Actions

Actions in HVR allows you to define the behavior of replication. Every action has a collection of parameters that provide a finer control of the behavior. To initiate data replication, at least two actions **Capture** and **Integrate** must be defined on source and target locations accordingly.

This section describes all HVR actions and their parameters.

- AdaptDDL
- AgentPlugin
- Capture
- CollisionDetect
- ColumnProperties
- DbObjectGeneration
- DbSequence
- Environment
- FileFormat
- Integrate
- LocationProperties
- Restrict
- Scheduling
- TableProperties
- Transform

Action Reference

Action	Parameter	Argument	Description
AdaptDDL	**/AddTablePattern**	*patt*	Add new tables to channel if they match.
	/IgnoreTablePattern	*patt*	Ignore new tables which match pattern.
	/CaptureSchema	*db_schema*	Database schema for matching tables.
	/IntegrateSchema	*db_schema*	Generate /Schema for target location(s).
	/RefreshOptions	*refr_opts*	Configure options for adapt's refresh of target.
	/OnDropTable	*pol*	Behavior when source table dropped. Default: from channel only.
	/KeepExistingStructure		Preserve old columns in target, and do not reduce data types sizes.
	/KeepOldRows		Preserve old rows in target during recreate.
AgentPlugin	**/Command**	*path*	Call OS command during replication jobs.
	/DbProc	*dbproc*	Call database procedure *dbproc* during replication jobs.
	/UserArgument	*str*	Pass argument *str* to each agent execution.
	/ExecOnHub		Execute agent on hub instead of location's machine.
	/Order	*int*	Specify order of agent execution.
	/Path	*dir*	Search directory *dir* for agent.
	/Context	*context*	Action only applies if Refresh/Compare *context* matches.
Capture	**/IgnoreSessionName**	*sess_name*	Capture changes directly from DBMS logging system.
	/Coalesce		Coalesce consecutive changes on the same row into a single change.
	/NoBeforeUpdate		Only capture the new values for updated rows.
	/NoTruncate		Do not capture truncate table statements.
	/SupplementalLogging	*action*	Mechanism used to enable supplemental logging for SQL Server tables.

/LogReadMethod	*method*	Method of reading SQL Server's transaction log.
/LogTruncate	*action*	Specify who advances transaction log truncation point.
/AugmentIncomplete	*col_type*	Capture job must select for column values. Can be **NONE**, **LOB** or **ALL**.
/ArchiveLogPath	*dir*	Read archives from an alternative directory.
/ArchiveLogFormat	*format*	Format of base filename of archive files in directory **/ArchiveLogPath**.
/ArchiveLogOnly		Capture data from archives only. Do not read from online redos.
/XLogDirectory	*dir*	Directory containing current PostgreSQL xlog files.
/LogJournal	*schema.journal*	Specify DB2-for-i journal.
/LogJournalSysSeq		Capture from journal using ***SYSSEQ**.
/CheckpointFrequency	*secs*	Checkpointing frequency in seconds for long running transactions, so the capture job can recover quickly when it restarts.
/CheckpointStorage	*STOR*	Storage location of capture checkpoint files for quick capture recovery.
/CheckpointRetention	*period*	Retains capture checkpoint files up to the specified *period*.
/TriggerBased		Capture changes through generated DMBS triggers.
/QuickToggle		Avoid shared lock on toggle table.
/ToggleFrequency	*secs*	Sleep between toggles instead of waiting for database alert (in seconds).
/KeyOnlyCaptureTable		Only keep keys in capture table; outer join others later.
/IgnoreCondition	*sql_expr*	Ignore changes that satisfy expression.
/IgnoreUpdateCondition	*sql_expr*	Ignore update changes that satisfy expression.
/HashBuckets	*int*	Hash structure to improve parallelism of captured tables.
/HashKey	*col_list*	Hash capture table on specific key columns.
/DeleteAfterCapture		Delete file after capture, instead of capturing recently changed files.
/Pattern	*pattern*	Only capture files whose names match *pattern*.
/IgnorePattern	*pattern*	Ignore files whose names match *pattern*.
/IgnoreUnterminated	*pattern*	Ignore files whose last line does not match *pattern*.

	/IgnoreSizeChanges		Changes in file size during capture is not considered an error.
	/AccessDelay	*secs*	Delay read for *secs* seconds to ensure writing is complete.
	/UseDirectoryTime		Check timestamp of parent dir, as Windows move doesn't change mod-time.
CollisionDetect	**/TreatCollisionAsError**		Do not resolve collisions automatically.
	/TimestampColumn	*col_name*	Exploit timestamp column *col_name* for collision detection.
	/AutoHistoryPurge		Delete history table row when no longer needed for collision detection.
	/DetectDuringRefresh	*colname*	During row–wise refresh, discard updates if target timestamp is newer.
	/Context	*context*	Action only applies if Refresh/Compare *context* matches.
ColumnProperties	**/Name**	*col_name*	Name of column in **hvr_column** catalog.
	/DatatypeMatch	*data_type*	Data type used for matching instead of **/Name**.
	/BaseName	*col_name*	Database column name differs from **hvr_column** catalog.
	/Extra		Column exists in base table but not in **hvr_column** catalog.
	/Absent		Column does not exist in base table.
	/CaptureExpression	*sql_expr*	SQL expression for column value when capturing or reading.
	/CaptureExpressionType		Type of mechanism used by HVR capture, refresh and compare job to evaluate value in parameter **/CaptureExpression**.
	/IntegrateExpression	*sql_expr*	SQL expression for column value when integrating.
	/ExpressionScope	*expr_scope*	Operation scope for expressions, e.g. **INSERT**, **DELETE** or **UPDATE_AFTER**.
	/CaptureFromRowId		Capture values from table's DBMS row-id.
	/TrimDatatype	*int*	Reduce width of data type when selecting or capturing changes.
	/Key		Add column to table's replication key.
	/SurrogateKey		Use column instead of the regular key during replication.
	/DistributionKey		Distribution key column.
	/PartitionKeyOrder	*int*	Define the column as a partition key and set partitioning order for the column.

	/SoftDelete		Convert deletes to update of this column to 1. Value 0 means not deleted.
	/TimeKey		Convert all changes to inserts, using this column for time dimension.
	/IgnoreDuringCompare		Ignore values in column during compare and refresh.
	/Datatype	*data_type*	Data type in database if it differs from **hvr_column** catalog.
	/Length	*int*	String length in db if it differs from length in catalog.
	/Precision	*int*	Precision in db if it differs from precision in catalog.
	/Scale	*int*	Integer scale in db if it differs from scale in catalog.
	/Nullable		Nullability in db if it differs from nullability in catalog.
	/Identity		Column has SQL Server identity attribute.
	/Context	*ctx*	Ignore action unless refresh/compare context *ctx* is enabled.
DbObjectGeneration	/NoCaptureInsertTrigger		Inhibit generation of capture insert trigger.
	/NoCaptureUpdateTrigger		Inhibit generation of capture update trigger.
	/NoCaptureDeleteTrigger		Inhibit generation of capture delete trigger.
	/NoCaptureDbProc		Inhibit generation of capture database procedures.
	/NoCaptureTable		Inhibit generation of capture tables.
	/NoIntegrateDbProc		Inhibit generation of integrate database procedures.
	/IncludeSqlFile	*file*	Search directory for include SQL file.
	/IncludeSqlDirectory	*dir*	Search directory for include SQL file.
	/BurstTableStorage		Storage for integrate burst table creation statement.
	/RefreshTableStorage		Storage for base table creation statement during refresh.
	/CaptureTableCreateClause	*sql_expr*	Clause for trigger-based capture table creation statement.
	/StateTableCreateClause	*sql_expr*	Clause for state table creation statement.
	/BurstTableCreateClause	*sql_expr*	Clause for integrate burst table creation statement.

	/FailTableCreateClause	*sql_expr*	Clause for fail table creation statement.
	/HistoryTableCreateClause	*sql_expr*	Clause for history table creation statement.
	/RefreshTableCreateClause	*sql_expr*	Clause for base table creation statement during refresh.
	/RefreshTableGrant		Executes a grant statement on the base table created during **HVR Refresh**.
DbSequence	**/CaptureOnly**		Only capture DB sequences, do not integrate them.
	/IntegrateOnly		Only integrate DB sequences, do not capture them.
	/Name	*seq_name*	Name of database sequence in HVR catalogs.
	/Schema	*db_schema*	Schema which owns DB sequence.
	/BaseName	*seq_name*	Name of sequence in DB if it differs from name in HVR.
Environment	**/Name**	*name*	Name of environment variable.
	/Value	*value*	Value of environment variable.
FileFormat	**/Xml**		Transform rows form/into xml-files.
	/Csv		Transforms rows from/into csv files.
	/Avro		Transforms rows into Apache AVRO format. Integrate only.
	/JSON		Transforms rows into JSON format. The content of the file depends on the value for parameter /JsonMode. This parameter only has an effect on the integrate location.
	/Parquet		Read and write files as Parquet format.
	/Compact		Write compact XML tags like <r> & <c> instead of <row> & <column>.
	/Compress	*algorithm*	Compress/uncompress while writing/reading. *algorithm* is **GZIP** or **LZ4**.
	/Encoding	*encoding*	Encoding of file.
	/HeaderLine		First line of file contains column names.
	/FieldSeparator	*str_esc*	Field separator. Defaults to comma (,). Examples:, \\x1f or \\t
	/LineSeparator	*str_esc*	Line separator. Defaults to newline (\\n). Examples: ;\\n or \r\\n
	/QuoteCharacter	*str_esc*	Character to quote a field with, if the fields contains separators. Defaults to quote (\").
	/EscapeCharacter	*str_esc*	Character to escape the quote character with. Defaults to quote (\").

	/FileTerminator	*str_esc*	File termination at end-of-file. Example: EOF or \xff
	/NullRepresent ation	*esc_str*	String representation for columns with NULL value.
	/AvroCompress ion	*codec*	Avro compression codec. Value should be **Deflate** .
	/AvroVersion	*version*	Version of Apache AVRO format. Possible values are **v1_6**, **v1_7** and **v1_8** (the default).
	/JsonMode	*mode*	Style used to write row into JSON format.
	/PageSize		Parquet page size in bytes.
	/RowGroupThr eshold		Maximum row group size in bytes for Parquet.
	/ParquetVersion	*version*	Category of data types to represent complex data into Parquet format.
	/ConvertNewlin esTo	*style*	Write files with UNIX or DOS style newlines.
	/CaptureConve rter	*path*	Run files through converter before reading.
	/CaptureConve rterArguments	*userarg*	Arguments to the capture converter.
	/IntegrateConv erter	*path*	Run files through converter after writing.
	/IntegrateConv erterArguments	*userarg*	Arguments to the integrate converter program.
	/Context	*context*	Action only applies if Refresh/Compare context matches.
Integrate	/Burst		Resort changes, load into staging table and apply with set-wise SQL.
	/BurstCommitF requency	*freq*	Frequency of commits. Values **STATEMENT**, **TA BLE** or **CYCLE**.
	/Coalesce		Coalesce consecutive changes on the same row into a single change.
	/ReorderRows	*mode*	Control order in which changes are written to files. Values **NONE**, **BATCH_BY_TABLE**, **ORDE R_BY_TABLE** or **SORT_COALESCE**.
	/Resilient	*mode*	Resilient integrate for inserts, updates and deletes. Values **WARNING** or **SILENT**.
	/OnErrorSaveF ailed		Write failed row to fail table.
	/DbProc		Apply changes by calling integrate database procedures.
	/TxBundleSize	*int*	Bundle small transactions for improved performance.

	/TxSplitLimit	*int*	Split very large transactions to limit resource usage.
	/NoTriggerFiring		Enable/Disable triggering of database rules.
	/SessionName	*sess_name*	Integrate changes with special session name *sess_name*.
	/Topic	*expression*	Name of the Kafka topic. You can use strings /text or expressions as Kafka topic name.
	/MessageBundling	*mode*	Number of messages written into single Kafka message. Kafka message contains one row by default.
	/MessageBundlingThreshold	*int*	The threshold for bundling rows in a Kafka message. The default value is 800,000 bytes.
	/MessageKey	*expression*	Expression to generate user defined key in a Kafka message.
	/RenameExpression	*expression*	Expression to name new files, containing brace substitutions.
	/ComparePattern	*patt*	Perform direct file compare.
	/ErrorOnOverwrite		Error if a new file has same name as an existing file.
	/MaxFileSize	*size*	Limit each XML file to *size* bytes.
	/Verbose		Report name of each file integrated.
	/TableName	*apitab*	API name of table to upload attachments into.
	/KeyName	*apikey*	API name of attachment table's key column.
	/CycleByteLimit	*int*	Max amount of routed data (compressed) to process per integrate cycle.
	/JournalRouterFiles		Move processed router files to journal directory on hub.
	/JournalBurstTable		Keep track of changes in the burst table during.
	/Delay	*N*	Delay integration of changes for *N* seconds.
	/Context	*ctx*	Action only applies if Refresh/Compare context matches.
LocationProperties	**/SslRemoteCertificate**	*file*	Enable SSL encryption to remote location; verify location with certificate.
	/SslLocalCertificateKeyPair	*path*	Enable SSL encryption to remote location; identify with certificate/key.
	/ThrottleKbytes	*kbytes*	Restrain net bandwidth into packets of *kbytes* bytes.
	/ThrottleMillisecs	*msecs*	Restrain net bandwidth by *msecs* second(s) wait between packets.
	/Proxy	*proxy*	Proxy server URL for FTP, SFTP, WebDAV or Salesforce locations.

	/Order	*N*	Specify order of hub->loc proxy chain.
	/StateDirectory	*path*	Directory for file location state files. Defaults to <top>/_hvr_state.
	/IntermediateDirectory	*dir*	Directory for storing 'intermediate files' that are g enerated during compare.
	/CaseSensitiveNames		DBMS table and columns names are treated case sensitive by HVR.
	/StagingDirectoryHvr	*URL*	Directory for bulk load staging files.
	/StagingDirectoryDb	*URL*	Location for the bulk load staging files visible from the Database.
	/StagingDirectoryCredentials	*credentials*	Credentials to be used for S3 authentication during RedShift bulk load.
	/S3Encryption	*keyinfo*	Key information to be used for S3 client side encryption.
	/BucketsCount		Number of buckets to be specified while creating a table in Hive ACID.
	/BulkAPI		Use Salesforce Bulk API (instead of the SOAP interface).
	/SerialMode		Force serial mode instead of parallel processing for Bulk API.
	/CloudLicense		Location runs on cloud node with on-demand licensing, for example in Amazon or Azure Marketplace.
Restrict	/CaptureCondition	*sql_expr*	Restrict during capture.
	/IntegrateCondition	*sql_expr*	Restrict during integration.
	/RefreshCondition	*sql_expr*	Restrict during refresh and compare.
	/CompareCondition	*sql_expr*	Restrict during compare.
	/HorizColumn	*col_name*	Horizontal partition table based on value in *col_name*.
	/HorizLookupTable	*tbl_name*	Join partition column with horizontal lookup table.
	/DynamicHorizLookup		Changes to lookup table also trigger replication.
	/AddressTo	*addr*	Only send changes to locations specified by address.
	/AddressSubscribe	*addr*	Get copy of any changes sent to matching address.
	/SelectDistinct		Filter duplicate records during refresh/compare.

	/Context	*ctx*	Action only applies if Refresh/Compare context matches.
Scheduling	/CaptureStartTimes	*times*	Trigger capture job at specific times, rather than continuous cycling.
	/CaptureOnceOnStart		Capture job runs for one cycle after trigger.
	/IntegrateStartAfterCapture		Trigger integrate job only after capture job routes new data.
	/IntegrateStartTimes	*times*	Trigger integrate job at specific times, rather than continuous cycling.
	/IntegrateOnceOnStart		Integrate job runs for one cycle after trigger.
	/RefreshStartTimes	*times*	Trigger refresh job at specific times.
	/CompareStartTimes	*crono*	Trigger compare job at specific times.
	/StatsMetrics		Set of metrics that **hvrstats** job must gather
	/StatsGranularity		Time granularity based on which the **hvrstats** job gathers metrics
	/StatsHistory	*size*	Size of history maintained by **hvrstats** job, before it purges its own rows.
	/StatsCycleDelay		Time delay between metric gather cycle.
TableProperties	/BaseName	*tbl_name*	Name of table in database differs from name in catalogs.
	/Absent		Exclude table (which is available in the channel) from being replicated/integrated into target.
	/DuplicateRows		Table has duplicate rows and no unique key.
	/Schema	*schema*	Database schema which owns table.
	/IgnoreCoerceError		Coerce illegal/big/small values to empty/max/min.
	/CoerceErrorPolicy		Defines a policy to handle type coercion error.
	/CoerceErrorType		Defines which types of coercion errors are affected by **/CoerceErrorPolicy**.
	/TrimWhiteSpace		Remove trailing whitespace from **varchar**.
	/TrimTime	*policy*	Trim time when converting from Oracle and SQL Server date.
	/MapEmptyStringToSpace		Convert between empty varchar and Oracle varchar space.
	/MapEmptyDateToConstant	*date*	Convert between constant **date** (dd/mm/yyyy) and Ingres empty date.

	/CreateUnicode Datatypes		On table creation use Unicode data types, e.g. map **varchar** to **nvarchar**.
	/DistributionKe yLimit	*int*	Maximum number of columns in the implicit distribution key.
	/DistributionKe yAvoidPattern	*patt*	Avoid putting given columns in the implicit distribution key.
	/CharacterMap ping	*rules*	Specify the replacement rules for unsupported characters.
	/MapBinary	*policy*	Specify how binary data is represented on the target side.
	/MissingRepres entationString	*str*	Inserts value *str* into the string data type column(s) if value is missing/empty in the respective column(s) during integration.
	/MissingRepres entationNumeric	*str*	Inserts value *str* into the numeric data type column(s) if value is missing/empty in the respective column(s) during integration.
	/MissingRepres entationDate	*str*	Inserts value *str* into the date data type column(s) if value is missing/empty in the respective column(s) during integration.
Transform	**/Command**	*path*	Path to script or executable performing custom transformation.
	/CommandArg uments	*userarg*	Value(s) of parameter(s) for transform (space separated).
	/SapAugment		Capture job selecting for de-clustering of multi-row SAP cluster tables.
	/SapXForm		Invoke SAP transformation for SAP pool and cluster tables.
	/UnpackTables		Transform will map *_**pack** tables into *_*_**unpac k*** tables.
	/ExecOnHub		Execute transform on hub instead of location's machine.
	/Parallel	*n*	Distribute rows to multiple transformation processes.
	/Context	*context*	Action only applies if Refresh/Compare *context* matches.