a configuration scenario where PSA updates were incorrectly identified as having the `IsCurrent` flag present for end dating even if it was switched off in the Configuration.

Bundle 62809

- Bundle version number realignment
- Added integrated Support for COZYROC Salesforce source SSIS custom component
- Added support for Hash Distribution Keys in Inferred Hubs for Azure SQL Data Warehouse optimization.

- Added Parent Batch requirement for nested projects. Existing Projects with a master Project without a Batch specified will need to be updated to include the main Batch.
- Added `FromSql` metadata definition to support additional query logic, including hints like `WITH (NOLOCK)`

Bundle 80329

- Added new feature. Control if the Data Vault Accelerator should create Link Satellites for generated Links. New setting `DvAccelerateLinkSatellite` has been added to the `Settings` sheet. This controls if Link Satellites should be accelerated. Breaking change This feature has a default value of `N`, meaning no Link Satellites will be accelerated unless it is updated to `Y`. Overrides can be created using attribute `SettingValue` definitions for individual source tables.

Warning

The default setting for `DvAccelerateLinkSatellite` is `N`.

Existing projects using Link Satellites should update it to `Y` to maintain behavior

Bundle 80325

- Add: update `CREATE DATABASE` default statements for generated SQL scripts to create databases with `Recovery Model` set to `Simple`
- New: support for global Extension Point for Project Parameters. This Project Parameter will be added to all Projects

Use Extension Point `ProjectParameter` with target `@@global`:

```
<#@ extension bundle="BimlFlex.bimlb" extensionpoint="ProjectParameter" target="@@global" #>
<Parameter Name="YourParameterName" DataType="String" IsRequired="true">DefaultParameterValue</Parameter>
```

Bundle 80321

- Fix: Issue with loading ELT into Data Vault for External Tables
- Update: Split `Use My ConnectionStrings` and `Use My Exclusions` in BimlStudio and BimlFlex Excel Add-in configuration
- New: Add `SsisAnnotations` to all packages
- Update: Change RowCount object name for Data Vault loads to reference destination rather than source object
- Update: Fix behavior for Azure DW ELT not deriving BK columns
- Update: Fix behavior for SSIS Connection Managers for Azure DW connections
- Fix: an issue where `wcf.SetCloneVersion` did not correctly manage `ClonedFromUID`
- Update: Allow SSDT project to build when no tables are defined
- Update: Add warning validation for projects without defined target
- Update: Add support for `AutoAdjustBufferSize` SSIS setting for supported SSIS versions
- Add: support for SQL Server 2017 custom SSIS components
- Add: support for `SelectBySql` for `DISTINCT, TOP n` scenarios
- Add: support for `OrderBySql` for `ORDER BY Column` scenarios

Bundle 80305

- Fix: Data Vault templates for SQL to load LSAT correctly
- New: Add `Quick Parse` option for BimlStudio options to only load placeholder objects when modelling
- Fix: A scenario where end dating of Satellites did not behave as expected
- Update: Optimization to SQL templates
- New: Add concurrency configuration to `AzCopy`
- New: Refactor Data Vault patterns to exclude end dating code when `EnableEndDateRdv` setting is set to `"N"`

Bundle 80223

- Performance update to BimlFlex to use local cache when no changes to metadata
- Performance update to minify Biml files in cache
- Fix: Remove IsNullable warning for External Tables
- Fix: Remove double underscores from certain name generations

Bundle 80216

- Fix: Orchestration truncate statements now exclude database name in statement
- Fix: Synchronized LNK and LSAT Hashes for ELT loads for on-premises SQL Server
- Fix: Added Execution Id parameter to SSIS calls to Stored Procedures for ELT loads for on-premises SQL Server

Bundle 80213

- New: Use `Page` compression for tables as default
- New: Added velocity profiling for PSA tables
- Fix: Connect PreDataFlow for FlatFile sources
- Fix: Added support to derive LSK columns when needed for external tables

Bundle 80212

- Fix: Update to Same As Link to derive both Hub sides
- Add `CONVERT` to SourceQuery to accommodate SQL Server and Polybase Maximum DMS row size limitations
- Fix: Removed Link Surrogate Keys from External Table definitions
- Added initial support to auto-generate SSDT projects for all SQL artefacts

Bundle 80207

- Change: Change PIT stored procedure to only use SAT FromDate to support insert only Data Vault
- Fix: Adding support for Same As Links in the Data Vault Accelerator
- Fix: Adding support for Azure Blob Storage for Connections to support both Azure SQLDW and Azure SQL VM
- Fix: Satellite Driving Key Lookups and Condensing Script
- Fix: Change BimlFlexUtility to remove temp folder before and after app runs. This is to avoid cached bundle from being deployed.
- Fix: PSA Source query to exclude source `JoinSql` and `GroupBySql`
- Fix: Link Degenerate Keys to map to Link and not LSAT
- Change: Remove dependency to define IsDerived on the Columns Table
- Change: Repoint DB2 import to use the SYSIBM tables instead of the SYSCAT tables.
- Change: BimlCatalog SetConfigVariable to handle additional "NULL" default dates '1899-12-31', '1900-01-01', '0000-01-01'

Bundle 80118

- Update Snapshot capture to database and add-in
- Update add-in to default IsNotPersistent for derived BusinessKeys on Import Metadata
- Change file compression to limit in memory compression to 2GB. Larger files will be compressed using FileStream to accommodate .Net limit
- Change `__Utility` sheet in add-in dynamic. Note this will raise Excel warning "Microsoft Excel will permanently delete this sheet...". Click Delete button and continue.

Bundle 80112

- Added Welcome documentation screen to BimlFlex Utility Application
- Added Release Notes screen to BimlFlex Utility Application

Bundle 80109

- Added Set Parameter for SQL based stored procedures
- Added support for Azure Blob Storage based staging
- Update to optimize Azure SQL Data Warehouse generated code

Bundle 71204

- Added additional logic to handle the `ParameterColumnExpression` with `ExecuteSqlOnSource`
- Update to rdv helper functions for Multi Active Key End Dating
- Update to accommodate a scenario where a Satellite exist with a Link and no Hub. This should not happen, but can if the Data Vault is modelled incorrectly
- Update to `SqlDwSatEndDate` to update `RowIsCurrent`
- Added `BusinessKeyQualifiedName`, `BusinessKeyName`, `PrimaryKeyQualifiedName` and `PrimaryKeyName` to `TableObject`
- Update to PSA parameters for Azure SQL Data Warehouse and Satellite End Dates
- Update to check is initial load before Data Flow Task for dimension and fact SSIS loads
- Add Ordering to the Data Vault Accelerator

Bundle 71116

- Update to End Dating of Multi Active Satellites
- Updates to row condensing script to accommodate change type if existing in data stream
- Refactor Data Mart Azure Data Warehouse Type 1 SQL logic and enable `SolveOrder` in SQL Batch
- Update to Data Type Mappings for `AnsiStringFixedLength` default mapping
- Add `User.DeleteImportFile` to variables passed to file archive script
- Add Parameters to Stored Procedure Data Mart scripts

Bundle 71108

- Update to `RowChangeType` definition in Type 1 Azure SQL Data Warehouse code
- Refactor of `BEGIN CATCH` to re-use error Variables
- Added `DataVaultProvider` class for optimization of Data Vault Azure SQL Data Warehouse code
- Remove `IsInitialLoad` project parameter dependency. `IsInitialLoad` is now automatically set by querying the PSA for existing rows
- Updates to dependencies in the SQL create scripts
- Updates to Create Table Script
- Updates to references in Data Vault Accelerator Publish function
- Internal Replacing of Tables classes with `TableObject` class
- Update to INIT from PSA table name when STG and PSA database is the same database
- Rename rdv flatten files to correct case
- Refactor Azure SQL Data Warehouse `DataMart` class to correctly End Date Type 2 Dimensions
- Added setting to Delete Import File without archiving

Bundle 71027

- Update to Initialize Object name in `StagingTable` and `PersistentStaging` class
- Update to `MultiActiveKey` column transformation `TargetColumn` alias
- Update to configurations for PSA merge statement
- Update to Data Vault Accelerator for duplicate indexes being previewed

Bundle 71025

- Added additional classes for Staging and Persistent Staging tables to simplify creation of Extension Points

Bundle 71023

- Update to `SourceTable` and `TargetTable` classes

Bundle 71020

- Update to LSAT Data Vault Accelerator and SQL Scripts
- Added `SourceTable`, `TargetTable` and `TableConfiguration` classes to simplify Extension Points

Bundle 71016

- Added option to choose between SSIS optimized Hashing or SQL compatible Hashing in the configurations
- Added `[]` delimiters in SSIS to accommodate lookup columns that contain spaces
- Added `sourceIntegrationStage` variable to all Biml templates
- Update to Data Mart Inferred Member support for columns with spaces and when no dimensions are to be inferred
- Update to SSAS placeholder files for adding Analysis services Extension Points

Bundle 71011

- Updated BimlFlex Utility Application to optionally redact Azure settings when exporting metadata
- Added logic for SSIS Expressions to be applied to files prior to data conversion
- Update to Data Vault Accelerator to add references to existing HUB if it already exists
- Update to BimlFlex Utility Application to allow filtering on Customer when creating a metadata export
- Update to BimlFlex Utility Application to allow automatic zipping of metadata export files
- Update to BimlFlex utility Application to always show changes to Connection Strings
- Update to accommodate issue with `ModelOverride` names in lookups and Data Vault Accelerator
- Update to Data Vault Accelerator logic to reuse existing Hubs when `ExcludeFromBuild` is set
- Update to indexes for PSA tables when `MultiActive` keys are used
- Potentially breaking change to SSIS Hashing Custom component. Use hashing compatible with Azure SQL Data Warehouse and other SQL based CHECKSUMS and HASHBYTES functions. When updating to using SQL compatible Hashing, all pre-existing Data Vault surrogate keys will need to be updated and rehashed from the Business Key

Bundle 70918

- Update to accommodate an issue with Duplicate Multi Active keys being created as Unique Keys
- Replaced `\r\n` with `\n` to simplify the Deploy Table Script
- Added logic to exclude any derived columns missing SSIS expressions for SSIS builds.
- Update to the Prefix/Suffix Logic in `GetSourceObjectName`
- Rename `UseColumnModelOverride` to `StageWithColumnModelOverride` in the configuration
- Rename `UseObjectModelOverride` to `StageWithObjectModelOverride` in the configuration

Bundle 70913

- Update to Infer Hub lookup to use correct business key and error column
- Update to Lookup join to better accommodate differences between Sat and Lsat
- Update to Azure SQL Data Warehouse code where Decimal conversion could swap scale and precision
- Update to formatting of Point In Time Stored Procedures
- Update to `ExcludeFromBuild` when using `ObjectInheritance` to include the table for modelling
- Update to precedence constraints for File to Azure SQL Data Warehouse template
- Added support for ODBC parameterized queries as Expressions
- Updates to Object Model Overrides for Staging Layer

Bundle 70912

- Updates to remove the need to emit Staging objects when using PolyBase and Azure SQL Data Warehouse projects
- Added optional `ProtectionLevel` to `CustomAttributes` to enable ability to compile for `EncryptSensitiveWithUserKey`
- Updates to BimlCatalog Orchestration to use Project Name when logging to accommodate duplicate package names
- Added `ApplyLookupFilterRdv` to filter SSIS lookups by joining to the Staging layer. This minimizes memory usage for lookup components. This cross database join functionality requires the databases to be co-located or the tables to be in the same database
- Added Lookup Filter for Surrogate Key Dimensional `LookupSql`
- Added Foreign Key Lookups for dimension surrogate key lookups
- Updates to `RowCount` SSIS Custom Component logic to add aggregate columns for source queries
- Added `RowCountSum` to the Metadata model and Add-in drop down for `CustomAttributes`
- Added validator to check that multiple source columns from a single source table aren't mapped to the same destination column

Bundle 70906

- Updates to Satellite End Dating to use correct table alias

Bundle 70905

- Breaking changes to the following Extension Points:
 - `LookupSql`
 - `LookupJoin`
 - `LookupParameter`
 - `LookupCache`

They now require `sourceTable` and `destinationTable` definitions

- Connection of RDV lookups to `sourceTable`

Bundle 70904

- Added `MakeFileSafe` to accommodate `#` in table names for Oracle sources
- Added multi file import and load to PolyBase load pattern.
- Updates to UTF-8 Conversion
- Removal of 7-zip dependency. Zipping component will now use embedded compression features to gzip files

Bundle 70829

- Updates to BimlFlex configurations, now split into configurations and settings tables
- Updates to Custom SSIS Component signing

Bundle 70824

- Added `CreateSql` and `OverrideSql` functionality to both database and Extension Points to support custom SQL DDL object creation and custom SQL source queries
- BimlFlex Utility Application updated to support optional redaction of connection strings in metadata exports

Bundle 70822

- Update to UTF-8 conversion to support extended characters previously resulting in overflows

Bundle 70821

- Updates to support non-BimlFlex packages added to the BimlStudio project

- Updates to Primary Keys to support Heaps when loading files
- Updates to Mapped Source functionality to only hash mapped and/or derived columns
- Updates to `GetConfigurations` for Persistent Staging during SQL ELT
- Updates to SQL ELT Data Mart code for Azure SQL Data Warehouse
- Updates to Link lookups for Business Keys to match new Link Satellite process
- Additional Source Connection Validator added

Bundle 70818

- Updates to Azure SQL Data Warehouse Hash Distribution keys for Persistent Staging
- Updates to LSAT Driving Key behavior for multiple Keys
- BimlFlex Utility Application updated with redaction of SQL login passwords
- Added Extension Point process for Blob Storage archiving using PowerShell
- Updated BimlFlex Utility Application to support deployment to Azure SQL Database
- Updates to Persistent Staging lookups when Staging and Persistent Staging are co-located in the same database

Bundle 70807

- Updates to SQL Server CDC load functions
- Refactor of Data Mart process to create Type 1 and Type 2 Stored Procedures
- Updated formatting in BimlFlex Utility Application

Bundle 70724 and earlier

BimlFlex general

- Added metadata model validators to the bundle
- Added initial support for toggle-able output of flat Biml and DDL for entire BimlFlex solution on build
- Added `BimlFlexJsonImporter` to import customer metadata.
- Added Json MetadataDump and add DebugUserSettings to allow Varigence BimlFlex Support to simulate a user project
- Fixed an issue with `IsNotPersistent` logic
- Added `BimlCatalogWrapper` project for OpenSource
- Added support for same name for STG/PSA database
- Fixed an issue with Flat File template
- Fixed an issue with `IsInitialLoad` and Add Inferred Hubs from Sat and Links
- Fixed an issue with PSA Merge Alias for PrimaryKey Join
- Fixed an issue for PSA and `IsNotPersistent`

Metadata Management

- Added AllowCustomInput to the DropDowns that should be optional
- Added Delete entities with selection and update mdv procs.
- Fixed an issue with FlexToBk with `@@rs` length and Validators for `ObjectInheritance`
- Added support for multiple Multi-Active keys (MAK keys)
- Added Table CDC Object Types for SQL Server CDC Source objects
- Added modelling Support for Multi Active Satellites and Customer columns to MDV

Infer Hub functionality

- Added functionality to infer loading of Hubs from Link source tables
- Added check for InferHub configuration into SQL procedure creation

Table creation scripts

- Fixed `CreateSql` script to include Schema and PSA where in some instances they would be missing
- Fixed `OverrideSql` check for PSA INIT to ensure that the `OverrideSql` is not applied twice

Data Type management

- Added extended support for some SQL Datatypes such as Geometry and Geography
- Fixed CustomTypes in BimlDataType

Data Vault Accelerator

- Added Custom Type logic to Data Vault Accelerator
- Enable SchemaGraph documentation and Fix DV publish with saved user settings
- Fixed issue with DV Accelerator publish using stale but cached metadata database credentials

Extension Points

- Added `PreSql` and `PostSql` Extension Points. This allows for easier custom definitions for SQL scripts, such as for injecting compression definitions into table create scripts
- Fixed issue with PIT ExtensionPoint for `LagSql`

DataTypeMappings

- Added functionality to map data types into Staging and RDV. This enables expansion of data types to accommodate future source system changes
- Added Stored Procedure `wcf.SetApplyDataTypeMappings` to support DataTypeMappings
- Fixed DataTypeMappings when target columns exist
- Fixed DataType mappings for Source to Stage adding Data Conversion and Error Handling
- Added additional variables and fix data type mappings.

Oracle Rdb support

- Add initial support for Oracle Rdb
- Add IgnoreSchema for Oracle Rdb integration
- Add logic to exclude schema name from source queries for Oracle Rdb

SSIS

- Updates to Custom Components DLL names. Existing installations using legacy names should consider an upgrade to new names.
- Adding check for Sat cache directory and clean up
- Fix PSA INIT load Source thread
- Fixed `SsisExpression` for LSK columns to correctly use `@@rs`
- Added SQL Server CDC source component

Azure SQL Data Warehouse

- Fixed issue with `SqlExpression` for SQLDW

File to Staging

- Added File archiving to File to Stage Template

BimlFlex Utility

Initial Release of separate BimlFlex Support Utility Application

- Has latest database definitions and BimIFlex Bundle files embedded

Supports the following functions

- Deploy BimIFlex metadata database to SQL Server
- Deploy BimICatalog orchestration database to SQL Server
- Copy BimIFlex Bundle file to installation folders. Search for and replace BimIFlex Bundle files in project folders
- Read metadata for a Customer/Version from BimIFlex database and save to file for easy submission to Varigence BimIFlex Support team for issue resolution