WebFOCUS
Server New Features
WebFOCUS Reporting Server Release 82 DataMigrator Server
Release 7707
Kerberos Per User Requirements .......................................................... 57
Using Kerberos for Single Sign-On on Linux ..................................... 57
Upload Support for Validation of File Extensions .............................. 58
Encoding Passwords for a Custom Security Provider ....................... 58
Support for One-Part or Two-Part Names for the Primary Security Provider ... 59
Simplified User Interface for Configuring Directory/File Privileges ......... 60
Registration of Users and Groups for inactive Security Providers ........ 61
Web Console .................................................................................. 61
Setting a Web Session Timeout Warning ......................................... 62
Using the Web Console Binocular Search ......................................... 62
Web Console Text Editor Enhancements .......................................... 64
Displaying the Connected User ID on the Web Console .................... 64

2. Adapter Enhancements ................................................................ 65
   All Adapters .............................................................................. 65
   Changing Common Adapter Settings ......................................... 65
   SQL Adapters .......................................................................... 66
   All SQL Adapters .................................................................... 66
       Optimization of Simplified Numeric Functions ....................... 66
       Optimization of Simplified Functions REPLACE, TOKEN, and POSITION ... 66
       Enhancement to the Optimization of LIKE for Fixed Length Fields .... 66
       PERSISTENCE Option for HOLD FORMAT sqlengine .................. 67
       Optimizing a Heterogeneous Join for a Qualified DBMS ............. 68
       Enhanced BY Clause Optimization ....................................... 68
       Tutorials Now Created With Foreign Key Constraints ............... 68
       Creating Tutorials With Lowercase Column Names ................ 69
       Saving Adapter Settings in a Profile or Stored Procedure ............ 69
       Mapping Spatial Data Stored in RDBMS Columns .................... 69
       Enhanced Messages About SQL Optimization ........................ 71
       Optimization of Function DTRUNC for First Day of Week .......... 71
       Optimization of Function DTRUNC for YEAR_END, QUARTER_END, MONTH_END
       and WEEK_END .................................................................. 71
       CONCAT Function Optimization ............................................. 71
Optimization of the DT_CURRENT_DATE, DT_CURRENT_DATETIME, and DT_CURRENT_TIME Functions ............................................ 72
SQL Adapters Support CREATE FILE With FOREIGNKEYS ........................ 73
Application Access Control for HyperStage Tables ............................. 73
SQL Adapters Optimize Calls to the GIS_POINT Function in a DEFINE ............. 73
Enhanced Optimization of Selection Tests ........................................ 75
Optimization of the PARTITION_REF Function ................................... 75
New Optimization Controls Using SET FEATOPT ................................ 77
New SQL Functions ........................................................................ 79
MOD: Returning the Remainder of a Division ....................................... 80
CEIL: Returning the Smallest Integer Greater Than or Equal to a Value ......... 80
FLOOR: Returning the Largest Integer Less Than or Equal to a Value .......... 81
LEAST: Returning the Smallest Value ............................................. 81
GREATEST: Returning the Largest Value ......................................... 82
SQL Adapters Create Unique Keys for HOLD FORMAT SQL_SCRIPT ............... 83
Enhancement to SQL Adapter Native Messaging Interface ....................... 85
Improved Messaging for Environmental Problems ............................... 85
Random Data Sampling in SQL .................................................... 85
Optimization of Expressions on Partial Dates ...................................... 86
Adapter for Apache Drill ................................................................. 86
Adapter for Db2 ........................................................................ 86
Change to Multi-Row Inserts on z/OS .............................................. 87
Support for TIMESTAMP for CDC_START on z/OS ................................. 87
Conversion to ANSI Date, Time, and Timestamp Literals .......................... 87
Support DECFLOAT Data Type as MATH and XMATH ............................. 87
Adapter for C9 INC ...................................................................... 88
New Setting Available for Temporal Lower Boundary Date ......................... 88
Adapter for Cloudera Impala ............................................................ 88
Adapter for EXASol ODBC ............................................................ 88
Adapter for EXASol ODBC: Extended Bulk Load ................................ 88
Adapter for Google BigQuery .......................................................... 89
Adapter for Hive ........................................................................ 89
Synonym Creation for SQL Strings .................................................. 89
Adapter for Sybase IQ Support for Unicode Extended Bulk Load.................. 94
Adapter for Teradata.............................................................................. 94
Support for Wide ODBC API ................................................................. 95
READONLY Fields in Extended Bulk Load ............................................. 95
Distinguishing Between Macros and Stored Procedures.......................... 95
Support for Version 15.10................................................................. 95

ERP Adapters ............................................................................... 95
Adapter for Microsoft Dynamics CRM................................................. 95
Adapter for SAP/R3........................................................................... 95
Enhanced Function Module Functionality.............................................. 96

GIS Adapters ............................................................................... 96
Adapter for Esri ArcGIS................................................................. 96
Unified Definition for ArcGIS Server URLs and Geographic Roles.............. 97
Configuring Geographic Roles........................................................... 99

OLAP Adapters ........................................................................ 104
Adapter for Essbase.......................................................................... 105
Adapter for Microsoft SQL Server Analysis Services (SSAS).................... 105
Creating Temporary Calculated Measures Using DB_EXPR................... 105
Support for Display Options M, p, and %.......................................... 108
Data Analysis Expression (DAX) Passthrough....................................... 109
Adapter for TM1........................................................................... 110
CAM Authentication Support.............................................................. 110
Integrated Login Support................................................................. 110

DBMS Adapters .......................................................................... 111
Adapter for Salesforce JOIN Support.................................................. 111

Procedures Adapters .................................................................. 111
Adapter for REST........................................................................... 112
Adapter for REST/WS: Pass Name:Value Pairs in the HTTP Header......... 112
Adapter for REST: Support for GZIP Response Format......................... 113
Adapter for REST: Support for Requests not Containing Service URL Extension... 113
Adapter for REST: Support for Token as Attribute in Chained Authentication... 113
Adapter for REST/WS: Type Needed for Elements when OBJ_TYPES=NO... 113
Adapter for REST/WS: Support for Attribute Referencing........................ 113
Adapter for RStat Predictive Analytics ............................................. 113
Adapter for Web Services .......................................................... 113
Microsoft Dynamics AX 2012 Integration ...................................... 113
Sequential and Indexed Adapters ................................................. 114
Reading Virtual Fields Through the File Listener ............................. 114
Special Characters Supported With Upload Wizard .......................... 114
Adapter for Excel (via Direct Retrieval) ......................................... 114
File Listener for Excel Worksheet ............................................. 114
Adapters for DFIX and Excel (via Direct Retrieval) ......................... 114
Date and Date-Time Recognition With Patterns .............................. 114
Adapters for Flat and Delimited Flat Files ..................................... 115
Adapter for DFIX: Respecting Server CDN option ......................... 115
Adapter for RMS ................................................................. 115
Direct 64bit DEC TIMESTAMP Support ..................................... 115
SET INSERTSIZE=n for Insert and Update .................................. 116
Update Locking Behavior ....................................................... 117
Social Media Adapters ............................................................ 117
Adapter for Google Analytics ..................................................... 117
Adapter for Facebook: Enhanced Create Synonym Functionality ............. 118
Social Media Demonstration Application ..................................... 118
XML-Based Adapters .............................................................. 119
Adapter for JSON ............................................................... 119
Adapter for JSON Support for UPDATE ..................................... 119
Support for ON TABLE HOLD FORMAT JSON ............................ 119
Ability to Set Encoding in the XML Header for XML Write ............... 119
Redirection Support for HTTP Status Codes 302, 303, and 307 .......... 119

3. DataMigrator Enhancements .................................................. 121
Adapters ........................................................................ 121
Change Data Capture for MySQL .............................................. 121
Connection for [Delimited] Flat File Targets ................................. 122
Extended Bulk Load Availability ............................................ 122
Extended Bulk Load for Hive and Impala ................................... 122
File Attributes from File Listener .......................................................... 122
Support SFTP Source from a DM Server on Windows .................................. 122
MS SQL Server ODBC Adapter in Linux ................................................... 123
Change Data Capture for MySQL and MariaDB .......................................... 123
Calculators .................................................................................................. 123
Allow a Variable in WHERE Filter for IN .................................................. 123
Notifications for Format Auto Correct ........................................................ 123
Enhanced DMTRUNC Function .................................................................... 123
Optimization of REPLACE Function .......................................................... 123
Using a Regular Expression in a SQL WHERE Condition .............................. 124
Data and Process Flows ............................................................................. 124
CMRUN on All Platforms ........................................................................... 124
Disallowing Multiple Instances of a Flow ................................................... 125
Expand and Collapse Functionality for Objects .......................................... 125
Long Names in Flows .................................................................................. 125
Optimize Load Option to Support MERGE ............................................... 125
Specifying Display Name in Target Properties .......................................... 126
Updates to Email Functionality ................................................................. 126
Updating Values in Target Transformations .............................................. 127
Data Management Console ....................................................................... 127
Downloading Files ...................................................................................... 127
Improvements in Column Name Display Strategy ...................................... 127
Improvements to Text Editor ...................................................................... 127
Kerberos and Single Sign-On From DMC ................................................... 127
Using Find in Properties Panes ................................................................. 128
Data Profiling ............................................................................................. 128
Data Profiling Hex View Report ................................................................. 128
Data Profiling Value Charts Option to Omit Missing Values ............................. 128
Quick Copy .................................................................................................. 128
Quick Copy Allows Delimited Flat File Targets ............................................ 128
Reporting .................................................................................................... 128
Data Lineage Report .................................................................................... 128
Dependencies Analysis Report .................................................................... 129
4. Resource Analyzer and Resource Governor Enhancements .......................... 131
   New Top n Reports ................................................................. 131
   New Field in SMCOLUMNS Table ..................................................... 131
   Converting Existing Repositories to New Repositories ............................... 132
   Vertica Now Supported ........................................................................ 132
   Special Characters Now Supported ..................................................... 132

5. Reporting Language Enhancements ................................................. 133
   Additional Support for Alignment Grid ...................................................... 134
   Defining Internal Borders in an Alignment Grid .......................................... 134
   Enhancements to PPTX Output Format ..................................................... 135
   PNG Charts Embedded in Slides ................................................................. 135
   PPTX Template Masters and Slide Layouts for Expanded Styling ...................... 136
   Support for Page Size and Orientation ..................................................... 137
   ReportCaster Bursting Support ............................................................... 137
   Enhanced PDF Support for JPG Compression ............................................ 137
   Defining Hyperlink Colors ........................................................................ 138
   Automatically Resizing the Width of a Window or Frame With HFREEZE .............. 139
   Enhanced Implementation of Peer Graphs .................................................. 139
      HEX and RGB Color Designations ......................................................... 139
   Setting a Color for Data Visualization Bar Graphs ...................................... 139
   SET FLOATMAPPING: Expanded Numeric Functionality ............................... 139
   New LOCALE Settings ............................................................................. 140
   Applying Selection Criteria to the Internal Matrix Prior to COMPUTE Processing .... 147
   PARTITION_AGGR: Creating Rolling Calculations ....................................... 153
   PARTITION_REF: Using Prior Field Values in Calculations ............................ 163
<table>
<thead>
<tr>
<th>Content</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increased Number of ACROSS Columns in a Report</td>
<td>167</td>
</tr>
<tr>
<td>Increased Number of Verb Objects in a TABLE Request</td>
<td>167</td>
</tr>
<tr>
<td>Specifying a Geographic Role in a DEFINE or COMPUTE</td>
<td>167</td>
</tr>
<tr>
<td>Limiting the Number of Stacked Commands and Loop Iterations</td>
<td>169</td>
</tr>
<tr>
<td>New Report-Level Prefix Operators</td>
<td>170</td>
</tr>
<tr>
<td>Controlling Processing of Delimited Files</td>
<td>173</td>
</tr>
<tr>
<td>Format Display Option for Percentage Calculation</td>
<td>173</td>
</tr>
<tr>
<td>Start of Week Parameter for the DTRUNC Function</td>
<td>174</td>
</tr>
<tr>
<td>Function DTRUNC Returns the Last Day of a Calendar Period</td>
<td>175</td>
</tr>
<tr>
<td>Simplified Numeric Functions</td>
<td>176</td>
</tr>
<tr>
<td>CEILING: Returning the Smallest Integer Value Greater Than or Equal to a Value</td>
<td>177</td>
</tr>
<tr>
<td>EXPONENT: Raising e to a Power</td>
<td>178</td>
</tr>
<tr>
<td>FLOOR: Returning the Largest Integer Less Than or Equal to a Value</td>
<td>179</td>
</tr>
<tr>
<td>MOD: Calculating the Remainder From a Division</td>
<td>181</td>
</tr>
<tr>
<td>POWER: Raising a Value to a Power</td>
<td>182</td>
</tr>
<tr>
<td>Simplified Character Functions</td>
<td>183</td>
</tr>
<tr>
<td>LAST_NONBLANK: Retrieving the Last Field Value That is Neither Blank nor Missing</td>
<td>183</td>
</tr>
<tr>
<td>REGEX: Matching a String to a Regular Expression</td>
<td>185</td>
</tr>
<tr>
<td>CONCAT: Concatenating Strings After Removing Trailing Blanks From the First</td>
<td>187</td>
</tr>
<tr>
<td>REPLACE: Replacing a String</td>
<td>189</td>
</tr>
<tr>
<td>SPLIT: Extracting an Element From a String</td>
<td>192</td>
</tr>
<tr>
<td>CHECKMD5: Computing an MD5 Hash Check Value</td>
<td>193</td>
</tr>
<tr>
<td>CHECKSUM: Computing a Hash Sum</td>
<td>194</td>
</tr>
<tr>
<td>Simplified Conversion Functions</td>
<td>195</td>
</tr>
<tr>
<td>CHAR: Returning a Character Based on a Numeric Code</td>
<td>196</td>
</tr>
<tr>
<td>CTRLCHAR: Returning a Non-Printable Control Character</td>
<td>197</td>
</tr>
<tr>
<td>EDIT2: Converting a Numeric, Date, or Date-Time Value to a Character String</td>
<td>199</td>
</tr>
<tr>
<td>HEXTYPE: Returning the Hexadecimal View of an Input Value</td>
<td>202</td>
</tr>
<tr>
<td>PHONETIC: Returning a Phonetic Key for a String</td>
<td>204</td>
</tr>
<tr>
<td>TO_INTEGER: Converting a Character String to an Integer Value</td>
<td>206</td>
</tr>
<tr>
<td>TO_NUMBER: Converting a Character String to a Numeric Value</td>
<td>207</td>
</tr>
<tr>
<td>Simplified Date Functions</td>
<td>207</td>
</tr>
<tr>
<td>DT_CURRENT_DATE: Returning the Current Date</td>
<td>208</td>
</tr>
</tbody>
</table>
Contents

DT_CURRENT_DATETIME: Returning the Current Date and Time......................... 208
DT_CURRENT_TIME: Returning the Current Time............................................. 209
Simplified System Functions ................................................................. 210
  ENCRYPT: Encrypting a Password..................................................... 210
  GETENV: Retrieving the Value of an Environment Variable....................... 211
  PUTENV: Assigning a Value to an Environment Variable............................ 211
Simplified Geography Functions ............................................................ 212
  Sample Geography Files........................................................................ 213
  GIS_DISTANCE: Calculating the Distance Between Geometry Points............... 218
  GIS_DRIVE_ROUTE: Calculating the Driving Directions Between Geometry Points... 220
  GIS_POINT: Building a Geometry Point.................................................. 224
  GIS_GEOCODE_ADDR: Geocoding a Complete Address................................ 227
  GIS_GEOCODE_ADDR_CITY: Geocoding an Address Line, City, and State......... 228
  GIS_GEOCODE_ADDR_POSTAL: Geocoding an Address Line and Postal Code....... 230
  GIS_GEOMETRY: Building a JSON Geometry Object................................... 231
  GIS_IN_POLYGON: Determining if a Point is in a Complex Polygon............... 235
  GIS_LINE: Building a JSON Line....................................................... 236
  GIS_SERVICE_AREA: Calculating a Geometry Area Around a Given Point......... 240
  GIS_SERV_AREA_XY: Calculating a Service Area Around a Given Coordinate..... 244
Support for the Comma Character as the Operation in an Autoprompt Dynamic Multi-Select
  List........................................................................................................ 248
Automatic Conversion Between Date and Date-Time Formats........................... 249
Chapter 1

Server Enhancements

This section describes new server features on the Web Console.

The server provides a wide range of capabilities and tools for adapter configuration, metadata creation, application and path management, security control, communications configuration, and for monitoring, tuning, and troubleshooting server performance. Authorized users can perform most server administration tasks from a graphical Web Console.

The server supports WebFOCUS reporting functions, extraction, load and transformation functions, and analysis and data access control functions.

In this chapter:

- Applications
- Configuration and Monitoring
- Core Server and Clients
- Diagnostics
- Metadata
- Security
- Web Console

Applications

This section provides detailed descriptions of new features for server applications.

Executing a Repository FOCEXEC From EDASPROF

In the current release, repository procedures can be executed, and Master and Access Files stored in a repository can be accessed, during server, group, and user profile execution after all DBMS connections and an APP MAP command to the repository have been executed.

For information about creating and managing an SQL Repository, see Chapter 3, Managing Applications, in the Server Administration manual.
**Storing Deferred Reports in an Application Directory or an SQL Repository**

Deferred requests and reports can be written to any application, including applications that are stored in an SQL repository.

To route deferred requests to any application:

1. Sign on to the server with a server administrator ID.
2. On the Workspace menu, expand *Special Services and Listeners*.
3. Right-click *SCHEDULER*, and select *Properties* from the shortcut menu.

   The Scheduler Configuration page opens.

   By default, dfm_app is set to a physical directory location.

4. You can accept or edit the dfm_dir location or select an application from the *dfm_app* drop-down list, as shown in the following image.

![Scheduler Configuration](image)

If you select an application, it will override the directory identified in dfm_dir, and the dfm_dir text box will be hidden.

5. Click *Save and Restart Scheduler*.

Deferred requests and output will be returned to that application.

**Unification of Deferred and Schedule Processing**

Deferred (DFM) execution functionality has been extended by unifying major features for deferred execution and scheduler processing. These extended features include the ability to send email notifications on start, completion, and failure of deferred jobs, deriving the email destination from the deferred job, deferred job settings, or the connected User ID, and unified logging. Scheduler and DFM properties have been restructured to reflect the unification of their configuration settings.

The new keywords dfm_email and dfm_email_from_userid were added for DFM jobs to provide a variety of email destinations.
To configure deferred and scheduler properties, right-click **SCHEDULER** in the Special Services and Listeners folder on the Workspace tab, and select **Properties**.

The Scheduler Configuration page opens. The page has Scanning, Execution, Logging and Output, and E-mail Notification sections.

**Reference:** **Scheduler Scanning Properties**

The Scheduler Scanning properties are shown in the following image.

![Scheduler Scanning Properties](image)

**sched_scan_id**

Determines what user ID the Scheduler uses to scan the application path for scheduled flows. It can be used to restrict the set of application directories scanned by the Scheduler to a subset of the APP PATH of the effective administrator. The default value is the effective administrator ID of the server.

**sched_autostart**

If set to `y`, scanning for scheduled events is started when the DFM listener or a Scheduler service starts. If a service is not running, turning on `sched_autostart` will not make scanning available.

If set to `n`, a user may still start the scanning manually.

Depending on the number of files in the application path, `sched_autostart` may affect server performance.

The default value is `n`.

**dfm_autostart**

If set to `y` (the default), the DFM listener/Scheduler Service is started with the server. If it is set to `n` the Service does not start automatically on server start. The user can start the Scheduler manually.

**dfm_dir**

Defines the location where deferred requests and responses are stored. This is ignored if `dfm_app` is set.
dfm_app

Defines the application name where deferred requests and responses are stored.

**Reference:** Scheduler Execution Properties

The Scheduler Execution properties are shown in the following image.

![Scheduler Execution Properties](image)

sched_run_id

Determines the user ID the a flow uses to run scheduled flows, either server_admin_id or user. The default value is server_admin_id.

**Note:** Flows submitted from the Data Management Console, the Web Console, or CMRUN are run under the userid that submits the specific flow.

sched_service

Defines the agent service that the Scheduler will use when running flows. If not set, the DEFAULT service will be used.

sched_restart_interval

Specifies the time interval in seconds between restart attempts for all restartable flows. The default value is 60 seconds.

sched_restart_failed

Determines whether scheduled agents will be restarted after they fail. The valid values are:

- **n** - Failed agents will be restarted at most n times.
- **0** - Failed agents will not be restarted.
- **-1** - Failed agents will be restarted infinitely.

The default value is -1.
CMASAP_clear_interval

 Specifies how often (in seconds) CMASAP agents are removed from the agents list. The default value is -1, meaning never.

Reference: Scheduler Logging and Output Properties

The Scheduler Logging and Output properties are shown in the following image.

<table>
<thead>
<tr>
<th>Logging and Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>sched_log_lines</td>
</tr>
<tr>
<td>sched_standalone_log</td>
</tr>
<tr>
<td>sched_dmlog_app</td>
</tr>
<tr>
<td>sched_log_commit_interval</td>
</tr>
<tr>
<td>sched_log_commit_maxlines</td>
</tr>
<tr>
<td>Merge formatted output into log</td>
</tr>
<tr>
<td>dfm_maxage</td>
</tr>
<tr>
<td>dfm_maxoutput</td>
</tr>
</tbody>
</table>

sched_log_lines

 Defines the maximum number of lines per request that the Scheduler will write to the log. When this number of lines is reached, logging stops. Exceeding this limit for one request does not affect logging for other requests.

When set to -1, the number of lines is unlimited. When set to 0, logging is disabled. The default value is -1.

sched_standalone_log

 If checked, generates a separate log file for each job.

sched_dmlog_app

 Is the name of the application in which the scheduler separate log and statistics files are located.
**sched_log_commit_interval**

Specifies the maximum time interval in seconds that can elapse before log data is committed to the log. Used in conjunction with `sched_log_commit_maxlines`, these settings can fine-tune Scheduler logging behavior. The logging subsystem will commit data to the log whenever one of these maximums has been reached. Finding the best commit interval is a compromise among latency (the delay between the time a log message is received and when it is committed to the log database), reliability, and performance. Longer intervals enable greater I/O efficiency, but increase the potential amount of log information lost in a crash. The default value is 10.

**sched_log_commit_maxlines**

Specifies the maximum number of log lines to collect before committing them to the log. Used in conjunction with `sched_log_commit_interval`, these settings can fine-tune Scheduler logging behavior. The logging subsystem will commit data to the log whenever one of these maximums has been reached. Larger numbers enable greater I/O efficiency, but increase the potential amount of log information lost in a crash. The default value is 1000.

**Merge formatted output into log**

Changes value of `sched_log_output_destination` keyword.

When checked, it specifies that only the job Log is stored in the Scheduler log (`sched_log_output_destination=Log`). The output is stored in the dfm_dir location.

When unchecked, it specifies that the job Log and Output are stored in the Scheduler log (`sched_log_output_destination=LogAndOutput`).

**dfm_maxage**

Defines the maximum number of days that deferred reports are kept in the server after they are created. The value zero (0) means the age is unlimited. The default value is 30.

**dfm_maxoutput**

Defines the maximum size of a deferred report. Reports that over this limit will be removed.

On z/OS PDS deployment, the number represents a number of records. The valid value is any number between 0 and 15,728,688 inclusive.

On all other platforms, the number represents a number of kilobytes or megabytes with the suffix K or M. A valid value is any number between 0 and 65535 inclusive.

The default value is zero (0), which means the output is unlimited.
Reference: Scheduler E-Mail Notification Properties

The Scheduler E-Mail Notification properties are shown in the following image.

On Start
Sends email when the procedure starts.

On Completion
Sends email when the procedure completes.

On Failure
Sends email if the procedure fails.

sched_email_address
Defines the list of email recipients. The list can contain one or multiple email addresses and/or user IDs. Multiple entries should be separated by semicolons (;).

dfm_email
When checked, the scheduler email settings are applied to deferred requests. The default is unchecked, which does not apply the scheduler email settings to deferred requests.

dfm_email_from_userid
When the response of a deferred request is ready to be retrieved, an email notification can be sent to a specific email address. If this keyword is set to n, no email notification will be sent. If it is set to y and there is no email address specified in the deferred request, an email notification will be sent to the email address of the requestor specified in the admin.cfg configuration file or in the corresponding security provider. The default value is n.
**Downloading Files From the Server**

The right-click shortcut menu for a file in an application on the Web Console and Data Management Console now has a Download option, as shown in the following image.

When this option is selected, the file is transferred to the Downloads folder of the user. On Windows, this folder is, by default, the following directory:

*C:\users\username\Downloads*

If you right-click a synonym, the shortcut menu gives you a choice of Master File or Access File.

**Simplified User Interface for Managing the Application Path**

The new User Interface for configuring the application path uses a double list.

You configure the application path from the Applications tab on the Web Console.

1. Click *Application Path* on the ribbon, or right-click the Application Directories tree and select *Application Path* from the shortcut menu.
The Application Path Configuration page opens as a double-list table. The left pane shows a list of all available applications, the right shows all applications in the active Application Path, as shown in the following image.

2. To add an application to the application path, drag it from the Available Applications list to any position on the Application Path list.

You can move applications up or down on the path by dragging them up or down on the list.

3. To remove an application from the application path, drag it from the Application Path list to the Available Applications list.

4. When you have finished, Click Save.

On both lists, you can

- Search for applications using the Binocular Search. For information about the binocular search, see Using the Web Console Binocular Search on page 62.

- Customize the columns displayed, or reset the display to its default, using the View icon.

On the Application Path list, you can:

- Select a different profile.

- Preview the commands in the selected profile.

- Click the Test button to view the APP PATH command and the effective application path.

Renaming Applications and Application Files

Rename has been added as a right-click option for files and folders in the Web Console and Data Management Console.
Selecting Rename from the shortcut menu opens a dialog box in which you can enter a new name and click OK, as shown in the following image.

![Rename dialog box]

You cannot rename mapped applications (including SQL Repository apps), foccache, myhome, homeapps, or baseapp.

**Note:** If you rename an application that is directly under approot, it will no longer be on the application path. You can manually add the renamed application to APP PATH, if you want it to be on the search path of the server.

### Configuration and Monitoring

This section provides detailed descriptions of new configuration and monitoring features.

#### Forcing a Scheduler Scan

A scheduler scan is initiated on server startup, if sched_autostart=y. In addition, the scheduler is updated for a procedure if you edit schedule information for a procedure or DataMigrator job using Manage Schedule and Email from the Web Console or DMC. At other times, you can force scanning of scheduler and deferred jobs in the following ways.

- On the Applications page, click the Schedule/E-Mail button on the ribbon and select *Force Scheduler Scan*.

- On the Workspace page, open the Special Services and Listeners folder, right-click Scheduler, and select *Force Scheduler Scan*.

- Issue the following command in the server command window.

  ```
  edastart -forcescan
  ```
Adding Cc and Bcc to Email Actions

On the Schedule and E-Mail, the Customizing Event Routing, and Customizing New Event Routing pages of the Web Console and Data Management Console, you can add carbon copy (Cc) and blind carbon copy (Bcc) email recipients, as shown in the following image.

You can enter lists of email addresses by separating the addresses with semicolons (;).

Increase in Maximum Length of a Server Node Name

The maximum length of a server node name has been increased from eight characters to 16 characters.

Core Server and Clients

This section provides detailed descriptions of new features for the core server and clients.

Keeping Browser Cache Files Separate for Each Release

The Web Console browser cache files for each server build will be stored separately. This means that multiple Web Console sessions using the same browser will not impact each other. Therefore, you will not have to clear the browser cache to switch between multiple servers using the same machine and browser.

To implement separate cache files, each Web Console URL will automatically be invoked using a unique identifier. For example, if you enter the following URL:

http://mypcname:8021/wc/bcstart.html

The URL will automatically be issued with a unique identifier. For example:
Returning a JSON Answer Set Using the SOAP Listener

Using the Simple Object Access Protocol (SOAP) Listener, you can run report requests or DataMigrator flows using RESTful Web Service calls. The answer set is returned in JSON format.

The following steps describe how to configure and use the SOAP Listener to make the web service calls and return the answer set.

1. First, configure the SOAP Listener.
   - On the Workspace tab, right-click the Special Services and Listeners folder, point to New, and then click SOAP.
   - The Listener Configuration page opens, as shown in the following image.

2. Enter the SOAP Listener port and click Save and Restart Listener.

3. Place the procedure to run in an application directory in the APP PATH of the Reporting Server.
Assuming you created the WebFOCUS Retail tutorial in the ibisamp application, the following FOCEXEC can be created in ibisamp to demonstrate the JSON Answer Set feature. If you created the WebFOCUS Retail tutorial in another application name, substitute application names where applicable.

```
TABLE FILE WF_RETAIL_LITE
    SUM
        COGS_US
    BY BRAND
    BY PRODUCT_CATEGORY
    BY PRODUCT_SUBCATEG
    IF PRODUCT_CATEGORY EQ '&CATEGORY'
    END
```

Note that the request has an IF test that requires a value for the variable &CATEGORY. You will supply the parameter name and value as part of the URL.

4. In a browser window, enter a URL in the following form.

```
http://host:port/rest/app/proc?param=value ...
```

where:

- **host**
  
  Is the Reporting Server host name.

- **port**
  
  Is the SOAP Listener port number.

- **app**
  
  Is the application that contains the procedure. This application must be on the APP PATH of the Reporting Server.

- **proc**
  
  Is the procedure name and extension.

- **param**
  
  Is a parameter name.

- **value**
  
  Is the parameter value.

For example, to run the retail_ws.fex procedure with the parameter &CATEGORY=Accessories, in the ibisamp application, on the host localhost, enter the following URL and press Enter.

```
http://localhost:8050/rest/ibisamp/retail_ws.fex?CATEGORY=Accessories
```
The following answer set is returned in JSON format. In this answer set, after the message header information and report statistics, each row is listed in an array named rows. The array of all rows is enclosed in square brackets ([ ]). Each row is enclosed in curly braces ({ }) and separated from the other rows by a comma (,). The rows consist of *fieldname*: *value* pairs separated by commas. Alphanumeric values are enclosed in double quotation marks (").

```json
{
  "_ibi_Report": {
    "Messages": [
      "0 NUMBER OF RECORDS IN TABLE= 10 LINES= 10"
    ],
    "rows": [
      { "BRAND": "Audio Technica", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Headphones", "COGS_US": 38000.00},
      { "BRAND": "Denon", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Headphones", "COGS_US": 25970.00},
      { "BRAND": "Grado", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Headphones", "COGS_US": 21930.00},
      { "BRAND": "Logitech", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Universal Remote Controls", "COGS_US": 61432.00},
      { "BRAND": "Niles Audio", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Universal Remote Controls", "COGS_US": 73547.00},
      { "BRAND": "Pioneer", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Headphones", "COGS_US": 16720.00},
      { "BRAND": "Samsung", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Charger", "COGS_US": 5405.00},
      { "BRAND": "Sennheiser", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Headphones", "COGS_US": 78113.00},
      { "BRAND": "Sony", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Charger", "COGS_US": 3168.00},
      { "BRAND": "Sony", "PRODUCT_CATEGORY": "Accessories", "PRODUCT_SUBCATEG": "Headphones", "COGS_US": 18592.00}
    ]
  }
}
```

**Automatic Java Lookup on Windows**

The server has an automatic Java lookup feature on Windows to enable the Java Listener to start without any direct server configuration steps.

At server startup time, if no explicit JAVA_HOME or JDK_HOME variables are set either at the system environment level or in the server environment configuration file (EDACONF\bin \edaenv.cfg), the Windows registry will be searched for an applicable Java JDK or JRE release (currently 1.7 or higher) and the appropriate variable will be set.

This feature assumes that a Java installation has been done using the Oracle Windows JRE or JDK installation tool (which sets the Windows registry) prior to running the server (but the server itself requires no configuration).
EDAMAIL Support for Carbon Copy and Blind Carbon Copy

The EDAMAIL command now supports carbon copy (cc) and blind carbon copy (bcc) options.

The syntax is:

```
EDAMAIL to=tadrlist1,
toaddr=tadrlist2,]
[cc=cadrlist2,]
[ccaddr=cadrlist2,]
[ccaddr=bccadrlist,]
[from=fadr1],
[fromaddr=fadr2],
[replyto=radr],
[importance={low|normal|high}],
[subject=string],
[flags={a|A|i}],
[filetype=extension],
[filename=file],
[message=bodymessage]
```

where:

**to=tadr**

Is the email recipient displayed by the user email client (and used in an email client Reply To All, if a toaddr header is not supplied). Multiple addresses are allowed, using a semi-colon as the address separator. A comma (,) is also allowed as a separator, if the overall address string is enclosed in single quotation marks (').

If the to parameter is used in conjunction with toaddr, the value of to may be an arbitrary string, such as "Group Managers", which most email clients will display as a pseudo title in the To field, without displaying the actual addresses used in the toaddr parameter. A forced blank can be supplied for the To field by using to="".

**toaddr=tadrlist2**

Is the email recipient. If not supplied, SMTP servers will use the to header. Email clients will use the toaddr value in an email client Reply To All. Multiple addresses are allowed, using a semi-colon as the address separator. A comma (,) is also allowed as a separator, if the overall address string is enclosed in single quotation marks (').

**cc=cadrlist1**

Is the carbon copy recipient displayed by the user email client (and used in an email client Reply To All, if a ccaddr header is not supplied). Multiple addresses are allowed, using a semi-colon as the address separator. A comma (,) is also allowed as a separator, if the overall address string is enclosed in single quotation marks (').
If the cc parameter is used in conjunction with ccaddr, the value of cc may be an arbitrary string, such as "Group Managers", which most email clients will display as a pseudo title in the Cc field, without displaying the actual addresses used in the ccaddr parameter. A forced blank can be supped for the Cc field by using to="".

ccaddr=cadrlist2

Is the carbon copy recipient. If not supplied, SMTP servers will use the cc header. Email clients will use the cccaddr value in an email client Reply To All. Multiple addresses are allowed, using a semi-colon as the address separator. A comma (,) is also allowed as a separator, if the overall address string is enclosed in single quotation marks (').

bccaddr=bcadrlist

Is the blind carbon copy recipient. Email clients will use the bccaddr value in an email client Reply To All. Multiple addresses are allowed, using a semi-colon as the address separator. A comma (,) is also allowed as a separator, if the overall address string is enclosed in single quotation marks (').

from=fadr1

Is the email sender displayed by the email client (and used in an email client Reply, unless overridden by a fromaddr or Reply To header). If the from parameter is used in conjunction with fromaddr, the from value may be an arbitrary string, such as The Boss, which most email clients will display as a pseudo title in the From field, and will not display the actual address used in the fromaddr parameter.

fromaddr=fadr2

Is the email sender. If not supplied, most email clients will use the From header when doing a reply.

replyto=radr

Is the email sender. If not supplied, most email clients will use the fromaddr or from parameter value.

importance={low|normal|high}

Is the email importance level for email clients that support importance flags. Valid values are high, normal, and low.

subject=string

Is the email subject string. If the subject string contains a comma (,), the string must be enclosed in quotation marks (').
flags=value

If set to a or A, it indicates that a file specified with the filename and filetype parameters is to be sent as an attachment. If set to i, the file is included as the body of the email.

filetype=extension

Defines the data file type for an email message body that uses a file as an attachment. Any application file type or ddname is valid, including MASTER, FOCEXEC, HTML, TEXT, and so on. Leave the parameter out to use the inline email message body feature.

message=bodymessage

Is the inline data stream containing the email message body. If used, it must be the last parameter in the EDAMAIL command. To use the inline data stream feature, the filetype and filename parameters cannot be supplied. The data stream may also be spread onto multiple lines if EDAMAIL is used with the EX-LINES {n|*} feature.

If an inline data stream message body is spread across multiple lines in the procedure, the resulting email is a single line of output. Multi-line message bodies are respected when the message body from a file option is used.

Example:  Mailing an HTML File as a Message Body Using Carbon Copy and Blind Carbon Copy

```
TABLE FILE file1
PRINT A B C
ON TABLE HOLD AS MYFILE FORMAT HTML
END
EX EDAMAIL to=Managers,toaddr=user1@corp1.com;user2@corp1.com,
cc=Subordinates,ccaddr=user3@corp1.com,bccaddr=boss@corp1.com,
from=support1@corp1.com,subject=File1 Report, filetype=HTML,
filename=MYFILE
```

Diagnostics

This section describes new diagnostic features.

Logging All Environment Variables in EDAPRINT

Starting with this release, the top of the EDAPRINT log file contains all environment variables. In addition, if JDK_HOME and JAVA_HOME are both not set, the following warning message displays in EDAPRINT:

```
W JDK_HOME= or JAVA_HOME= required for Java Listener start
```
The following example shows a sample of the top portion of the EDAPRINT log file:

02/24/2016 09:57:11.753 I Workspace initialization started
02/24/2016 09:57:11.753 I EDACONF=C:\ibi\srv99\wfs
02/24/2016 09:57:11.753 I EDAHOME=C:\ibi\srv99\home
02/24/2016 09:57:11.753 I Environment Variables...
02/24/2016 09:57:11.753 I ALLUSERSPROFILE=C:\Documents and Settings\All Users
02/24/2016 09:57:11.753 I APPDATA=C:\Documents and Settings\se02703\Application Data
02/24/2016 09:57:11.753 I CLASSPATH=.\C:\Program Files\Java\jre6\lib\ext\QTJava.zip
02/24/2016 09:57:11.769 I CommonProgramFiles=C:\Program Files/Common Files
02/24/2016 09:57:11.769 I COMPUTERNAME=INFORMAT-31B4D7
02/24/2016 09:57:11.769 I ComSpec=C:\WINDOWS\system32\cmd.exe
02/24/2016 09:57:11.769 I DEFLOGDIR=C:\Documents and Settings\All Users\Application Data\McAfee\DesktopProtection
02/24/2016 09:57:11.769 I EDACONF=C:\ibi\srv99\wfs
02/24/2016 09:57:11.769 I EDAEXTSEC=OFF
02/24/2016 09:57:11.769 I EDAHOME=C:\ibi\srv99\home
02/24/2016 09:57:11.769 I EDAPRINT=C:\ibi\srv99\wfs
02/24/2016 09:57:11.769 I EDASHARE=C:\ibi\srv99\wfs
02/24/2016 09:57:11.769 I EDATEMP0=C:\ibi\srv99\wfs\edatemp
02/24/2016 09:57:11.769 I FP_NO_HOST_CHECK=NO
02/24/2016 09:57:11.769 I HOMEDRIVE=H:
02/24/2016 09:57:11.769 I HOMEPATH=\""

Metadata

This section provides detailed descriptions of new server metadata features.

Business View Plus (BV+) Candidate for Release

Business View Plus (BV+) combines the power and capabilities of Business View (BV) and Dimension View (DV) into one feature that allows a customized view of the data source and enables the use of joins, measures, hierarchies, attributes, expressions, and filters. This allows for robust report, chart, document, and visualization development using a customized logical view of a data source.

A traditional Business View offered users a customized logical view of a data source by grouping related items into folders that reflect business logic for an application, rather than the physical position of items in the data source. However, the fields in these folders did not have any indication of their roles in a request.
A traditional Dimension View, on the other hand, categorized fields on the basis of their roles in a request. Measures were placed in measure groups, hierarchies were organized within dimensions, levels were organized within hierarchies, and attributes were organized within levels. Then, when a field was double-clicked in InfoAssist+ or dragged onto the report or chart canvas in App StudioDeveloper Workbench, it was added as a sort field or aggregation field depending on its placement in the Dimension View structure. Dimension Views, however, offered no ability to create a custom logical view of the data source.

BV+ combines Business Views and Dimension Views by enabling you to group fields into folders and, for each field, assign a role that indicates its role in a request. The syntax is clear and simple, and it provides all of the functions of both traditional Business Views and Dimension Views. In addition, BV+ gives you total flexibility in creating folders anywhere in the structure, and in reusing fields in multiple folders.

For example, if you assign the role DIMENSION to a field, it will automatically be added to the By field container for reports and the horizontal axis for charts if you double-click or drag the field onto the report or chart canvas. If you assign the role Drill Level to successive fields in a folder and turn AUTODRILL on, automatic drilldowns will be generated from the top level to the bottom level on the generated output.

You can create or edit a synonym to use BV+ in the Reporting Server Web Console, the Data Management Console, or the App StudioDeveloper Workbench Metadata Canvas.
Introduction to BV Namespace Modes

When you upload a file or create or open a synonym in the Reporting Server Web Console or Data Management Console, the synonym opens in one of two modes for creating a BV+ structure, BV_NAMESPACE=OFF mode or BV_NAMESPACE=ON mode. The mode in which the synonym opens determines the types of BV+ structures you can add to your synonym. It also controls the type of field name qualifiers that will be used, after the synonym is saved, when you create a request using the WebFOCUSWeb Query tools.

You can set your Web Console preferences to open Data Assist, the wizards, the DMC, and the App StudioDeveloper Workbench Metadata Canvas in your preferred BV Namespace mode by going to the Workspace tab, clicking FOCUS Sets and Info, clicking Settings for Web Console Preferences, and setting the parameter AUTO_BV_NAMESPACE to OFF (the default if you install a new server) or ON. If the synonym already contains a DV or BV structure, the presence of this existing structure will override your setting with BV_NAMESPACE=OFF or BV_NAMESPACE=ON, respectively.

If you create a new synonym, or open an existing synonym that has no DV or BV, it will respect the setting you set in the Web Console. When you upload a file, the synonym has no existing DV or BV structure, so it respects the mode you set in the Web Console.

Reference: BV_NAMESPACE=OFF Mode

BV_NAMESPACE=OFF mode uses physical segment names to qualify field references in reports. If you open an existing synonym that has a DV, it will open in this mode, regardless of the setting you configured. In this mode, you can only add nodes that conform to the DV structure. That is, you can add new measure groups and dimensions under the existing measure group and dimension structure.
The following describes the structures and actions available in new or existing synonyms in BV_NAMESPACE=OFF mode:

Creating a new cluster synonym or opening an existing synonym that has no BV or DV. If the synonym has no logical view defined, it opens with three default nodes in the Business View pane, Filters and Variables, Measure Groups, and Dimensions, as shown in the following image.

These nodes are based on a DV structure, and any edits you make to the synonym in this pane using BV_NAMESPACE=OFF mode must adhere to this structure. The right-click menu options provide only the structures available for a DV. You can insert a new measure group under the Measure Group node or a new dimension folder under the Dimensions node, but you cannot create a new node in the synonym. In a measure group folder, you can only place measures. In a dimensions folder, you can only place dimensions. Under a dimension, you can only add hierarchies, and under a hierarchy field, you can only add attributes.
Opening an existing synonym with a DV defined. It opens showing the DV organization and icons in the Business View pane. The synonym syntax is converted to use BV+ folders with the DV structure. You cannot create new folders, you are limited to the structure already defined in the Business View pane. The right-click menu options provide only the structures available for a DV, as shown in the following image that shows the right-click options for the Measure Group folder.

Reference: BV_NAMESPACE=ON Mode

BV_NAMESPACE=ON mode uses logical folder names to qualify field references in reports. If you open an existing synonym that has a BV, it will open in this mode, regardless of the setting you configured. In this mode, you can create your own logical view of the synonym. You are free to add new folders and assign DV roles to fields and folders.
The following describes the structures and actions available in new or existing synonyms in BV_NAMESPACE=ON mode:

- **Creating a new cluster synonym or opening an existing synonym with no DV or BV.** No nodes are pre-configured, as shown in the following image.

![Image showing a table with various columns such as "ID Sales", "ID Store", etc., and values like 1, 0, 1, etc.]  

The right-click menu options provide all BV+ options.
Opening an existing synonym with a BV defined. It opens in BV_NAMESPACE=ON mode. It will open showing the Business View structure defined within it. The right-click menu options provide BV+ options, as shown in the following image.

Reference: Usage Notes for BV_NAMESPACE Modes

- In BV_NAMESPACE=ON or OFF mode, the folder structure must include all of the fields referenced in the request, as fields not included in folders will not be accessible.

- In BV_NAMESPACE=ON mode, moving a field from one folder to another may also make the field inaccessible, as the request will not have the correct folder qualifier for the field.

- Synonyms in BV_NAMESPACE=OFF mode have the attribute BV_NAMESPACE=OFF on the file declaration.

- You can change BV Namespace modes in the Reporting Server Web Console by clicking the BV menu at the top of the Business View pane and selecting BV_NAMESPACE=ON or BV_NAMESPACE=OFF.

- If you generate a request using the WebFOCUSWeb Query tools in one BV_NAMESPACE mode, it may not run if the synonym is converted to the other BV_NAMESPACE mode, as the field qualifiers will be incorrect for the new mode.
If you change the BV_NAMESPACE mode, a popup window opens to warn you of the potential effects on existing requests.

When you begin the cycle of developing reports and metadata in parallel, use BV_NAMESPACE=OFF mode to make sure that procedures developed earlier in the cycle will continue to run as changes are made to the metadata.

Any changes to metadata should be thoroughly tested before use in production environments.

Assigning DV Roles

In BV+, you define folders, which function as segments to provide a view of the synonym and to define the accessible fields and their relationships. Folder relationships are the same as segment relationships, with parent folders, child folders, and sibling folders.

While you have total flexibility defining a structure using any fields from your data source, when you issue a report request against the synonym, the retrieval path for the data must conform to any constraints imposed by your DBMS entity diagrams and by the rules of WebFOCUSWeb Query retrieval.

Only the folders will be displayed in the WebFOCUSWeb Query tools, not the real segments, and only the fields within the folder structure will be accessible for reporting.
You can assign a DV role to a folder or field by right-clicking the folder or field and selecting a DV role. The following image shows the right-click options for a folder.

You can explicitly assign a DV role to a folder or field, or have it automatically inherit its role from its parent. If you explicitly assign a DV role, that role moves with the object if you drag it to another location within the BV+ structure. If you do not explicitly assign a DV role, the role changes as you move the object under a new parent, except if you drop it onto a field with the Drill Level role. If dropped onto a Drill Level field, the moved field inherits the Drill Level role.

The following DV roles can be assigned.

- **Dimension.** A dimension field, when double-clicked or dragged onto the report or chart canvas in the WebFOCUSWeb Query tools, will automatically be added to the request as a vertical (BY) sort field.

  A folder can be assigned the role Dimension.
A field can be assigned the role Dimension (Standalone) or Dimension (Drill Level). When it is assigned the role Dimension (Drill Level), it will become part of a hierarchy where the levels depend on the order of the fields in the folder. Then, when AUTODRILL is turned on, automatic drill-downs will be created on the report or chart output. The following image shows the choices for DV role when you right-click a dimension field.

For a folder assigned the DV role Dimension or a field assigned the DV role Dimension (Standalone), the following attribute is added to the folder or field declaration in the synonym.

\textbf{DV\_ROLE=DIMENSION}

For a field assigned the DV role Dimension (Drill Level), the following attribute is added to the field declaration in the synonym.

\textbf{DV\_ROLE=LEVEL}

A folder can contain only one drill level hierarchy. However, you can use the same fields in multiple hierarchies by placing each hierarchy in a separate folder. A folder with a drill level hierarchy is not limited to just the hierarchy. It can contain other fields with different DV\_ROLES. 
- **Measure.** A measure field, when double-clicked or dragged onto the report or chart canvas in the WebFOCUS Web Query tools, will automatically be added to the request as an aggregated value (SUM), if it is numeric. If it is alphanumeric, it will be added as a vertical (BY) sort field. A folder or field can be assigned the role Measure.

For a folder or field assigned the DV role Measure, the following attribute is added to the folder or field declaration in the synonym.

```
DV_ROLE=MEASURE
```

- **Attribute.** An attribute field, when double-clicked or dragged onto the report or chart canvas in the WebFOCUS Web Query tools, will automatically be added to the request as an aggregated value (SUM), if it is numeric, or as a vertical sort field (BY), if it is alphanumeric.

For a folder or field assigned the DV role Attribute, the following attribute is added to the folder or field declaration in the synonym.

```
DV_ROLE=ATTRIBUTE
```

- **Folder.** A folder is a virtual segment in a BV+. It can be assigned the roles Dimension, Measure, or Attribute.

**Note:** When a folder is inserted as a child of a field, the attribute PARENT_FIELD describes this relationship. By default, such a folder and its fields will be assumed to have the Attribute role.

- **None.** If no role is assigned, the field or folder will inherit its role from its parent. If a role has been assigned, you can remove it by selecting the option to inherit its role from its parent, as shown in the following image.
Example:  Sample BV+ Declarations

The following declarations show sample BV+ folder and field definitions. Note that the
declaration for each field in a BV+ folder specifies the real segment it actually belongs to.

Sample Dimension Folder Declaration

The DV_ROLE for the PRODUCT_CATEGORY folder is DIMENSION.

```
FOLDER=PRODUCT_CATEGORY, PARENT=FOLDER1,  
   DV_ROLE=DIMENSION,  
   DESCRIPTION='Product and Vendor', $
```

Sample Attributes Folder Declared as a Child of a Field

The ATTRIBUTES1 folder has the DV_ROLE ATTRIBUTE. Its parent folder is
PRODUCT_CATEGORY, and its parent field is MODEL.

```
FOLDER=ATTRIBUTES1, PARENT=PRODUCT_CATEGORY, PARENT_FIELD=MODEL,  
   DV_ROLE=ATTRIBUTE,  
   DESCRIPTION='Model Details', $
```

Sample BV+ Level Hierarchy and Standalone Dimension

The PRODUCT_CATEGORY folder has a hierarchy defined consisting of the
PRODUCTCATEGORY, PRODUCT_SUBCATEG, BRAND, and MODEL fields. The field
VENDOR_NAME is a standalone dimension field.

```
FOLDER=PRODUCT_CATEGORY, PARENT=FOLDER1,  
   DV_ROLE=DIMENSION,  
   DESCRIPTION='Product and Vendor', $
   FIELDNAME=PRODUCT_CATEGORY, ALIAS=PRODUCT_CATEGORY,  
      BELONGS_TO_SEGMENT=WF_RETAIL_PRODUCT,  
      DESCRIPTION='Product Category',  
      DV_ROLE=LEVEL,  $
   FIELDNAME=PRODUCT_SUBCATEG, ALIAS=PRODUCT_SUBCATEG,  
      BELONGS_TO_SEGMENT=WF_RETAIL_PRODUCT,  
      DESCRIPTION='Product Subcategory',  
      DV_ROLE=LEVEL,  $
   FIELDNAME=BRAND, ALIAS=BRAND,  
      BELONGS_TO_SEGMENT=WF_RETAIL_PRODUCT,  
      DESCRIPTION='Product Brand',  
      DV_ROLE=LEVEL,  $
   FIELDNAME=MODEL, ALIAS=MODEL,  
      BELONGS_TO_SEGMENT=WF_RETAIL_PRODUCT,  
      DESCRIPTION='Product Model',  
      DV_ROLE=LEVEL,  $
   FIELDNAME=VENDOR_NAME, ALIAS=VENDOR_NAME,  
      BELONGS_TO_SEGMENT=WF_RETAIL_VENDOR,  
      DESCRIPTION='Vendor Name',  
      DV_ROLE=DIMENSION,  $
```
Sample Measure Field

The PRODUCT_COST field has been assigned the DV_ROLE MEASURE.

FIELDNAME=PRODUCT_COST, ALIAS=PRODUCT_COST,
BELONGS_TO_SEGMENT=WF_RETAIL_PRODUCT,
DESCRIPTION='Product Cost',
DV_ROLE=MEASURE,

Data Assist

Data Assist is a unified environment for managing and enhancing synonyms. It is available on the WebFOCUSWeb Query Reporting Server Web Console and is incorporated in the Upload Wizard.

Data Assist provides an extensive user interface using a ribbon and right-click sensitive paradigm for enhancing synonyms. You see the results of all changes in the tabbed output pane.

The following are some of the features you might want to add to the synonym to enhance your data access and reporting capabilities.

- Add virtual columns (DEFINE fields) and columns for aggregated values (COMPUTE fields).
- Apply functions to fields.
- Grouping, phonetic grouping, and binning field values.
- Add filters to specify data selection criteria.
- Change the format of fields (for example, the size of an alphanumeric field or the format of a date field).
- Create a cluster join view by linking available synonyms to create a multi-segment (multi-table) file for reporting.
- Create Business Views of the metadata in order to limit the fields available to any retrieval request that references the Business View and to group fields together based on their roles in an application.
- Pivot data, for example, pivot repeating columns or groups of repeating columns into rows.
- Define hierarchies for automatic drilldowns.
- Apply security rules for fields and values.

For more information, see the Server Administration manual.
LNGPREP Utility: Preparing Metadata Language Files

The LNGPREP utility extracts TITLE, DESCRIPTION, CAPTION, and PROMPT attribute values from application Master Files into specially formatted language translation files for each language you need. Once you have the contents of these language files translated, your users can run these applications in the language they choose.

LNGPREP does two things. It extracts attribute values from a Master File into language files, and it inserts or updates the TRANS_FILE attribute in the Master File with a value identifying the application folder where the language files reside and a prefix used for naming the set of language files. If the Master File is part of a cluster, LNGPREP will extract translatable strings from every Master File referenced in the cluster, and will update each with the same TRANS_FILE value.

LNGPREP requires an input file listing the three-character codes of the languages you need. The name of each language file starts with the prefix specified in the TRANS_FILE value, followed by a three-character language code, and the extension .lng.

For example, assume the language input file contains the French and Spanish language codes:

fre
spa

If the Master File specifies:

trans_file = xlate/xl_

The language translation files would be in the xlate application folder, named:

xl_fre.lng for French
xl_spa.lng for Spanish

Reference: The Base Language File

Each Master File must have a single base language in which the DESCRIPTION, TITLE, CAPTION, and PROMPT attributes are specified. This language need not be English.

LNGPREP extracts these attribute values into the base language file, whose language code, for historical reasons, is eng. In this case, eng does not mean English. It means whatever language the Master File is written in.

The base language file (prefixeng.lng) should never be hand edited. All other .lng files must be hand edited by translators, to translate the string values from the base language to the appropriate language.
Translating Applications into English

Since language code *eng* is reserved to mean base language, you cannot use it to contain English translations of an application whose base language is not English. In those cases, use any of the other English dialect language codes, such as AME, UKE, CAE, or AUE. For example, if the base language is German, specify AME in the languages file, run LNGPREP, and it will produce `prefixeng.lng` and `prefixame.lng` files, both in German. Translate the contents of `prefixame.lng` into English. Leave `prefixeng.lng` untouched.

Reference: How Translated Master File Attributes Are Displayed

Each language file contains a line for each attribute value from a related set of Master Files. Each attribute value has a unique index number assigned to it. For example, if the Master File contains `FIELDNAME=PRODUCT_CATEGORY`, `TITLE='Product,Category'`, and that `TITLE` happens to be the 39th translatable attribute value, LNGPREP will produce lng files all containing the line:

```
39 = Product,Category
```

Your French translator will edit `prefixfr.lng`, leaving the index values unchanged while translating the string values, producing, in this case,

```
39 = Produit,Catégorie
```

At run time, when the `TITLE` for field `PRODUCT_CATEGORY` needs to be displayed, if WebFOCUS is configured for `LANG=FRE`, WebFOCUS looks up "Product,Category" in `prefixeng.lng`, finds index value 39, looks up 39 in `prefixfr.lng`, and displays the `TITLE` as "Produit,Catégorie."

LNGPREP Modes

You can run LNGPREP from the Web Console using the Prepare Translation Files option, or you can run it using syntax. In either case, you must first create a configuration file containing the three-character language codes for each translation file you need, one language code on each line. The first invocation of LNGPREP for a given Master File adds the TRANS_FILE attribute in that and all related Master Files, creates the base language file by scanning the Master Files for supported attribute values, and creates a copy of the base language file with the correct name for each additional language. Then, a translator has to translate the values in each additional language file from the base language to the correct language for that file.

On each subsequent run, LNGPREP will check for updates to the list of related Master Files and attribute values and update the files as needed. Translators will then have to translate any attribute values added to the language files.
Reference: LNGPREP Best Practice

The recommended best practice is to create an app directory solely for the purpose of containing .lng files, and use this appname and a common prefix value for all LNGPREP commands. In addition, put the languages fn.cfg file in this app folder. This will create one set of .lng files for all apps, minimizing the time and effort spent on translation.

Procedure: How to Prepare Metadata Language Files Using the Web Console

1. Right-click a synonym, select Metadata Management, then Prepare Translation Files, as shown in the following image.

The Set Translation Files page opens, as shown in the following image.

Set Translation Files for ibisamp/ggsales.mas

Application for Translation Files: ibisamp
Prefix:
Languages File:

OK Cancel

2. Enter the following values or accept the defaults.

**Application for Translation Files**

Is the name of the application where the language files will be stored. You can click the ellipsis to select an application from the current application path. By default, it is the application where the synonym resides.

**Prefix**

Is the prefix value for the translation files for the selected synonym.
Languages File
Is the file containing the list of language codes for which translation files should be prepared. The file must have the extension .cfg, be stored in an application directory on the application path, and have one language code on each line. You can click the ellipsis to select the application where the languages file is stored.

3. Click OK.

The language files are prepared using the application, prefix, and languages configuration file you specified. A status page will open listing the language files created and the Master Files processed.

Syntax: How to Run the LNGPREP Command Using Syntax

LNGPREP FILE n_part_name LNGAPP appname LNGPREFIX prefix LNGFILE appname/fn

where:

n_part_name
  Specifies the n-part (app1/app2...) name of a Master File.

appname
  Specifies the location where .lng files will be written and updated.

prefix
  Specifies the literal characters that will precede the three-character language code in the names of the .lng files.

appname/fn
  Specifies the appname and filename of a user-created .cfg file containing the list of three-character language codes, one per line. For example, the following file named langretail.cfg contains language codes for American English, French, and Japanese:

  ame
  fre
  jpn

Example: Sample LNGPREP Command

Assume the lnglist.cfg file contains the language codes fre (French) and spa (Spanish):

  fre
  spa

Issue the following LNGPREP command:

LNGPREP FILE weather/forecast LNGAPP xlate LNGPREFIX tq_ LNGFILE xlate/lnglist
Alternately, you can right-click the forecast synonym, point to Metadata Management, and select Prepare Translation Files. The Set Translation File for weather/forecast.mas page opens. Enter the values shown in the following image and click OK.

The following language files will be created:

- xlate/tq_eng.lng
- xlate/tq_fre.lng
- xlate/tq_spa.lng

The Master File weather/forecast.mas will be updated with the following attribute

TRANS_FILE= xlate/tq_

Translators then have to translate the values in xlate/tq_fre.lng and xlate/tq_spa.lng.

**Security**

This section provides detailed descriptions of new security features.

**Encrypting the admin.cfg File**

The admin.cfg file contains the users and groups registered for the internal server security (PTH). To encrypt this file, do the following.

1. Connect to the Web Console as a Server Administrator.
2. On the Access Control page ribbon, click Settings and select Encryption Settings.
3. Select y from the encrypt_admincfg drop-down list.
4. Click Apply and Restart Server.

When the server restarts, admin.cfg will be encrypted.
**Improved User Interface for Configuring an LDAP Security Provider**

The user interface for configuring an LDAP security provider has been redesigned. The configuration parameters have been grouped into categories, and there are new buttons for retrieving sample lists of users and groups.

1. To configure a new LDAP provider, on the Access Control page, right-click **LDAP** under Security Providers and select **New** from the shortcut menu.

   The following LDAP Security Provider Configuration page opens on which you configure the provider name and connection parameters.

   **LDAP Security Provider Configuration**

   - **LDAP_PROVIDER**
     - Value: ldap02

   - **Connection**
     - **ldap_host**
       - Value: Sample: medsun
     - **ldap_secure_connection**
       - Value: No
     - **ldap_port**
       - Value: 389
     - **security**
       - Value: Anonymous or Windows security - NEGOTIATE
     - **ldap_search_timeout**
       - Value: 60

2. Configure the following connection parameters.

   **LDAP_PROVIDER**

   Specifies a name for this provider.

   **ldap_host**

   Is a host identifier consisting of a host name or an IPv4 dotted string representing the IP address of a host running the LDAP server to connect to.

   Alternatively, it may contain a list of space-delimited host identifiers. Each host identifier may include a trailing colon and port number. In the case where more than one host identifier is specified, each host identifier in turn will be contacted until a connection can be established. For example:

   ```
directory.example.com
192.0.2.0
directory.example.com:1050 people.catalog.com 192.0.2.0
```
**ldap_secure_connection**

Specifies whether the server uses a *Secure Socket Layer (SSL) session with the LDAP server*. Select No or Yes. The server default is No.

An LDAP (Lightweight Directory Access Protocol) security provider supports Secure Sockets Layer (SSL) API calls to establish an SSL/TLS connection. Using server authentication only, the Reporting Server initiates API calls to verify that the LDAP server being connected to is the same server that provided certification.

You can set the LDAP secure connection from the Web Console:

- Select No, the default value, if you do not wish to enable SSL.
- Select Yes to enable an encrypted Secure Sockets Layer (SSL) session with the LDAP server.

If you have selected IBM, Sun, or Novell as your `ldap_lib_vendor`, when you select Yes in the `ldap_secure_connection` field, additional options are added to the Connection tab:

- For Sun and IBM, `ldap_ssl_certificate` is added.
- For Novell, `ldap_ssl_certificate` and `ldap_ssl_certification_encoding` are added.

**ldap_ssl_certificate.** Enter the name of the LDAP attribute used by the API to establish the SSL/TLS connection. The server employs server authentication only, checking through API calls that the LDAP server you are connecting to is the one that provided the certificate. Values depend on the LDAP vendor, as follows:

- **Novell API** specifies the file name, including path, of the Trusted Root Certificate that the LDAP server provided for authentication.
- **Sun/Netscape API** specifies the path to cert7.db, Netscape certificate database, excluding the file name, that the LDAP server provided for authentication.
- **IBM API** specifies the file name, including the path, for ldapkey.kdb (IBM key database file that the LDAP server provided for authentication). The ldapkey.sth password stash file should be in the same directory. Note that in addition to IBM LDAP client libraries, Global Security Kit libraries are needed to make SSL work. On Windows machines, GSK must be installed.
- **Microsoft API** ignores the `ldap_ssl_certificate` configuration parameter since it is not used in an Active Directory. The server certificate should be installed in a certificate store.
**ldap_ssl_certificate_encoding.** For Novell, select the standard used to encode the certificate from the drop-down list. Encryption and file format depend on API vendor specifications. The options are B64 and DER.

**ldap_port**

Is a positive integer that defines the TCP port number used to connect to the LDAP server. Note that ldap_port is ignored for any host identifier which includes a colon and port number. The server default port is 389 or 636 (for SSL connection).

**security**

Determines the type of bind used. Can be one of the following.

**Anonymous**

The bind is performed using no credentials. This is the internal default value.

**Windows security (NEGOTIATE)**

The reporting server authentication is performed against Active Directory utilizing a Windows-specific API.

The bind is done under the Windows account that started the server.

The windows machine that hosts the reporting server should be in the same domain as Active Directory.

**Explicit**

The bind is performed under the account that is defined by configuration parameters `ldap_principal` and `ldap_credentials`.

**Note:** When connecting to Active Directory using `Explicit` or `NEGOTIATE`, `ldap_user_attribute` should have the value `sAMAccountName` or `userPrincipalName`.

**ldap_search_timeout**

Specifies the timeout in seconds for ldap_search. The server default value is 60 seconds.

3. Click Next.
The User Search category of parameters opens, as shown in the following image. The common parameters for User Search and Group Search are automatically populated.

**LDAP Security Provider Configuration**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Sample: dc=ibi,dc=com</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldap_user_base</td>
<td>dc=ibi,dc=com</td>
<td></td>
</tr>
<tr>
<td>ldap_user_scope</td>
<td>subtree</td>
<td></td>
</tr>
<tr>
<td>ldap_user_class</td>
<td>person</td>
<td></td>
</tr>
<tr>
<td>ldap_user_attribute</td>
<td>uid</td>
<td></td>
</tr>
<tr>
<td>ldap_group_attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ldap_user_description</td>
<td>description</td>
<td></td>
</tr>
<tr>
<td>ldap_user_email</td>
<td>mail</td>
<td></td>
</tr>
</tbody>
</table>

4. Configure the following user search and group search parameters.

**User properties**

**ldap_user_base**

Specifies the DN of the entry that serves as the starting point for the search. Consists of attribute=value pairs, separated by commas. The server default is dc=ibi,dc=com.

**ldap_user_scope**

Specifies the scope with which the LDAP realm should search for users. Select **Subtree, Onelevel, or Base**:

- **Subtree** scope indicates that the LDAP realm should search everything under the base DN.
- **Onelevel** scope tells the LDAP server to only search entries one level down from the base DN.
- **Base** indicates that the search should be done at the search base only.

The server default is **Subtree**.
**ldap_user_class**

Specifies the object class used when searching for user entries. The server default is `person`.

**ldap_user_attribute**

Specifies the LDAP attribute used when searching for user entries. `uid` is the default value for LDAP and `sAMAccountName` is the suggested value for Active Directory. One possible reason to change the default value would be to allow users to logon with an email address instead of a user ID. In this case, you might change the value to `mail` or `userPrincipalName` (if this corresponds with the name of the appropriate attribute in your directory).

**ldap_user_group_attribute**

Specifies the LDAP attribute used to identify a group in a user object.

The Active Directory standard is `Memberof`.

**ldap_user_description**

Optional. Specifies the name of the attribute whose value contains description of an object (user, group). The server default is `description`.

**ldap_user_email**

Optional. Specifies the name of the attribute whose value contains the user email address. The server default is `mail`.

**Note:** `ldap_user_class`, `ldap_user_attribute`, `ldap_group_class`, `ldap_group_attribute` are parameters that form a search filter.

The search filter standard syntax conforms to the following structure:

```
(&(Property_Name=Property_Value)(Property_Name=Property_Value))
```

If you change value of the `ldap_user_class` and `ldap_group_class` parameters to an asterisk (*), the search filter syntax can be reduced to the following simplified form (although group support will not work properly):

```
(Property_Name=Property_Value)
```

By specifying an asterisk for these parameters, you achieve simplified search filter syntax, but in effect, disable group support.
**Group properties**

`ldap_group_base`

Specifies the DN of the entry that serves as the starting point for the search. The server default is the `ldap_user_base` value.

`ldap_group_scope`

Specifies the scope with which the LDAP realm should search for groups. Select *Subtree, Onelevel, or Base*:

*Subtree* scope indicates that the LDAP realm should search everything under the base DN.

*Onelevel* scope tells the LDAP server to only search entries one level down from the base DN.

*Base* indicates that the search should be done at the search base only.

The server default is *Subtree*.

`ldap_group_class`

Specifies the object class used when searching for group entries. The server default is *groupofuniquenames*. The Active Directory standard is *group*.

`ldap_group_attribute`

Specifies the LDAP attribute used to identify the name of the group. The server default is *cn*.

`ldap_member_attribute`

Specifies the LDAP attribute used to identify users in a group. The server default is *uniqueMember*. The Active Directory standard is *Member*.

`ldap_nested_groups`

Disables or enables LDAP nested groups support. Select *No* or *Yes*. The server default is *No*, which disables nested group support.

`ldap_group_description`

Optional. Specifies the name of the attribute whose value contains description of an object (user, group). The server default is *description*.
5. Specify whether the server should accept trusted connections.

```
trust_ext
```

Specifies whether the server should accept trusted client connections. If there are multiple security providers, some may allow trusted connections and some may not. In these cases, if the trusted connection is made using a provider without the ability to accept trusted connections, the user will get an authentication error. The server default is `n`.

6. To test the configuration, click Test User Authentication.


Enter a valid user ID and password for this LDAP security provider and click Continue.

If your configuration and credentials are valid, a window opens telling you that you were successfully authenticated.

If they are not valid, you will get a corresponding message.

7. To generate a sample user list, click Sample User List.

A partial list of users with their descriptions and group memberships opens.

8. To generate a sample group list, click Sample Group List.

A partial list of groups with their descriptions and a partial list of members opens.

9. To save this configuration, click Save. To go back to the first configuration page, click Back.

**Support for OpenSSL Ciphers and DH and ECDH Key Exchange**

The following parameters are now supported in the edaserve.cfg configuration file to support OpenSSL ciphers and Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) key exchange.

```
ssl_ciphers
```

Configures the list of supported cipher strings. For example, the following is the default.

```
ALL:!ADH:!LOW:!EXP:@STRENGTH
```

```
ssl_dhparams_file
```

Specifies the full path to the Diffie-Hellman (DH) parameters file. To create a DH key exchange with key size 1024 bits, issue the following command.

```
'openssl dhparam -outform PEM -out dHParm.pem 1024'
```
ssl_ecdhcurve
Specifies a curve name for a specific Elliptic Curve Diffie-Hellman key exchange. To list all available curves, issue the following command.

'openssl ecparam -list_curves'

Server-Wide Kerberos Support for Hive/Cloudera

Connections to a Hive server with Kerberos enabled can be run in one of two ways:

- **Server wide.** The same Kerberos credential is used for all connections. You must obtain a Kerberos ticket before starting the server.

- **Per user.** Each user connecting to the server connects to Hive using their own credentials from the Hive Adapter connection in the user profile.

To setup connections to a Kerberos enabled Hive instance:

1. The Reporting Server has to be secured. The server can be configured with security providers PTH, LDAP, DBMS, OPSYS, or Custom, as well as multiple security providers environment,

2. The following jar files have to be added to the server CLASSPATH or IBI_CLASSPATH variable. These are default locations and vary by distribution and release:

   /hive_home/lib/hive-jdbc-standalone.jar
   /hadoop_home/hadoop-common.jar
   /hadoop_home/client/hadoop-auth.jar

Kerberos Server-Wide Requirements

In this configuration, all connections to the Hive instance will be done with the same Kerberos user ID derived from the Kerberos ticket that is created before the server starts.

1. Create Kerberos ticket using **kinit**:

   kerbid01

   where:

   kerbid01

   Is a Kerberos ID.
2. Verify Kerberos ticket using `klist`. The following message should be returned:

```
Ticket cache: FILE:/tmp/krb5cc_532
Default principal: kerbid01@REALM.COM

Valid starting   Expires       Service principal
04/29/16 16:26:50 04/30/16 02:26:53 krbtgt/REALM.COM@REALM.COM
renew until 05/06/16 16:26:50
```

3. Before configuring the Hive Adapter connection to a Kerberos enabled instance, the connection should be tested. Log in to the system running Hive and use Beeline, the native tool, to test it.

4. Start the server in the same Linux session where the Kerberos ticket was created. Log in to the Web Console and click the Adapters tab.

5. Right-click Apache Hive also Cloudera Impala. Use the following parameters to configure the adapter:

   **URL**
   
   Enter the same URL that you use to connect to the Hive Instance using Beeline.

   ```
   jdbc:hive2://server:10000/default;principal=hive/server@REALM.COM
   ```

   **Security**
   
   Set to Trusted.

6. In the Select profile drop-down menu, select the `edasprof` server profile.

7. Click Configure.

8. Next, configure Java services. Click the Workspace tab and expand the Java Services folder.

9. Right-click `DEFAULT` and select Properties.

10. Expand the JVM Settings section. In the JVM options box, add the following:

    ```
    -Djavax.security.auth.useSubjectCredsOnly=false
    ```

11. Restart Java services.

Once these steps are completed, the adapter can be used to access a Kerberos-enabled Hive instance.
Kerberos Per User Requirements

In this configuration, each connected user has a Hive Adapter connection with Kerberos credentials in the user profile.

1. Enable multi-user connection processing for Kerberos by adding the following line to your profile (edasprof.prf):

   ```
   ENGINE SQLHIV SET ENABLE_KERBEROS ON
   ```

2. Configure the Hive Adapter Connection in the user profile using the following values:

   **URL**
   
   ```
   jdbc:hive2://server:10000/default;principal=hive/server@REALM.COM;
   auth=kerberos;kerberosAuthType=fromSubject
   ```

   **Security**
   
   Set to Explicit

   **User and Password**
   
   Enter your Kerberos user ID and password. The server will use those credentials to create a Kerberos ticket and connect to a Kerberos-enabled Hive instance.

   Note that the user ID that you use to connect to the server does not have to be the same as the Kerberos ID you use to connect to a Kerberos enabled Hive instance.

   **Select Profile**
   
   Select your profile or enter a new profile name consisting of the security provider, an underscore and the user ID. For example, *ldap01_pgmxxx*.

3. Click **Configure**.

Using Kerberos for Single Sign-On on Linux

A server started with security provider OPSYS can be configured for Kerberos connections.

To implement the single sign on Kerberos security:

1. On the Access Control page, right-click the OPSYS provider, and select **Properties** from the shortcut menu.

   The OPSYS Security Configuration page opens.

2. In the krb5_srv_principal * field, enter your server principal used for Kerberos security.

3. Click **Save**.

   The edaserve.cfg file is updated with this attribute.
4. On the Workspace page, expand the Special Services and Listeners folder, right-click TCP/HTTP, and select Properties of TCP.

5. Check SECURITY=KERBEROS.

6. Click Save and Restart Listener.

The odin.cfg file is updated with this attribute.

When the server is started, a user can connect to the Web Console from Internet Explorer without a prompt for user ID and password. The Login Info shows connection type Kerberos. The connection is done using the Kerberos ticket from the browser. The connected user ID is derived from this ticket.

Connection to the server requires that there is a local OPSYS user ID with the same name as the Kerberos user ID on the operating system running the server. This user ID is used for tscom3 process impersonation.

If a user signs off from the Kerberos connection, the user can make explicit connections with the local Unix user ID and password. Connection with another Kerberos user ID as an explicit connection will not work.

Upload Support for Validation of File Extensions

You can configure uploads to operate on a specific list of file extensions.

1. On the Applications tab, click the Application Settings button on the ribbon, or right-click the Application Directories tree and select Application Settings from the shortcut menu.

By default, the upload_allowed entry field contains a comma-separated list of all file extensions that the server can upload.

2. Edit the list to contain only the file extensions that should be allowed.

3. Click Save and Restart Server.

After the server has restarted, the Upload Wizard will not show files with unsupported extensions if you click Select Upload File. If you try to drag a file with an unsupported file extension to the Upload Wizard, a message will be displayed that it is an invalid type of file.

Encoding Passwords for a Custom Security Provider

You can configure Custom Provider authentication to encode passwords so that they cannot be viewed when they are passed as parameters to the authentication procedure.

1. On the Access Control tab, right-click a Custom Security Provider and select Properties from the shortcut menu, or right-click the CUSTOM folder and select New from the shortcut menu.

The CUSTOM Security Provider Configuration page opens.
2. Select y from the cust_hashpasswd drop-down list.
3. Click Save.

When a user is authenticated, the server will send SHA-256 hashed passwords to the authentication procedure. This means that the passwords will be transformed into values that will be unusable for signing in to the Web Console and server. The hashed values have to be stored as passwords in the SQL database being used for authentication. As an alternative, you can create a utility called from the CUSTOM provider authentication procedure that decodes hashed passwords before sending a request to the SQL database for authentication.

**Support for One-Part or Two-Part Names for the Primary Security Provider**

The server administrator can choose whether to register users and groups for the primary security provider with one-part or two-part names. With two-part names, the user ID or group name is prepended with the provider name and a backslash. For example:

OPSYS\user1

or

LDAP01\user2

By default, all users and groups are registered using two-part names. To configure one-part registrations:

1. On the Access Control tab, click Settings and then Access Control Settings on the ribbon, or right-click the Access Control tree and select Settings and then Access Control Settings from the shortcut menu.
2. Select $n$ from the `prepend_provider_name` drop-down list in the General section of the page, as shown in the following image.

![Access Control Settings](image)

3. Click Apply and Restart Server.

The `prepend_provider_name=n` attribute is added to the edaserve.cfg file.

**Simplified User Interface for Configuring Directory/File Privileges**

The server administrator can now access Directory/File Privileges as a right-click menu option from any security subject (role, user, or group) on the Access Control tree.

The new Directory/File Privileges page enables the server administrator to configure privileges for all directories and files in those directories from a single page.

The new page has individual check boxes for Read, Write, and Execute privileges that can be changed on this page. The page is refreshed to update privileges for all inherited locations.

Each directory on the page has a right-click option to Show All Directories/Files. If this option is chosen, all nested folders and files are listed with their privileges.
The new Directory/File Privileges page is shown in the following image.

### Directory/File Privileges

<table>
<thead>
<tr>
<th>Directory/File</th>
<th>Read</th>
<th>Write</th>
<th>Execute</th>
<th>List</th>
<th>Status</th>
<th>Inherit Subject</th>
<th>Inherit Directory</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>AppRoot</td>
</tr>
<tr>
<td>~</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>homeapps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>user1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>etc</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>/home/apps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>/home/apps/user1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>/usr</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>/var</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>/etc</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

### Registration of Users and Groups for inactive Security Providers

From the Access Control page, the server administrator can now register users and groups in server roles for a security provider that is not currently activated. These registrations will be available when the server is started with security ON and with that provider activated.

In addition, a server running with security OFF now has an Access Control page that a server administrator can use to register users and groups and to activate and deactivate providers.

One use for this feature is to register a Server Administrator for a Provider that will be active after the next server restart.

### Web Console

This section describes new Web Console features.
Setting a Web Session Timeout Warning

The server can post a warning about web session expiration before the session actually expires.

To set the number of minutes prior to session expiration that the warning displays:

1. On the Workspace tab, expand the Special Services and Listeners folder.
2. Right-click TCP/HTTP and select Properties of HTTP from the shortcut menu.

The Listener Configuration page opens.

3. Expand the Sessions Control group, as shown in the following image.

4. Enter a number of minutes, \( n \), in the TIMEOUT_WARNING text box.

   By default, the value is zero (0), meaning that no warning will be displayed.

5. Click Save and Restart Server.

   The \( \text{TIMEOUT\_WARNING} = n \) attribute is added to the odin.cfg file.

Responding to the warning message resets the session expiration timer.

Using the Web Console Binocular Search

The binocular search is available on all web Console pages that display a list of files. For example, the binocular search is available on the Synonym Candidates pages, the Manage Files page, and the Application Path Configuration page.
Clicking the binoculars icon opens a Find box where you can enter characters, as shown in the following image.

As you click on each row with a match, its number within the total number of matches is displayed in the form $n$ of $m$.

All matching characters on the list will be highlighted. You can move from one matching row to the next or prior match using the arrow keys.

You can filter the list to display only the rows with matching characters by clicking the Filter icon. Clicking the Filter icon again restores the list.
Web Console Text Editor Enhancements

The Web Console Text Editor that is used to edit procedures, files, and server configuration files was enhanced with line numbers and an improved search facility. A search for characters in the Find text box opens a window with the list of lines matching the pattern, as shown in the following image.

As you click each line in the Find window, the Web Console text editor scrolls to display that line.

When you edit configuration files from the Workspace tab, the editor toolbar has new actions to:

- Refresh from disk.
- Reset to the server default settings.

Displaying the Connected User ID on the Web Console

When a server is running with security enabled, the upper right corner of the Web Console browser window displays the security provider (for two-part names) and the user ID of the connected user.
Adapter Enhancements

This section describes new adapter features. All adapters can be used for WebFOCUS Web Query Reporting and SQL requests, and as sources for DataMigrator flows.

The server supports adapters designed to access a wide variety of data sources. Using the graphical Web Console, you can configure these adapters and create the metadata you need to seamlessly access the corresponding type of data.

On the Web Console, adapters are grouped as SQL, XML-based, ERP, OLAP, Procedures, Sequential and Indexed, DBMS, and Social Media.

**In this chapter:**
- All Adapters
- SQL Adapters
- ERP Adapters
- GIS Adapters
- OLAP Adapters
- DBMS Adapters
- Procedures Adapters
- Sequential and Indexed Adapters
- Social Media Adapters
- XML-Based Adapters

**All Adapters**

This section provides descriptions of new features for all adapters.

**Changing Common Adapter Settings**

The Web Console Adapters page has a new button in the Troubleshooting group labeled *Change common adapter settings* that allows you to customize data type mappings and other miscellaneous settings that are common to all adapters.
Adapter-specific settings are customizable as well, and can be accessed by right-clicking a configured adapter folder and selecting Change Settings from the shortcut menu.

**SQL Adapters**

This section provides detailed descriptions of new features for SQL adapters.

**All SQL Adapters**

This section provides detailed descriptions of new features for all SQL adapters.

**Optimization of Simplified Numeric Functions**

The following new simplified numeric functions are optimized by the SQL adapters:

- **CEILING.** Returns the smallest integer value greater than or equal to a value.
- **EXPONENT.** Raises the constant e to a power.
- **FLOOR.** Returns the largest integer less than or equal to a value.
- **MOD.** Calculates the remainder from a division.
- **POWER.** Raises a value to a power.

**Optimization of Simplified Functions REPLACE, TOKEN, and POSITION**

The following simplified character functions are optimized by the SQL adapters.

- **REPLACE.** Replaces all instances of a string.
- **TOKEN.** Extracts a token (substring) from a string.
- **POSITION.** Returns the first position of a substring.

**Enhancement to the Optimization of LIKE for Fixed Length Fields**

In prior releases, the LIKE operator was not optimized for those RDBMS engines (such as Db2) that count trailing blanks when comparing columns because the FOCUSWeb Query LIKE operator is not sensitive to trailing blanks. Now, LIKE is optimized by removing trailing blanks in the SQL passed to those RDBMS engines using the RTRIM function. (RTRIM(column) LIKE mask).

LIKE optimization works as long as the mask does not end with an underscore character (_). Optimization does work when the mask starts or ends with a percent sign (%).
**Example:** Optimizing the LIKE Operator for Fixed Length Fields

The following request creates a fixed-length field named CATEGORY and uses it in a WHERE test with the LIKE operator.

```sql
SET TRACEUSER=ON
SET TRACEON=STMTRACE//CLIENT
SET TRACESTAMP=OFF
DEFINE FILE WF_RETAIL_LITE
CATEGORY/A10 = PRODUCT_SUBCATEG;
END
TABLE FILE WF_RETAIL_LITE
SUM COGS_US
BY CATEGORY
WHERE CATEGORY LIKE '%phone'
ON TABLE SET PAGE NOLEAD
END
```

The following SQL is generated, which trims trailing blanks from the CATEGORY column.

```sql
AGGREGATION DONE ...
SELECT
T7."PRODUCT_SUBCATEG",
SUM(T1."COGS_US")
FROM
D999OL29_wf_retail_sales T1,
D999OL29_wf_retail_product T7
WHERE
(T7."ID_PRODUCT" = T1."ID_PRODUCT") AND
(TRANSLATE("TRIM(TRAILING ' ' FROM T7."PRODUCT_SUBCATEG") LIKE '%$phone')
GROUP BY
T7."PRODUCT_SUBCATEG"
ORDER BY
T7."PRODUCT_SUBCATEG"
FOR FETCH ONLY;
```

The output is shown in the following image.

![Output Image](image)

**PERSISTENCE Option for HOLD FORMAT sqlengine**

The PERSISTENCE option has been added to the HOLD command when it is used to create an SQL table.
This new command option allows you to generate intermediate tables of different types that will be used only during UPLOAD and EBL requests to accelerate performance by keeping all processing on the DBMS server instead of downloading data into a HOLD file. The actual type of the intermediate table will be determined at run time, based on specific DBMS-supported features and the data-populating mechanisms being used.

The syntax is:

```
HOLD FORMAT sqlengine PERSISTENCE {STAGE | PERMANENT}
```

where:

- `sqlengine` Identifies the relational DBMS in which to create the table.
- `STAGE` Will create either a Volatile or GLOBAL TEMPORARY table, for a DBMS that supports that functionality, currently HP Vertica, Db2, Oracle, Teradata, MS SQL, and MySQL. For a DBMS that does not support that functionality, a message will display and the table will not be created.
- `PERMANENT` Will create a regular SQL table with a uniquely-generated name that will be used in the request and will be available for further use after the request ends, but will be dropped at the end of the session. This is the default value for PERSISTENCE for HOLD FORMAT `sqlengine`.

Optimizing a Heterogeneous Join for a Qualified DBMS

Joins, including outer joins, between segments that belong to different SQL engines now are eligible for SQL optimization. The SQL adapters will optimize a join for segments that belong to the same engine, and then intermediate results from each engine will be combined on the server into one resulting join.

Enhanced BY Clause Optimization

The relational adapters now optimize reports that contain a BY field more efficiently, by no longer passing MAX(field) in the SELECT list to the RDBMS.

Tutorials Now Created With Foreign Key Constraints

When creating Star Schema or Retail Demo tutorial tables from the Web Console or Data Management Console, the fact tables now have foreign key constraints for each of the dimension tables. This facilitates building a cluster join from these tools. It also prevents loading rows in the fact table without corresponding rows in the dimension tables.
For example, the following is the Access File for the wf_retail_labor fact table, which has a foreign key constraint for the wf_retail_employee table.

```sql
SEGNAME=WF_RETAIL_LABOR,
   TABLENAME=wrd_wf_retail_labor,
   CONNECTION=wfretail,
   KEY=ID_LABOR,
   FOREIGN_KEY=fact_labor_id_employee_fk,
   PRIMARY_KEY_TABLE=wrd_wf_retail_employee,
   PRIMARY_KEY_COLUMN=ID_EMPLOYEE,
```

Creating Tutorials With Lowercase Column Names

Previously, column names for the tutorials were always created in uppercase. However, certain databases, such as PostgreSQL and those based on it (For example, Greenplum) expects column names to be lowercase.

Now, when the adapter setting ALIAS_CASE is set to LOWER, tutorial column names will be lowercase.

Saving Adapter Settings in a Profile or Stored Procedure

When you customize SQL adapter settings by right-clicking an adapter and selecting Change Settings from the shortcut menu, the top of the page lets you select either Profile or Procedure from the Save settings in drop-down list, as shown in the following image.

Change Settings for MS SQL Server

Save settings in Procedure [v] Select \hisa\mp/sets_sqlss.fex

If you select Procedure, a procedure is automatically generated with the settings you changed. You can accept the default application and procedure name, or you can select a different application in which to store the procedure and enter a file name for the procedure.

The procedure can be added to a Process Flow, making you able to define flow-level settings rather than global settings using the Web Console.

Mapping Spatial Data Stored in RDBMS Columns

The Adapters for MS SQL Server, Oracle, Teradata, and MySQL now support direct read from DBMS spatial columns that store Esri data.

The Access File contains the expression needed to retrieve the spatial data, which is different for each DBMS. The geographic data is returned as a Geometry object that can be used to render maps in WebFOCUSWeb Query GRAPH requests.
For example, in the following SQL Server Master File, the field GEO contains spatial data representing geographic areas. In SQL Server, the data type for this field is GEOMETRY or GEOGRAPHY. In the Master File, it is mapped as text, with a GEOGRAPHIC_ROLE attribute.

```
FIELDNAME=GEO, ALIAS=GEO, USAGE=TX50, ACTUAL=TX, MISSING=ON,
GEOGRAPHIC_ROLE=GEOMETRY_AREA, $
```

The following are the Access File attributes for this field.

```
IELD=GEO,
SQL_FLD_OBJ_TYPE=OPAQUE,
SQL_FLD_OBJ_PROP=GEOMETRY_SHAPE,
SQL_FLD_OBJ_EXPR='DB_EXPR("GEO".STGeometryType() 
+ 
+ CAST("GEO".STSrid AS VARCHAR(10) )
+ 
+ "GEO".STAsText())', $
```

The following request retrieves the GEO field.

```
TABLE FILE EUROPE_SWASIA
PRINT NAME GEO
WHERE NAME IN ('FRANCE','GERMANY','UNITED KINGDOM')
END
```

The SQL passed to SQL Server as a result of this request follows.

```
SELECT
T1."NAME",
(T1."GEO".STGeometryType() + ',' + CAST(T1."GEO".STSrid AS VARCHAR(10) ) + ',' +
T1."GEO".STAsText())
FROM
EUROPE_SWASIA T1
WHERE
(T1."NAME" IN ('FRANCE','GERMANY','UNITED KINGDOM'));`
The following are sample Access File attributes for Oracle.

```
SQL_FLD_OBJ_EXPR='DB_EXPR(CASE "GEO".Get_GType()
WHEN 1 THEN ''Point''
WHEN 2 THEN ''LineString''
WHEN 3 THEN ''Polygon''
WHEN 4 THEN ''GeometryCollection''
WHEN 5 THEN ''MultiPoint''
WHEN 6 THEN ''MultiLineString''
WHEN 7 THEN ''MultiPolygon''
WHEN 8 THEN ''PolyhedralSurface''
ELSE ''Geometry'' END
| ''',
| CAST( "GEO".SDO_SRID AS VARCHAR(10) )
| ''',
| "GEO".Get_WKT())', $

--Teradata:
SQL_FLD_OBJ_EXPR='DB_EXPR(Substring("GEO".ST_GeometryType() From 4)
| ''',
| CAST( "GEO".ST_SRID() AS VARCHAR(10) )
| ''',
| "GEO".ST_AsText())', $
```

**Enhanced Messages About SQL Optimization**

More descriptive messages have been developed when SQL Optimization is not done for a request. In some cases, a message will contain information about ways to possibly change the syntax so that optimization can be used.

**Optimization of Function DTRUNC for First Day of Week**

Calls to the DTRUNC function can be optimized when using the new WEEK parameter that returns the first day of the week in which the date occurs.

**Optimization of Function DTRUNC for YEAR_END, QUARTER_END, MONTH_END and WEEK_END**

Calls to the DTRUNC function can be optimized when using the new YEAR_END, QUARTER_END, MONTH_END and WEEK_END parameters that return the last day of the period in which the date occurs.

**CONCAT Function Optimization**

Simplified character function CONCAT is passed to the Relational Adapters in the generated SQL.
Optimization of the DT_CURRENT_DATE, DT_CURRENT_DATETIME, and DT_CURRENT_TIME Functions

The simplified functions DT_CURRENT_DATE, DT_CURRENT_DATETIME, and DT_CURRENT_TIME are passed to the Relational Adapters in the generated SQL request.

Example: Optimizing the DT_CURRENT_DATETIME Function

The following request calls the DT_CURRENT_DATETIME function.

```
SET TRACEUSER=ON
SET TRACEON=STMTRACE//CLIENT
SET TRACESTAMP=OFF
DEFINE FILE WF_RETAIL_LITE
CURRDT/HYYMDS = DT_CURRENT_DATETIME(SECOND);
END
TABLE FILE WF_RETAIL_LITE
SUM QUANTITY_SOLD CURRDT
BY PRODUCT_CATEGORY
ON TABLE SET PAGE NOLEAD
END
```

The following SQL is generated for Microsoft SQL Server.

```
SELECT
T7."PRODUCT_CATEGORY",
SUM(T1."QUANTITY_SOLD"),
MAX(SYSDATETIME())
FROM
( wrd_wf_retail_sales T1
LEFT OUTER JOIN
wrd_wf_retail_product T7
ON T7."ID_PRODUCT" = T1."ID_PRODUCT" )
GROUP BY
T7."PRODUCTCATEGORY"
ORDER BY
T7."PRODUCTCATEGORY";
```
The output is shown in the following image.

<table>
<thead>
<tr>
<th>Product Category</th>
<th>Quantity Sold</th>
<th>CURRDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>2,034</td>
<td>2016/03/25 14:03:54</td>
</tr>
<tr>
<td>Camcorder</td>
<td>1,956</td>
<td>2016/03/25 14:03:54</td>
</tr>
<tr>
<td>Computers</td>
<td>865</td>
<td>2016/03/25 14:03:54</td>
</tr>
<tr>
<td>Media Player</td>
<td>3,140</td>
<td>2016/03/25 14:03:54</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>4,679</td>
<td>2016/03/25 14:03:54</td>
</tr>
<tr>
<td>Televisions</td>
<td>362</td>
<td>2016/03/25 14:03:54</td>
</tr>
<tr>
<td>Video Production</td>
<td>887</td>
<td>2016/03/25 14:03:54</td>
</tr>
</tbody>
</table>

**SQL Adapters Support CREATE FILE With FOREIGNKEYS**

The CREATE FILE command now supports defining foreign key constraints to enforce referential integrity.

**Syntax:**

**How to Issue a CREATE FILE Command With Foreign Keys**

```sql
CREATE FILE app/synonym [WITHFK [DROP]]
```

where:

- `app/synonym`

  Is the synonym that describes the table to be created.

- `DROP`

  Drops the table if it already exists and then creates it.

**Application Access Control for HyperStage Tables**

**SQL Adapters Optimize Calls to the GIS_POINT Function in a DEFINE**

Given a WKID (Well-Known ID) spatial reference, longitude, and latitude, the GIS_POINT function builds a JSON point defining a Geometry object with the provided WKID, longitude, and latitude. The function is optimized for those SQL engines that can build a JSON geometry object.
**Example: Optimizing Calls to the GIS_POINT Function**

The following request calls the GIS_POINT function in a DEFINE command. Note that the field WKID is defined in the WF_RETAIL_LITE Master File.

```plaintext
SET TRACEUSER=ON
SET TRACEON = STMTRACE//CLIENT
SET TRACESTAMP=OFF

DEFINE FILE wf_retail_lite
Point/A150 MISSING ON ALL=GIS_POINT(WKID, CONTINENT_LONGITUDE, CONTINENT_LATITUDE);
END

TABLE FILE wf_retail_lite
PRINT Point
BY WF_RETAIL_GEOGRAPHY_STORE.CONTINENT_CODE AS 'Continent Code'
ON TABLE SET PAGE NOLEAD
END
```

The following SQL is generated.

```sql
SELECT
T1."ID_STORE",
T1."ID_CUSTOMER",
T2."ID_GEOGRAPHY",
T3."ID_GEOGRAPHY",
T3."CONTINENT_LATITUDE",
T3."CONTINENT_LONGITUDE",
T5."ID_GEOGRAPHY",
T6."ID_GEOGRAPHY",
T6."CONTINENT_CODE"
FROM
( ( ( wrd_wf_retail_sales T1
INNER JOIN
wrd_wf_retail_customer T2
ON (T2."ID_CUSTOMER" = T1."ID_CUSTOMER") )
LEFT OUTER JOIN
wrd_wf_retail_geography T3
ON T3."ID_GEOGRAPHY" = T2."ID_GEOGRAPHY" )
INNER JOIN
wrd_wf_retail_store T5
ON (T5."ID_STORE" = T1."ID_STORE") )
LEFT OUTER JOIN
wrd_wf_retail_geography T6
ON T6."ID_GEOGRAPHY" = T5."ID_GEOGRAPHY"
ORDER BY
T6."CONTINENT_CODE";
```
Enhanced Optimization of Selection Tests

In prior releases, all optimized selection tests were passed to the RDBMS and also re-evaluated when the answer set was returned to WebFOCUS. Starting in this release, individual tests passed to the SQL engines may be excluded from execution. This enhances performance and also allows tests to be optimized that are not totally compatible with how WebFOCUS evaluates them.

Optimization of the PARTITION_REF Function

Calls to the PARTITION_REF function in an aggregation request with a WHERE TOTAL phrase can be passed to SQL engines that have a LAG function, such as Db2, Oracle, Teradata, and Microsoft SQL Server.
Example: Optimizing the PARTITION_REF Function

The following request using the Adapter for Oracle calls the PARITION_REF function, which displays values from prior rows.

```
TABLE FILE WF_RETAIL_LITE
SUM
   COGS_US COMPUTE COGS1 MISSING ON = PARTITION_REF(COGS_US,TABLE, -1);
   BY PRODUCT_CATEGORY
WHERE TOTAL COGS_US GT 421;
ON TABLE SET PAGE NOPAGE
END
```

The following SQL is generated that calls the Oracle LAG function.

```
SELECT
   "SK001_PRODUCT_CATEGORY",
   "VB001_SUM_COGS_US",
   "LAG002"
FROM (SELECT
   T7."PRODUCT_CATEGORY" AS "SK001_PRODUCT_CATEGORY",
   SUM(T1."COGS_US") AS "VB001_SUM_COGS_US", LAG( SUM(T1."COGS_US"), 1)
   OVER( ORDER BY 
   T7."PRODUCT_CATEGORY") AS "LAG002"
FROM ( D99964R6_sales T1
      LEFT OUTER JOIN 
      D99964R6_product T7
      ON T7."ID_PRODUCT" = T1."ID_PRODUCT" )
GROUP BY
   T7."PRODUCT_CATEGORY"
) X
WHERE ("VB001_SUM_COGS_US" > 421);
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Product Category</th>
<th>Cost of Goods</th>
<th>COGS1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camcorder</td>
<td>$799.00</td>
<td>421.00</td>
</tr>
<tr>
<td>Computers</td>
<td>$1,936.00</td>
<td>799.00</td>
</tr>
<tr>
<td>Media Player</td>
<td>$3,506.00</td>
<td>1,936.00</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>$2,561.00</td>
<td>3,506.00</td>
</tr>
</tbody>
</table>
New Optimization Controls Using SET FEATOPT

The adapter SET FEATOPT command can be used to selectively disable optimization of specific WebFOCUS features when the results may be undesirable. The ability to disable IF-THEN-ELSE and IF TOTAL or WHERE TOTAL optimization has been added. By default, all optimization features are enabled.

FEATOPT settings are displayed in Web Console debug mode.

The SET OPTIFTHENELSE command has been deprecated.

**Syntax:** How to Control IF TOTAL and WHERE TOTAL Optimization

```sql
ENGINE sqlengine SET FEATOPT TOTALTEST {ON|OFF}
```

where:

*sqlengine*

Specifies the adapter.

*ON*

Passes the HAVING clause to the DBMS, as well as CASE logic, when applicable. The HAVING clause results from passing an IF TOTAL or WHERE TOTAL phrase from the WebFOCUS request. ON is the default value.

*OFF*

Suppresses adding the HAVING clause to the generated SELECT.

**Example:** Disabling WHERE TOTAL Optimization Using FEATOPT

The following request against SQL Server has a WHERE TOTAL test.

```sql
SET TRACEUSER=ON
SET TRACEON = STMTRACE\CLIENT
SET TRACESTAMP=OFF
TABLE FILE WF_RETAIL_LITE
SUM COGS_US GROSS_PROFIT_US
BY PRODUCT_CATEGORY
WHERE TOTAL COGS_US GT 400000
END
```
The default generated SQL follows. The WHERE TOTAL test has been passed to the DBMS as a HAVING clause.

```sql
SELECT T7."PRODUCT_CATEGORY",
       SUM(T1."COGS_US"),
       SUM(T1."GROSS_PROFIT_US")
FROM (wrd_wf_retail_sales T1
       LEFT OUTER JOIN
       wrd_wf_retail_product T7
       ON T7."ID_PRODUCT" = T1."ID_PRODUCT" )
GROUP BY T7."PRODUCT_CATEGORY"
HAVING (SUM(T1."COGS_US") > 400000)
ORDER BY T7."PRODUCT_CATEGORY";
```

Adding the ENGINE SQLMSS SET FEATOPT TOTALTEST OFF command generates the following SQL. The WHERE TOTAL test has not been passed in the generated SQL and will be handled by WebFOCUS.

```sql
SELECT T7."PRODUCT_CATEGORY",
       SUM(T1."COGS_US"),
       SUM(T1."GROSS_PROFIT_US")
FROM (wrd_wf_retail_sales T1
       LEFT OUTER JOIN
       wrd_wf_retail_product T7
       ON T7."ID_PRODUCT" = T1."ID_PRODUCT" )
GROUP BY T7."PRODUCT_CATEGORY"
ORDER BY T7."PRODUCT_CATEGORY";
```

**Syntax:** How to Control IF-THEN-ELSE Optimization

```
ENGINE sqlengine SET FEATOPT IFTHENELSE {ON|OFF}
```

where:

- `sqlengine`

  Specifies the adapter.

- `ON`

  Passes the IF-THEN-ELSE expressions used to create a DEFINE field to the DBMS as CASE logic, when possible. ON is the default value. This optimization is subject to the limitations on optimizing DEFINE fields. The DEFINE field must be an object of a selection test or an aggregation request.
OFF

Suppresses passing IF-THEN-ELSE expressions in the generated SQL.

Note: The SET OPTIFTHENELSE command has been deprecated.

Example: Disabling IF-THEN-ELSE Optimization Using FEATOPT

The following request against SQL Server has a DEFINE field created using IF-THEN-ELSE syntax.

```
SET TRACEUSER=ON
SET TRACEON = STMTRACE//CLIENT
SET TRACESTAMP=OFF
DEFINE FILE WF_RETAIL_LITE
DEF1 = IF (PRODUCT_CATEGORY EQ ' Accessories') AND (BRAND EQ 'Sony ') AND (PRODUCT_SUBCATEG EQ 'Headphones') THEN 1 ELSE 0;
END
TABLE FILE WF_RETAIL_LITE
PRINT PRODUCT_CATEGORY PRODUCT_SUBCATEG BRAND
WHERE DEF1 EQ 1
END
```

The default generated SQL follows. The IF-THEN-ELSE syntax has been passed to the DBMS as CASE logic.

```
SELECT
T7."BRAND",
T7."PRODUCT_CATEGORY",
T7."PRODUCT_SUBCATEG"
FROM
wrd_wf_retail_product T7
WHERE
((CASE WHEN ((T7."PRODUCT_CATEGORY" = ' Accessories') AND (T7."BRAND" = 'Sony ') AND (T7."PRODUCT_SUBCATEG" = 'Headphones')) THEN 1 ELSE 0 END) = 1);
```

Adding the ENGINE SQLMSS SET FEATOPT IFTHE NELSE OFF command generates the following SQL. The IF-THEN-ELSE syntax has not been passed in the generated SQL and will be handled by WebFOCUS.

```
SELECT
T7."BRAND",
T7."PRODUCTCATEGORY",
T7."PRODUCT_SUBCATEG"
FROM
wrd_wf_retail_product T7;
```

New SQL Functions

The SQL Functions MOD, CEIL, FLOOR, LEAST, and GREATEST are now supported.
**MOD: Returning the Remainder of a Division**

The SQL function MOD returns the remainder of the first argument divided by the second argument.

**Syntax:**

**How to Return the Remainder of a Division**

MOD(n, m)

where:

n
Numeric
Is the dividend (number to be divided).

m
Numeric
Is the divisor (number to divide by). If the divisor is zero (0), MOD returns NULL.

**Example:**

**Returning the Remainder of a Division**

MOD returns the remainder of \( n \) divided by \( m \).

**MOD(n, m)**

For \( N=16 \) and \( M=5 \), the result is 1.

For \( N=34.5 \) and \( M=3 \), the result is 1.5.

**CEIL: Returning the Smallest Integer Greater Than or Equal to a Value**

CEIL returns the smallest integer value not less than the argument. CEILING is a synonym for CEIL.

**Syntax:**

**How to Return the Smallest Integer Greater Than or Equal to a Value**

CEIL(n)

where:

n
Numeric or Alphanumeric
Is the value less than or equal to the returned integer. For exact-value numeric arguments, the return value has an exact-value numeric type. For alphanumeric or floating-point arguments, the return value has a floating-point type.
Example: Returning an Integer Greater Than or Equal to a Value
CEIL returns an integer greater than or equal to the argument.

\[
\text{CEIL}(N)
\]

For \(N=1.23\), the result is 2.
For \(N=-1.23\), the result is -1.

FLOOR: Returning the Largest Integer Less Than or Equal to a Value
FLOOR returns the largest integer value not greater than a value.

Syntax: How to Return the Largest Integer Less Than or Equal to a Value

\[
\text{FLOOR}(n)
\]

where:

\(n\)

Numeric or Alphanumeric

Is the value greater than or equal to the returned integer. For exact-value numeric arguments, the return value has an exact-value numeric type. For alphanumeric or floating-point arguments, the return value has a floating-point type.

Example: Returning an Integer Less Than or Equal to a Value
FLOOR returns an integer less than or equal to the argument.

\[
\text{FLOOR}(N)
\]

For \(N=1.23\), the result is 1.
For \(N=-1.23\), the result is -2.

LEAST: Returning the Smallest Value
With two or more arguments, LEAST returns the smallest (minimum-valued) argument. The arguments are compared using the following rules:

- If any argument is NULL, the result is NULL. No comparison is needed.
- If the return value is used in an INTEGER context, or all arguments are integer-valued, they are compared as integers.
- If the return value is used in a floating-point context, or all arguments are floating-point-valued, they are compared as floating-point values.
If the arguments comprise a mix of numbers and strings, they are compared as numbers.

If any argument is a character string, the arguments are compared as character strings. In all other cases, the arguments are compared as binary strings.

Syntax: How to Return the Smallest Value

LEAST(value1, value2, ..., valuen)

where:

value1, value2, ..., valuen

Numeric or alphanumeric

Are the values to be compared.

Example: Returning the Smallest Value

LEAST returns the smallest argument.

LEAST(X, Y, Z)

For X=2, Y=0, and Z=-1, the result is -1.

For X='B', Y='A', and Z='C', the result is 'A'.

GREATEST: Returning the Largest Value

With two or more arguments, GREATEST returns the largest (maximum-valued) argument. The arguments are compared using the following rules:

If any argument is NULL, the result is NULL. No comparison is needed.

If the return value is used in an INTEGER context, or all arguments are integer-valued, they are compared as integers.

If the return value is used in a floating-point context, or all arguments are floating-point-valued, they are compared as floating-point values.

If the arguments comprise a mix of numbers and strings, they are compared as numbers.

If any argument is a character string, the arguments are compared as character strings. In all other cases, the arguments are compared as binary strings.

Syntax: How to Return the Largest Value

GREATEST(value1, value2, ..., valuen)
where:

\(value_1, value_2, \ldots, value_n\)

Numeric or alphanumeric

Are the values to be compared.

**Example:** Returning the Largest Value

GREATEST returns the smallest argument.

\[ \text{GREATEST}(X, Y, Z) \]

For \(X=2\), \(Y=0\), and \(Z=-1\), the result is 2.

For \(X='B'\), \(Y='A'\), and \(Z='C'\), the result is 'C'.

**SQL Adapters Create Unique Keys for HOLD FORMAT SQL_SCRIPT**

KEY information for SQL_SCRIPT files created using HOLD FORMAT SQL_SCRIPT is propagated to the Access File, which improves JOIN optimization.

**Example:** Propagating Key Information to the Access File Generated by HOLD FORMAT SQL_SCRIPT

The following request generates an SQL_SCRIPT file with two keys.

```
TABLE FILE WF_RETAIL_LITE
SUM MIN.COGS_US MAX.GROSS_PROFIT_US
BY BUSINESS_SUB_REGION
BY STATE_PROV_CODE_ISO_3166_2
WHERE BUSINESS_SUB_REGION EQ 'Midwest' OR 'East'
WHERE COUNTRY_NAME EQ 'United States'
ON TABLE HOLD AS RETAILS FORMAT SQL_SCRIPT
END
```
This request produces the following script file, retails.sql.

```sql
SELECT
T3."BUSINESS_SUB_REGION" AS "SK001_BUSINESS_SUB_REGION",
T3."STATE_PROV_CODE_ISO_3166_2" AS "SK002_STATE_PROV_CODE_ISO_3166",
MIN(T1."COGS_US") AS "VB001_MIN_COGS_US",
MAX(T1."GROSS_PROFIT_US") AS "VB002_MAX_GROSS_PROFIT_US"
FROM
wrd_wf_retail_sales T1,
wrd_wf_retail_customer T2,
wrd_wf_retail_geography T3
WHERE
(T2."ID_CUSTOMER" = T1."ID_CUSTOMER") AND
(T3."ID_GEOGRAPHY" = T2."ID_GEOGRAPHY") AND
(T3."COUNTRY_NAME" = 'United States') AND
(T3."BUSINESS_SUB_REGION" IN('Midwest', 'East'))
GROUP BY
T3."BUSINESS_SUB_REGION",
T3."STATE_PROV_CODE_ISO_3166_2"
```

The RETAILS Master File follows.

```sql
FILENAME=RETAILS, SUFFIX=SQLMSS , $
SEGMENT=RETAILS, SEGTYPE=S0, $
   FIELDNAME=BUSINESS_SUB_REGION, ALIAS=SK001_BUSINESS_SUB_REGION,
   USAGE=A25V, ACTUAL=A25V,
   MISSING=ON,
   TITLE='Customer,Business,Sub Region', $
   FIELDNAME=STATE_PROV_CODE_ISO_3166_2,
   ALIAS=SK002_STATE_PROV_CODE_ISO_3166, USAGE=A5V, ACTUAL=A5V,
   MISSING=ON,
   TITLE='Customer,State,Province,ISO-3166-2,Code', $
   FIELDNAME=COGS_US, ALIAS=VB001_MIN_COGS_US, USAGE=D20.2M, ACTUAL=D8,
   MISSING=ON,
   TITLE='Cost of Goods', $
   FIELDNAME=GROSS_PROFIT_US, ALIAS=VB002_MAX_GROSS_PROFIT_US,
   USAGE=D20.2M, ACTUAL=D8,
   MISSING=ON,
   TITLE='Gross Profit', $
```

The RETAILS Access File contains the key information.

```sql
SEGNAME=RETAILS,
   CONNECTION=wfretail,
   DATASET=RETAILS.SQL,
   KEY=BUSINESS_SUB_REGION/STATE_PROV_CODE_ISO_3166_2,
   SUBQUERY=Y, $
```
Enhancement to SQL Adapter Native Messaging Interface

In prior releases, the messages returned by an SQL adapter were displayed as one long string, making them difficult to read and understand. Starting in this release, native SQL messages will be presented in readable blocks, each as a FOC1500 message on a separate line. For example:

(FOC1400) SQLCODE IS -1 (HEX: FFFFE000) XOPEN: 42000
(FOC1500) : Microsoft SQL Server Native Client 10.0: [42000] Syntax error,
(FOC1500) : permission violation, or other nonspecific error
(FOC1406) SQL OPEN CURSOR ERROR. : __WF_RETAIL__WF_RETAIL__SALES

Improved Messaging for Environmental Problems

Messages that the SQL Adapters return to edaprint in case of a non-existent or incorrect path have been improved.

- Each environmental path is now clearly printed in edaprint.
- Those that do not exist are marked as follows.
  ----NOT FOUND

Random Data Sampling in SQL

Certain SQL Adapters such as Microsoft SQL Server, Oracle (ODBC and JDBC), and SAP HANA support random data sampling. For SAP HANA, sampling is only supported for column-based tables. A report can return a random sample of data from these RDBMSs using the following settings.

```
ENGINE engine SET TABLESAMPLING_TYPE {OFF|ROW|BLOCK}
ENGINE INT SET TABLESAMPLING_PERCENTAGE n
```

where:

**engine**

- Specifies the adapter.

**OFF**

- Turns off table sampling.

**ROW**

- Uses row-based random sampling. This is the default value.

**BLOCK**

- Uses page-based random sampling.
Is a number between 1 and 99 that specifies the percentage of records to be returned. The value blank, which is the default, indicates that sampling is not in effect.

For example, the following settings for SAP HANA return about 5% of the requested answer set.

```
ENGINE INT SET TABLESAMPLING_PERCENTAGE 5
ENGINE SQLHANA SET TABLESAMPLING_TYPE ROW
```

Consider the following request.

```
TABLE FILE filename WRITE D END
```

The following SQL is generated as a result of these settings.

```
SELECT 
 T1."D",
 SUM(T1."D")
FROM tablename T1 TABLESAMPLE BERNOULLI(05)
GROUP BY 
 T1."D"
ORDER BY 
 T1."D";
```

Using BLOCK instead of ROW will generate native TABLESAMPLE SYSTEM(n) instead of BERNOULLI(n).

**Optimization of Expressions on Partial Dates**

Partial date formats such as YY, Q, D, and M, which are actually integer values, can now be optimized when used in expressions.

**Adapter for Apache Drill**

Apache Drill is a query engine for data managed by Hadoop in HDFS or Mapr-FS, a variety of NoSQL databases including HBase and MapR-DB, and MongoDB, and other data sources. Drill can use Hive metadata or derive metadata from self-describing data files.

**Adapter for Db2**

This section provides detailed descriptions of new features for the Adapter for Db2.
Change to Multi-Row Inserts on z/OS

Db2 on z/OS changed the default behavior for multiple-row inserts from non-atomic to atomic. This feature resets the default behavior of the Adapter for Db2 back to non-atomic. Non-atomic maximizes the number of valid rows that are added to a table when a non-fatal error (such as duplicate insert) occurs.

Support for TIMESTAMP for CDC_START on z/OS

Db2 version 10 now supports the Timestamp value of a START parameter in a CDC synonym.

Conversion to ANSI Date, Time, and Timestamp Literals

When the Adapter for Db2 converts a request that contains a date, time, or timestamp literal to SQL, it converts the literal to ANSI standard format in the generated SQL.

For example, consider the following WHERE phrase:

```
WHERE DATECONST1 EQ '19010228'
```

The adapter will convert the WHERE phrase to the following predicate in the generated SQL:

```
WHERE (T1."DATECONST1" = DATE '1901-02-28')
```

Support DECFLOAT Data Type as MATH and XMATH

The new Db2 data types MATH and XMATH support the necessary precision for compatible decimal computation. The Db2 data type DECFLOAT(16) can be mapped as ACTUAL format MATH or FLOAT, and the Db2 DECFLOAT(34) data type can be mapped as ACTUAL format XMATH or FLOAT using the following settings.

```
SQL DB2 SET CONV_DECFLOAT16 MATH
SQL DB2 SET CONV_DECFLOAT16 FLOAT
SQL DB2 SET CONV_DECFLOAT34 XMATH
SQL DB2 SET CONV_DECFLOAT34 FLOAT
```

The default precision is 18 and the default scale is 2 for the MATH data type. You can change these defaults using the following commands, where $p$ is the precision and $s$ is the scale.

```
SQL DB2 SET CONV_DECFM_PREC $p$
SQL DB2 SET CONV_DECFM_SCALE $s$
```
The default precision is 34 and the default scale is 2 for the XMATH data type. You can change these defaults using the following commands, where \( p \) is the precision and \( s \) is the scale.

\[
\begin{align*}
\text{SQL DB2 SET CONV_DECFX_PREC } & \ p \\
\text{SQL DB2 SET CONV_DECFX_SCALE } & \ s
\end{align*}
\]

**Adapter for C9 INC**

This section provides detailed descriptions of new features for the Adapter for C9 INC.

**New Setting Available for Temporal Lower Boundary Date**

The TREND_LOW_BOUNDARY = YYYY-MM-DD setting provides a lower boundary for the temporal trending period that will be used when the lower boundary is not specified by a TABLE query FROM-TO clause.

The default c9 value is 1970-01-01. Historical data filtered by the following expression may be excessive and cause performance problems.

\[
\text{WHERE DAILY/MONTHLY/QUARTERLY/YEARLY_TREND FROM '1970-01-01' TO CURRENT_DATE}
\]

This setting will relieve such problems.

**Adapter for Cloudera Impala**

The Adapter for Cloudera Impala provides for analysis of both structured and complex data. Impala provides a JDBC driver and the Hive Query Language, a SQL-like interface with a real time query capability that shares the metadata layer and query language with Hive.

**Adapter for EXASol ODBC**

The Adapter for EXASol ODBC is new in this release and can be found under SQL Adapters on the Web Console. The adapter provides read/write access to data stored in an EXASOL in-memory database is available on Windows and RH Linux. The adapter uses the EXASolution ODBC driver API.

**Adapter for EXASol ODBC: Extended Bulk Load**

An extended bulk load option is available that uses the IMPORT command for faster loading.
Adapter for Google BigQuery

The Adapter for Google BigQuery is new in this release and can be found under SQL Adapters on the Web Console. The adapter is used to load data into the Google BigQuery environment and report against the information that is residing in the Google BigQuery environment. BigQuery provides a REST-based API that requires all requests to be authorized by an authenticated user or a service account.

Adapter for Hive

This section provides detailed descriptions of new features for the Adapter for Hive.

Synonym Creation for SQL Strings

The adapter supports Create Synonym for an External SQL Script.

Insert/Update for ORC Files

As of Hive 0.14, the UPDATE operation is supported for data stored as Optimized Row Columnar (ORC) format. While not designed for OLTP, this does provide an option for dimension tables and for correcting data.

DataMigrator and MODIFY as well as SQL Pass through can take advantage of this capability.

When the Hive metadata is created it must contain these additional parameters:

CLUSTERED BY (column [,column..]) INTO n BUCKETS
STORED AS ORC TBLPROPERTIES ('transactional'='true')

Where the number of buckets and one more column names that are used to cluster the data are specified.

Note that DataMigrator currently requires a key column to be identified to perform update operations. While Hive does not identify columns as keys, since bucketed columns cannot be updated, they should be so identified. Before creating a data flow to perform updates, edit the synonym for the target table and select the Key Component checkbox for the bucketed columns and any others required to uniquely identify the row.

Adapter for i Access

This section provides detailed descriptions of new features for the Adapter for i Access.

JDBC Adapter Configuration Available

A new SQLIIA JDBC-based adapter is available to enable access to the IBM i platform from UNIX, Windows, and Linux.
Adapter for Informix

This section provides detailed descriptions of new features for the Adapter for Informix.

**Adapter for Informix Supports Extended Bulk Load**

DataMigrator with a target table in Informix supports Extended Bulk Load using the native DBLOAD utility.

Adapter for Jethro

Jethro is an acceleration engine that makes real-time Business Intelligence work on Big Data. The adapter is available for ODBC (on Windows only) and JDBC (on Linux/Unix/Windows).

Adapter for MariaDB

The Adapter for MariaDB is new in this release and can be found under SQL Adapter on the Web Console. The adapter provides read/write access to data stored in the open source MariaDB database using the MariaDB Connector/J.

**Adapter for MariaDB: Change Data Capture (CDC) Support**

DataMigrator supports Change Data Capture (CDC) for MariaDB with Load Type IUD.

Adapter for Microsoft SQL Server

The following features are supported for Microsoft SQL Server in this release.

**Adapter for Microsoft SQL Server: Support for Computed Columns as R/Only**

Adapters for MS SQL (SQLMSS and MSODBC) now map SQL Server computed columns as read-only, with the FIELDTYPE=R attribute.

**JDBC and ODBC Adapters for Microsoft SQL Server Support Version 2016**

ODBC and JDBC Adapters for Microsoft SQL Server support the recently released Microsoft SQL Server version 2016, ODBC driver version 13, and JDBC driver version 6.

**ODBC Adapter for Microsoft SQL Server Supported on Red Hat Linux**

ODBC based access to MS SQL Server is provided from Linux Red Hat 6 on the Intel x64 platform. Performance savings for large answer set retrieval can reach 50% compared to JDBC access from Linux.
**ODBC Adapter for Microsoft SQL Server Support for Extended Bulk Load**

The Adapter for MSODBC Supports extended bulk loading via MS Bulk Copy utility on Windows and Linux.

**ODBC Adapter for Microsoft SQL Server Supports Change Data Capture for NVARCHAR(MAX) Fields**

When a column has the data type NVARCHAR(MAX), the Synonym Editor now creates the synonym with USAGE and ACTUAL formats A32765V when the Synonym Candidates are restricted to Table Log Records.

**Note:** The server must be configured for Unicode to properly report from National Fields.

If the length of the data is longer than 32765, the following message is generated:

(FOC1362) COLUMN name IS TRUNCATED IF DATA LENGTH EXCEEDS 32765

**Adapter for MySQL**

This section provides detailed descriptions of new features for the Adapter for MySQL.

**Change Data Capture (CDC) Support**

The Adapter for MySQL now supports Change Data Capture (CDC).

**Adapter for Netezza**

This section provides descriptions of new features for the Adapter for Netezza.

**Adapter for Netezza: Support for Bulk Load**

The Adapter for Netezza supports extended bulk loading using the nzload utility.

**Adapter for ODBC**

This section provides descriptions of new features for the generic ODBC Adapter.

**Adapter for ODBC: Support for Wide API**

The Generic ODBC Adapter has been upgraded to utilize the wide ODBC API.

**Adapter for Oracle**

This section provides detailed descriptions of new features for the Adapter for Oracle.
Support for Power Linux LE With Instant Client

The Adapter for Oracle is now supported on IBM Power Linux Little Endian systems using the Oracle Instant Client Basic package.

New Setting to Map FLOAT Data Type as DECIMAL

The following new Adapter for Oracle setting maps the Oracle FLOAT and FLOAT(126) data types to Float or to Decimal.

\[
\text{ENGINE SQLORA SET ORAFLOAT \{FLOAT|DECIMAL[p s]\} }
\]

where:

- **FLOAT**
  
  Maps the Oracle FLOAT and FLOAT(126) data types to USAGE=D20.2, ACTUAL=D8. This is the default value

- **DECIMAL[p s]**
  
  Maps the Oracle FLOAT and FLOAT(126) data types to Decimal with precision = p and scale = s. If precision and scale values are omitted, the precision defaults to 33 and the scale to 2.

This setting is expected to be used on case-by-case basis after consulting with Information Builders Technical Support Services. It compensates for the Oracle proprietary implementation of FLOAT and FLOAT(126) data types that deviate from the IEEE 754 standard.

Note that Oracle BINARY_FLOAT and BINARY_DOUBLE data types are not affected by the setting since these types conform to the IEEE 754 standard.

Connection to Database Running on Amazon RDS

The Adapter for Oracle can access Amazon RDS running an Oracle DB instance. This is implemented by using a long connection string, such as the following, instead of a TNS alias.

\[
(\text{DESCRIPTION=} (\text{ADDRESS=} (\text{PROTOCOL=} TCP) (\text{HOST=} dns\_of\_db\_instance) (\text{PORT=} listener\_port)) (\text{CONNECT\_DATA=} (\text{SID=} database\_name)))
\]

where:

- **dns\_of\_db\_instance**
  
  Is the DNS of the database instance.

- **listener\_port**
  
  Is the Oracle TCP listener port number.
Support for Extended Bulk Load

The Adapter for Oracle Supports Extended Bulk Load based on the Direct Path Load API.

Adapter for Oracle Reads XMLType Data Type Using DB_EXPR

The Oracle XMLType data type can be mapped with USAGE=TX50L, ACTUAL=TX. The Access File then specifies attributes needed for retrieving the data using the DB_EXPR function.

The following Master File declares a field named `xmltypefield` whose data type is XMLType:

```plaintext
FILENAME=ORAXML_SRZ, SUFFIX=SQLORA , $
SEGMENT=ORAXML_SRZ, SEGTYPE=S0, $
FIELDNAME=I, ALIAS=I, USAGE=I11, ACTUAL=I4, 
MISSING=ON, $
FIELDNAME=xmltypefield, ALIAS=X, USAGE=TX50L, ACTUAL=TX, 
MISSING=ON, $
```

The following Access File specifies the attributes for reading the XMLType field:

```plaintext
SEGNAME = segname, TABLENAME = tablename,$
FIELD=xmltypefield, 
SQL_FLD_OBJ_TYPE=OPAQUE, 
SQL_FLD_OBJ_PROP=, 
SQL_FLD_OBJ_EXPR='DB_EXPR(XMLSERIALIZE(DOCUMENT "xmltypefield"))',$
```

where:

```plaintext
SQL_FLD_OBJ_TYPE=OPAQUE
```

Indicates that the field can be processed only through DB_EXPR.

```plaintext
SQL_FLD_OBJ_EXPR='DB_EXPR(XMLSERIALIZE(DOCUMENT "xmltypefield"))'
```

Is the Oracle SQL Expression that reads the XML document stored in the field.

**Example:** Using an Oracle XMLType Field in a Request

The following request retrieves the XML document stored in column XMLElementFIELD for id=112.

```plaintext
TABLE FILE ORAXML_SRZ
PRINT XMLElementFIELD
WHERE I EQ 112
END
```
The following SQL is generated and passed to Oracle.

```
SELECT T1."I",
(XMLSERIALIZE(DOCUMENT T1."XMLTYPEFIELD"))
FROM PODOC T1
WHERE
(T1."I" = 112);
```

**Adapter for Oracle: Support for Long IN-list**

The Adapter for Oracle now splits a long IN-list generated, for example, when BY HIERARCHY is used in reports. This enables it to overcome the Oracle limit of 1000 elements and pass optimized SQL to the DBMS.

**Adapter for Apache Phoenix for HBase**

Apache Phoenix provides SQL read/write access to Apache HBase (and MapR-DB) a NoSQL database distributed with Hadoop.

For full functionality this adapter requires Phoenix release 4.5.0 or later.

**Adapter for SAP HANA**

The following features have been added for the Adapter for SAP HANA.

**Optimization of Simplified Functions POSITION and TOKEN**

Simplified functions POSITION and TOKEN are optimized by the Adapter for SAP HANA.

**Support for NCLOB/TEXT Data Types**

The Adapter for SAP HANA supports the TEXT data type in HANA column mode as format NCLOB.

**Adapter for Sybase**

This section provides detailed descriptions of new features for the Adapter for Sybase.

**Adapter for Sybase IQ Support for Unicode Extended Bulk Load**

The Adapter for Sybase IQ supports Unicode Extended Bulk Load.

**Adapter for Teradata**

This section provides detailed descriptions of new features for the Adapter for Teradata.
Support for Wide ODBC API

A new interface for Teradata is introduced that is based on the wide ODBC API. This interface supports Unicode and offers better performance on some platforms (for example, Windows and Linux x64/x86) compared to the Teradata ODBC interface that is not utilizing the wide API. This wide ODBC API interface is supported with TTU v.15.0 or higher.

READONLY Fields in Extended Bulk Load

Loading into Teradata using Extended Bulk Load is now supported even when the server metadata contains READONLY fields. Such fields will be skipped and remained unchanged during loading.

Distinguishing Between Macros and Stored Procedures

The following Access File attribute has been introduced to identify whether the synonym represents a Teradata Macro or Stored Procedure.

\[ \text{STPTYPE} = \{\text{MACRO}|\text{SP}\} \]

Support for Version 15.10

The Teradata CLI and ODBC adapters support Read/Write access to Teradata Version 15.10.

ERP Adapters

This section describes new features for ERP adapters.

Adapter for Microsoft Dynamics CRM

The Microsoft Dynamics CRM Online Adapter is new in this release and can be found under the ERP adapters. It allows reporting against the Microsoft Dynamics CRM Online environment, which is a customer relationship management software package that works in the Cloud. It enables companies to manage their prospect and customer account in a centralized way. The information that can reported against includes account, account leads, campaign activity, contacts, contact orders, and workflow.

Adapter for SAP/R3

The following have been added for the Adapter for SAP/R3.
Enhanced Function Module Functionality

To prevent and avoid the malicious injection of ABAP code in the context of a WebFOCUS query, function modules provided by Information Builders (DYNAMIC_RUN, REP_CREATE, REP_GET_B_D, REP_RUN) are enhanced to perform whitelisting of the dynamically generated ABAP4 code, by default. (See the SEC_CHK function module for the list of permitted commands).

Any command not whitelisted will be denied execution.

In addition, the function modules provided by Information Builders are enhanced with user exits by inserting calls to customer defined function modules on start (USR_EXT_ENT*), and exit (USR_EXT_EXT*). These user exits will be called with the same set of parameters as the original calling function module (IMPORT/TABLES only). A dummy version is supplied by Information Builders.

* A customer can modify these user exit function modules and implement all additions and authorizations (SAP authority checks) that are required.

GIS Adapters

This section provides detailed descriptions of new features for GIS adapters.

Adapter for Esri ArcGIS

The Adapter for Esri ArcGIS provides access to the Esri cloud for the purposes of drawing maps and demographic layers, and also providing the following geometry functions that retrieve geometry data and geocode addresses:

- **GIS_DISTANCE.** Calculates the distance between two geometry points.
- **GIS_DRIVE_ROUTE.** Returns the driving route between two geometry points.
- **GIS_GEOCODE_ADDR.** Returns the geometry point for a complete address.
- **GIS_GEOCODE_ADDR_CITY.** Returns the geometry point for address, city, and state.
- **GIS_GEOCODE_ADDR_POSTAL.** Returns the geometry point for address, postal_code.
- **GIS_GEOMETRY.** Builds/converts a JSON geometry object given a geotype, WKID (Well-Known ID) spatial reference, and a initial JSON geometry.
- **GIS_IN_POLYGON.** Given a geometry point and a polygon definition, returns the value 1 (TRUE), if the point is in the polygon or 0 (FALSE), if is not.
- **GIS_LINE.** Given two geometry points or lines, GIS_LINE builds a JSON line defining a geometry line.
- **GIS_POINT.** Given a WKID (Well-Known ID) spatial reference, longitude, and latitude, returns a JSON point defining a geometry object.

- **GIS_SERVICE_AREA.** Calculates the geometry area with access boundaries within the given time or distance from the provided geometry point.

- **GIS_SERV_AREA_XY.** Calculates the geometry area with access boundaries within the given time or distance from the provided coordinates and WKID spatial reference.

For detailed information about the geography functions, see BAD XREF HERE "Reporting Language Enhancements"

### Unified Definition for ArcGIS Server URLs and Geographic Roles

The server comes with a file named `geo_services.xml` that configures all ArcGIS geographic services that are available out of the box when the server is installed.

In order to correctly retrieve and use location data in a request, the geographic roles for the fields that represent locations must be accurately defined in the request or the metadata. Geographic roles may be part of a hierarchy. For example, a city may be part of a state, and the state part of a country. The hierarchical relationships are needed in order to identify and retrieve the correct location data.

Hierarchy definitions also provide WebFOCUS with the information needed to perform AutoDrilling from a geographic role at a higher level of the geographic hierarchy to a geographic role at a lower level of the geographic hierarchy.
When you use the Web Console Data Assist tool to assign a geographic role to a field, the hierarchies display their member roles within the appropriate contexts. For example, the following image shows the hierarchies available by default, with the World hierarchy open to its first level of member roles.

Note that, by default, the available hierarchies are the EU hierarchy, the US hierarchy, and the World hierarchy.

Customers with Enterprise Data often have map layers that represent their territories, events or logistical information. These are published as Map Services to either a subscription based in the Esri Cloud (ArcGIS.com) or on an internal portal. In order for these customized geographic roles to be available through the server, they must be added to the geo_services.xml file.
Configuring Geographic Roles

The geographic configuration file, `geo_services.xml`, is located in the catalog directory under the server home directory:

```
_edahome/catalog/
```

The server builds the calls needed to retrieve geographic roles dynamically using this configuration file. Each role definition in the configuration file, when used in a WebFOCUS request, generates metadata and a request that is sent to Esri.

A geographic role can be part of a hierarchy. For example, the World geographic role is at the top of a hierarchy that contains continents, countries, states, and cities. These hierarchies are also described in the `geo_services.xml` file.

To add a custom geographic role, you must add the necessary parameters for the geography to this file.

Following standard XML syntax rules, each element is enclosed in element start and end tags (`<elementname>`, `</elementname>`), and attribute values are enclosed in double quotation marks (").

Reference: Geographic Role Definitions

A geographic role is stored as `geo_role` element in the `geo_roles` object of the `geo_services.xml` file. A geographic role must be defined with:

- An ID that will identify the role in the configuration file.
- A format and length for the data to be returned.
- A role name.
- A display title for the role name (to appear as a selection in the front-end tools).
- An optional role format (if the role can have multiple formats, such as a name and an abbreviation).
- A display title for the format.
- A role type (GEOGRAPHY for polygons or GEOMETRY for points).
- An optional vocabulary rule element containing vocabulary elements for associating the role with a field in the metadata.
The following attributes define a geographic role.

**id**
Is an alphanumeric uppercase value, up to 50 characters, used to identify the geographic role.

**type**
Is the data type for the ID. Can be one of the following.

- "alpha" for alphanumeric data, formats An or In.
- "integer" for integer numeric data, format In.
- "numeric" for fractional numeric data, formats Pn.m, Dn.m, or Fn.m.
- "text" for text data, format TXn.

**value_size**
Is the optional number of characters in USAGE format length (any, if not set).

**role_name**
Is the name of the geographic role.

**role_name_title**
Is the title of the geographic role, to be displayed in the tools for selection.

**role_format**
Is an optional format for the geographic role, useful when the role can be referenced using multiple formats, such as a name, an ISO code, and an abbreviation.

**role_format_title**
Is an optional title for the format of the geographic role. It will be shown in parentheses along with the role title in the tools, for example, State (Abbreviation).

**geo_type**
Is one of the following predefined role types.

- "geography", for geographic objects such as country or state.
- "geometry", for geometry objects such as geometry point and geometry area.
- "coordinate", for coordinates such as latitude and longitude.
**vocabulary_rules**

Is an element that consists of a group of vocabulary elements that explicitly describe column names for the geographic role. These rules will be used to select the best geographic data for the role.

Elements in a rule are connected by the Boolean logic operation OR (only one needs to be satisfied). Each vocabulary element contains words enclosed with special characters. Words in the rule element are connected by the Boolean logic operation AND (all need to be satisfied).

A word may be prefixed and/or suffixed with the percent character (%), which is a placeholder for any sequence of characters. If an element contains more than one word, each word has to be prefixed by the character plus (+) or minus (−). Plus indicates that the word must be found in the column name. Minus indicates that word must not be found in the column name.

**Example:**  
**Sample Geographic Role Definitions**

The following defines the State Abbreviation geographic role. The role ID is USSTATE_ABBR. The role name is USSTATE with a role format of ABBR. The titles that show in the tools are US state (Abbreviation). The format is A2, and the vocabulary rules specify that the characters state must be present, but the characters iso, capital, and population must not be present. The geo type is geography, indicating that the returned data will be a geographic area.

```xml
<geo_role
  id="USSTATE_ABBR"
  value_size="2"
  type="alpha"
  role_name="USSTATE"
  role_name_title="US state"
  role_format="ABBR"
  role_format_title="Abbreviation"
  geo_type="geography">
  <vocabulary_rules>
    <vocabulary>%state%-%iso%-%capital%-%population%</vocabulary>
  </vocabulary_rules>
</geo_role>
```
The following is a role definition for latitude values. The role ID is LATITUDE. The role name is also LATITUDE. Its format is numeric. The title that displays in the tools is Latitude. The geo type is geometry, indicating that the returned data will be points or areas described using points. The vocabulary rules specify that the characters latitude must be present.

```xml
<geo_role
    id="LATITUDE"
    type="numeric"
    role_name="LATITUDE"
    role_name_title="Latitude"
    geo_type="geometry">
    <vocabulary_rules>
        <vocabulary>%latitude%</vocabulary>
    </vocabulary_rules>
</geo_role>
```

The following is the definition for the city role. The ID is CITY. The role name is also CITY. Its format is NAME. The title that displays in the tools is City (Name). The definition has a set of vocabulary elements. Only one of the elements in the list must be true. Therefore, the characters city, or town, or country plus capital, or state plus capital must be present.

```xml
<geo_role
    id="CITY"
    type="alpha"
    role_name="CITY"
    role_name_title="City"
    role_format="NAME"
    role_format_title="Name"
    geo_type="geography">
    <vocabulary_rules>
        <vocabulary>+%city%-%population% </vocabulary>
        <vocabulary>+%town%-%population% </vocabulary>
        <vocabulary>+%country%+%capital%-%population% </vocabulary>
        <vocabulary>+%state%+%capital%-%population% </vocabulary>
    </vocabulary_rules>
</geo_role>
```

**Reference:** Geographic Hierarchy Definitions

Some geographic roles exist as part of a hierarchy, and the data for the hierarchical roles are stored at the same map services endpoint (URL). Hierarchical role relationships are stored as hier elements in the geo_services.xml file.

Each hierarchy definition in geo_services.xml has the name of the hierarchy (attribute ID) and a group of LEV elements with the attributes level, geo_role, and, optionally, value. Not all defined roles can be used in hierarchies. The same role can be included in more than one hierarchy and may be on different hierarchical levels in each. However, the same role cannot be used more than once in the same hierarchy. Multiple geographic roles can be assigned to the same hierarchical level in a hierarchy.
Geographic hierarchies are defined with the following attributes:

**id**

Is a name of up to 50 alphanumeric characters used to identify the hierarchy.

**level**

Is a natural number (integer, starting with 1 for the top level) that specifies the level of the role within the hierarchy.

**geo_role**

Is the ID attribute of a geographic role (geo_role element).

**value**

Is an alphanumeric value, up to 50 characters, predefined for this geo role in this hierarchy.

**Example:** Sample Geographic Hierarchy Definition

The following element defines the world hierarchy. The top level is CONTINENT, both the Name role and the ISO code role. Level 2 has four COUNTRY geographic roles, corresponding to four different country formats. Level 3 contains three state formats, level 4 contains the city name, and level 5 contains two address formats and the postal code.

```xml
<hier id="World">
  <lev level="1" geo_role="CONTINENT"/>
  <lev level="1" geo_role="CONTINENT_ISO2"/>
  <lev level="2" geo_role="COUNTRY"/>
  <lev level="2" geo_role="COUNTRY_FIPS"/>
  <lev level="2" geo_role="COUNTRY_ISO2"/>
  <lev level="2" geo_role="COUNTRY_ISO3"/>
  <lev level="3" geo_role="STATE"/>
  <lev level="3" geo_role="STATE_ISO_SUB"/>
  <lev level="3" geo_role="STATE_FIPS"/>
  <lev level="4" geo_role="CITY"/>
  <lev level="5" geo_role="ADDRESS_FULL"/>
  <lev level="5" geo_role="ADDRESS_LINE"/>
  <lev level="5" geo_role="POSTAL-CODE"/>
</hier>
```

**Reference:** Adding the Federal Reserve Districts Geographic Role

These steps describe how to add the Federal Reserve Districts geographic role to the geo_services.xml file.

1. Open the geo_services.xml file. The default location on Windows is:

   \[C:\ibi\srvnn\home\catalog\geo_services.xml\]

   where \(nn\) is your release of the server. For example, 77 for Release 7707.
2. Add the role to the end of the GEO_ROLES object:

```
<geo_role id="FED-DIST" value_size="50" type="alpha" role_name="FED-DIST" role_name_title="FED District" role_format="FR_Distric" role_format_title="FED District Name" geo_type="geography">
    <vocabulary_rules>
        <vocabulary>+%FR_Distric%</vocabulary>
    </vocabulary_rules>
</geo_role>
```

The ID is FED-DIST. The role name is also FED-DIST. Its format is FR_Distric. The title that displays in the tools is FED District. The definition has a vocabulary rule. The characters `FR_Distric` must be present.

3. Add this role to the US Hierarchy:

```
<hier id="US">
    <lev level="1" value="United States" geo_role="COUNTRY"/>
    <lev level="1" value="US" geo_role="COUNTRY_ISO2"/>
    <lev level="1" value="USA" geo_role="COUNTRY_ISO3"/>
    <lev level="2" geo_role="USSTATE"/>
    <lev level="2" geo_role="USSTATE_ABBR"/>
    <lev level="2" geo_role="USSTATE_FIPS"/>
    <lev level="3" geo_role="USCOUNTY"/>
    <lev level="3" geo_role="USCOUNTY_FIPS"/>
    <lev level="4" geo_role="USCITY"/>
    <lev level="4" geo_role="USCITY_FIPS"/>
    <lev level="5" geo_role="ADDRESS_FULL"/>
    <lev level="5" geo_role="ADDRESS_LINE"/>
    <lev level="5" geo_role="ZIP3"/>
    <lev level="5" geo_role="ZIP5"/>
    <lev level="6" geo_role="FED-DIST"/>
</hier>
```

4. Add the URI to the map server layer for this role at the end of the URIS object:

```
<uri description="FedReserve Districts">
    <returned_geometry>GEOMETRY_AREA</returned_geometry>
    <returned_georole>FED-DIST</returned_georole>
    <url type="esri" authorization="none" synonym="">
        http://services7.arcgis.com/L95Ww90jRQ0tjAs/arcgis/rest/services/FRDISTRICTS/FeatureServer/0</url>
    <parameters>
        <parm order="1" parm_name="FR_Distric" parm_georole="FED-DIST"/>
    </parameters>
</uri>
```

This role will now be available to use in a request and will display in the tools.

**OLAP Adapters**

This section provides detailed descriptions of new features for OLAP adapters.

104  Information Builders
**Adapter for Essbase**

The Adapter for Essbase now supports Essbase 11.1.2.3 and 11.1.2.4 on the Windows 64-bit platform.

**Adapter for Microsoft SQL Server Analysis Services (SSAS)**

You can use the DB_EXPR function to calculate temporary measures and use currency and percent display options for measures.

**Creating Temporary Calculated Measures Using DB_EXPR**

You can use the DB_EXPR function to generate a query-scope calculated measure as an MDX expression. The result of that expression must be type value. In the calculated measure expression, you can use any MDX function.

You are responsible for how the expression will work in the context of the whole generated MDX query. For example, note that if the expression references the [Calendar] hierarchy, the result will depend on which (if any) level of that hierarchy is referenced in a BY clause of the TABLE request. If the level is not taken into account, the query may produce unexpected results or errors.

**Example: Using MDX Functions in a Calculated Measure**

The following request references three MDX functions to define the ISA_YTD measure, one member type function (CURRENTMEMBER), and two value type functions (Sum, Ytd).

```plaintext
DEFINE FILE ADVENTURE_WORKS_MIXED_CASE
ISA_YTD/D20.2 WITH Internet_Sales_Amount =
DB_EXPR(Sum(Ytd([Date].[Calendar].CURRENTMEMBER),
[Measures].[Internet Sales Amount]));
END
TABLE FILE ADVENTURE_WORKS_MIXED_CASE
SUM Internet_Sales_Amount
ISA_YTD
BY Fiscal_Year NOPRINT
BY Fiscal_Quarter
ON TABLE SET PAGE NOPAGE
END
```
The following MDX request is generated, which includes the calls to the MDX functions.

WITH
MEMBER [Measures].[ISA_YTD] as 'Sum(Ytd( [Date].[Calendar].CURRENTMEMBER ), [Measures].[Internet Sales Amount])'
SET H24 as '[Date].[Calendar].[Calendar Quarter].ALLMEMBERS'
SET H33 as '[Date].[Fiscal].[Fiscal Year].ALLMEMBERS'
SELECT {
[Measures].[Internet Sales Amount],
[Measures].[ISA_YTD]}
ON AXIS(0),
NON EMPTY CROSSJOIN(H24, H33)
DIMENSION PROPERTIES
MEMBER.Caption
,[Date].[Calendar].[Calendar Quarter].[Fiscal Quarter] ON AXIS(1)
FROM [Adventure Works]

The output is shown in the following image.
The following request references three MDX functions to define the measure $I_{S\_A\_4\_MONTHS\_ROLLING\_AVERAGE}$, two member type functions (CURRENTMEMBER and Lag), a range (:), and one value type function (Avg).

```mdx
DEFINE FILE ADVENTURE_WORKS_MIXED_CASE
$I_{S\_A\_4\_MONTHS\_ROLLING\_AVERAGE}/D20.2M$
WITH Internet_Sales_Amount =
  DB_EXPR(Avg({ [Date].[Calendar].CURRENTMEMBER.Lag(3) :
    [Date].[Calendar].CURRENTMEMBER },
    [Measures].[Internet Sales Amount]));
END
TABLE FILE ADVENTURE_WORKS_MIXED_CASE
SUM Internet_Sales_Amount
$I_{S\_A\_4\_MONTHS\_ROLLING\_AVERAGE}$ AS 'Rolling Average'
BY Fiscal_Year NOPRINT
BY Fiscal_Quarter
ON TABLE SET PAGE NOPAGE
END
```

The following MDX request is generated, which includes the calls to the MDX functions.

```mdx
WITH
MEMBER [Measures].[I_S_A_4_MONTHS.Rolling_Average] as '
  Avg({ [Date].[Calendar].CURRENTMEMBER.Lag(3) :
    [Date].Calendar.CURRENTMEMBER },
    [Measures].[Internet Sales Amount])'
SET H24 as ' [Date].Calendar.[Calendar Quarter].ALLMEMBERS'
SET H33 as ' [Date].[Fiscal].[Fiscal Year].ALLMEMBERS'
SELECT {
  [Measures].[Internet Sales Amount],
  [Measures].[I_S_A_4_MONTHS.Rolling_Average]}
ON AXIS(0),
NON EMPTY CROSSJOIN(H24, H33)
DIMENSION PROPERTIES
MEMBER_CAPTION
,[Date].[Calendar].[Calendar Quarter].[Fiscal Quarter] ON AXIS(1)
FROM [Adventure Works]
```
Support for Display Options M, p, and %

The following display options are now supported and are used when generating synonyms for the sample data sources provided with SQL Server Analysis Services.

- **M**, which places a floating currency symbol to the left of a measure.
- **p**, which converts a measure to a percent by multiplying it by 100, and then displays it with a percent symbol on the right.
- **%**, which does no conversion, but displays the existing measure with a percent symbol on the right.

<table>
<thead>
<tr>
<th>Fiscal Quarter</th>
<th>Internet Sales Amount</th>
<th>Rolling Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 FY 2002</td>
<td>$1,453,522.89</td>
<td>$1,453,522.89</td>
</tr>
<tr>
<td>Q2 FY 2002</td>
<td>$1,812,850.77</td>
<td>$1,633,186.83</td>
</tr>
<tr>
<td>Q3 FY 2002</td>
<td>$1,791,698.45</td>
<td>$1,686,024.04</td>
</tr>
<tr>
<td>Q4 FY 2002</td>
<td>$2,014,012.13</td>
<td>$1,768,021.06</td>
</tr>
<tr>
<td>Q1 FY 2003</td>
<td>$1,396,833.62</td>
<td>$1,753,848.74</td>
</tr>
<tr>
<td>Q2 FY 2003</td>
<td>$1,327,799.32</td>
<td>$1,632,585.88</td>
</tr>
<tr>
<td>Q3 FY 2003</td>
<td>$1,413,530.30</td>
<td>$1,538,043.84</td>
</tr>
<tr>
<td>Q4 FY 2003</td>
<td>$1,623,971.06</td>
<td>$1,440,533.57</td>
</tr>
<tr>
<td>Q1 FY 2004</td>
<td>$2,744,340.48</td>
<td>$1,777,410.29</td>
</tr>
<tr>
<td>Q2 FY 2004</td>
<td>$4,009,218.46</td>
<td>$2,447,765.07</td>
</tr>
<tr>
<td>Q3 FY 2004</td>
<td>$4,283,629.96</td>
<td>$3,165,289.99</td>
</tr>
<tr>
<td>Q4 FY 2004</td>
<td>$5,436,429.15</td>
<td>$4,118,404.51</td>
</tr>
<tr>
<td>Q1 FY 2005</td>
<td>$50,840.63</td>
<td>$3,445,029.55</td>
</tr>
</tbody>
</table>
**Example:** Using the Display Options M, p, and %

The following request prints `INTERNET_SALES_AMOUNT` twice, first with no currency symbol, and then with a currency symbol (M option). It prints `GROSS_PROFIT_MARGIN` three times, first as a number, then as a percent (p option), and last as a number with a percent symbol (% option).

```
TABLE FILE ADVENTURE_WORKS_UPPER_CASE
PRINT
INTERNET_SALES_AMOUNT/P20.2 AS 'P20.2'
INTERNET_SALES_AMOUNT/P20.2M AS 'P20.2M'
GROSS_PROFIT_MARGIN/F10.4 AS 'F10.4'
GROSS_PROFIT_MARGIN/F10.2p AS 'F10.2p'
GROSS_PROFIT_MARGIN/F10.4% AS 'F10.4%'
BY COUNTRY
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

![Table Output]

**Data Analysis Expression (DAX) Passthrough**

A sample DAX query can be executed using the existing MDX Passthru mechanism. The recommended way is to place a DAX query into a file with extension `.sql` and create a synonym for it. The synonym can be further edited for attributes such as proper formatting. Reporting is then done against that synonym.
Adapter for TM1

The Adapter for TM1 is a special configuration of the Adapter for SQL Server Analysis Services. Configure the Adapter for TM1 by selecting TM1OLAP Provider on the Change Settings for SQL Server Analysis Services page of the Web Console. You must also provide the default database and, possibly, additional connection string parameters on the Change Connect Parameters for SQL Server Analysis Services Web Console page. The Adapter for TM1 supports Cognos Access Manager (CAM) authentication and Integrated Login.

CAM Authentication Support

To configure Cognos Access Manager (CAM) authentication, you must add the CAMNamespace parameter to the connection string.

You can configure this parameter on the Web Console. To configure the Adapter for TM1, use the Add Connection page for the Adapter for SQL Server Analysis Services, as shown in the following image.

For TM1, the catalog to be accessed must be part of the connection string. Enter the catalog name in the Default Database text box in order to add it to the connection string.

Enter the CAMNamespace parameter in the Additional connection string keywords text box. It will be added to the connection string as the ProviderString parameter when you press Configure. For example, if you enter CAMNamespace=NTLM_NAMESPACE, a connection string similar to the following is generated in the edasprof.prf server profile:

```
ENGINE SSAS SET CONNECTION_ATTRIBUTES CON1
SQL2012x64-02/user1,pwd1;SData:CAMNamespace=NTLM_NAMESPACE
```

The ENGINE SSAS SET PROVIDER TM1OLAP command must be in effect when using the Adapter for TM1. This command can be issued in edasprof.prf or in a FOCEXC.

Integrated Login Support

Integrated Login enables you to use Microsoft Windows network authentication to control access to IBM Cognos TM1 data.
In this security model, user and group Microsoft Windows login information has to be moved into the Cognos TM1 database. Integrated Login matches the domain-qualified name you use to sign in to Microsoft Windows with a name stored in the internal database.

Integrated Login is supported on Microsoft Windows only.

You can configure Integrated Login in the Web Console, using the Add Connection page of the Adapter for SQL Server Analysis Services. When you configure the Adapter for TM1, select a trusted connection, as shown in the following image.

The connection string must specify the name of the TM1 catalog to be accessed. Enter the catalog name in the **Default Database** text box. A connection string similar to the following will be generated in the edasprof.prf server profile.

```
ENGINE SSAS SET CONNECTION_ATTRIBUTES CON1
   SQL2012x64-02/,;SData
```

Since the connection is configured as trusted, no user ID or password is included in the connection string. The Integrated SSPI login causes the credentials from the Windows login to be used to access TM1.

The ENGINE SSAS SET PROVIDER TM1OLAP command must be in effect when using the Adapter for TM1. This command can be issued in the Server Profile or in a FOCUS procedure.

**DBMS Adapters**

This section provides detailed descriptions of new features for the DBMS adapters.

**Adapter for Salesforce JOIN Support**

Child-to-parent retrieval (reversed join) queries are now enabled for custom objects. This feature adds support for queries with multiple objects.

**Procedures Adapters**

This section provides detailed descriptions of new features for the Procedures adapters.
**Adapter for REST**

This section provides detailed descriptions of new features for the Adapter for REST.

**Adapter for REST/WS: Pass Name:Value Pairs in the HTTP Header**

Name:Value pairs can now be passed in the HTTP Header of a SOAP/REST request.

For a SOAP request, add the following attributes in the Access File.

```
FIELD=fieldname, PARAMSTYLE=header,
```

Note that the field belongs to the root segment, but not in the header group (unlike with the REST adapter). The fields in the header group will go to the SOAP header, not the HTTP header.

For the REST adapter, the fields in the header group, if specified as PARAMSTYLE=header, go to the HTTP header, otherwise they are added to the URL.

The field to be placed in the HTTP header should be positioned within the header group of the root segment.

The following Access File record indicates the header group.

```
HEADER=groupname
```

The following declaration indicates the field whose value is to be placed in the HTTP header.

```
FIELD=fieldname, PARAMSTYLE=header, $
```

In the Master File, identify the value to be placed in the header for the field.

For example, consider the following Access File.

```
SEGNAME=M6ILO,
 HEADER=GRHEADER,
 ...$ FIELD=TOKEN, PARAMSTYLE=header, $
```

The Master File should contain the following attributes.

```
GROUP=GRHEADER, ELEMENTS=1, ACCESS_PROPERTY=(INTERNAL),
 TITLE='GRHEADER', $
 FIELDNAME=TOKEN, ALIAS=token, USAGE=A80, ACTUAL=A80,XDEFAULT='oxxxxx', $
```

This causes the following to be placed in the HTTP header.

```
token: oxxxxx
```
Adapter for REST: Support for GZIP Response Format
The Adapter for REST now supports responses in GZIP Format.

Adapter for REST: Support for Requests not Containing Service URL Extension
Support has been added for REST Web Services calls that do not have a service URL extension.

Adapter for REST: Support for Token as Attribute in Chained Authentication
Support has been added for retrieving a token from a Chained Authentication request when it resides as an attribute or as a child segment with multiple entries.

Adapter for REST/WS: Type Needed for Elements when OBJ_TYPES=NO
In the Web Services adapter, xsi:type will be sent for specific elements in the SOAP request based on the defined FIELDNAMES in the Access file, even though sending xsi:types is turned off in the Access file (OBJ_TYPES=NO).

Adapter for REST/WS: Support for Attribute Referencing
Create Synonym for the Web Services adapter now supports attribute referencing.

Adapter for RStat Predictive Analytics
The Adapter for RStat Predictive Analytics is new in this release and can be found under Procedure Adapters on the Web Console. The adapter allows RStat models to be deployed from the Reporting Server Console, in addition to model deployment from App Studio. A Master File and Access File are also created for the deployed model, enabling access through a Master File Join or the function DB_LOOKUP().

Adapter for Web Services
The following features were added for the Adapter for Web Services.

Microsoft Dynamics AX 2012 Integration
Microsoft Dynamics AX 2012 is an ERP system. It offers complete functionality for accounting, inventory control, warehouse management, transportation management, demand and supply chain planning, material requirements planning, production management, quality assurance, product lifecycle management, and asset management.

Using the Web Service adapter, reports can be created to report against the data within the Dynamics AX environment while respecting its security.
Sequential and Indexed Adapters

This section provides detailed descriptions of new features for the Sequential and Indexed adapters.

Reading Virtual Fields Through the File Listener

When using the File Listener with a pick up mode of MONITOR there are virtual fields available that in previous releases were only available with pickup strategy MONITOR, (which does not actually read the files). They are now also available when reading the file with pickup strategy IMMEDIATE.

Special Characters Supported With Upload Wizard

Upload Wizard now allows the special characters dot (.) and slash (/) when you explicitly redefine the target DBMS table name, in order to support multi-part schema/owner based names. The library specification (for example, mylib/mytable or mylib.mytable) separator is changed to an underscore (for example, mylib_mytable).

Adapter for Excel (via Direct Retrieval)

This section provides detailed descriptions of new features for the Adapter for Excel (via direct retrieval).

File Listener for Excel Worksheet

The File Listener component of DataMigrator can process Microsoft Excel workbooks that are delivered to a specified directory.

In addition to the synonym for the workbook as Excel (via direct read), this requires a synonym for a flat file with the normal file listener parameters specified. An additional segment contains a copy of the synonym for the Excel workbook.

Adapters for DFIX and Excel (via Direct Retrieval)

This section provides descriptions of features for the Adapters for DFIX and Excel (via direct retrieval).

Date and Date-Time Recognition With Patterns

Previously, when creating a synonym for a Delimited Flat File or Excel workbook using the direct read adapter, columns with timestamp or date-time values were created as alphanumeric fields.
Now such columns are recognized, and the synonym is created with appropriate USAGE and ACTUAL formats.

**Adapters for Flat and Delimited Flat Files**

The following features have been added for the Adapters for flat files and delimited flat files.

**Adapter for DFIX: Respecting Server CDN option**

The CDN parameter can be set in the server profile, in the Access File for a delimited data source, or in a request. When you upload a delimited file to the server, you can select the CDN option for that file from a drop-down list, and it is placed in the Access File. The supported CDN values are the following.

- **OFF**
  - The thousands separator is a comma (,), and the decimal separator is a period (.). This is the default value.

- **ON**
  - The thousands separator is a period (.), and the decimal separator is a comma (,).

- **SPACE**
  - The thousands separator is a space, and the decimal separator is a comma (,).

- **QUOTE**
  - The thousands separator is a single quotation mark ('), and the decimal separator is a comma (,).

- **QUOTEP**
  - The thousands separator is a space, and the decimal separator is a period (.).

**Adapter for RMS**

The following features have been added for the Adapter for RMS.

**Direct 64bit DEC TIMESTAMP Support**

The OpenVMS Adapter for RMS now supports the direct read and write of native OpenVMS 64-Bit DEC TIMESTAMP data using H date-time formats for USAGE and ACTUAL formats when the following command is in effect.

```
SET VMSTIMESTAMP = VMS
```

This eliminates the need for the WRTTIME() and CVTSTIME() subroutines in the writing and displaying of dates stored in this format in MODIFY and other applications.
Any H date is valid, but more commons ones follow.

The following displays as 2004/12/01 15:18:17.610.

...,USAGE=HYYMDs,ACTUAL=HYYMDs, ...

The following displays as 2004/12/01 15:18:17.

...,USAGE=HYYMDs,ACTUAL=HYYMDs, ...

The following displays as 01-Dec-2004 15:18:17.61.

...,USAGE=HDMtYY-2,ACTUAL=HDMtYY-2, ...

The format HDMtYY-2 most closely mimics the OpenVMS 64-Bit TIMESTAMP format (as displayed by commands such as DIRECTORY/DATE), except that the month is displayed in mixed case.

If your server application only uses RMS files, SET VMSTIMESTAMP = VMS command should be added to the server EDASPROF.PRF profile (which is in the EDACONF etc directory).

Some SQL examples follow.

```
SQL INTO MYTABLE
VALUES ('123456789','2015-07-08 12:08:07.12') ;
END

SQL UPDATE MYTABLE
SET SOLD_TIMESTAMP='2015-07-07 12:08:07.12'
WHERE ACCT='123456789' ;
END

SET INSERTSIZE=n for Insert and Update
```

Support for the command SQL SET INSERTSIZE=n for prepared SQL UPDATE, INSERT, and DELETE statements against non-SQL Data Adapters has been implemented to improve performance.

The feature is only active for prepared SQL logic, such as the following.

```
SQL PREPARE SQLIN FROM
INSERT INTO MYTABLE (FIELD_KEY,FIELD_A) VALUES (?,?)
END
SQL BIND SQLIN USING INTEGER, VARCHAR(10);
END
SQL EXECUTE SQLIN USING 1, 'One';
END
```

Internally, records will be collected and then executed (under MODIFY) when the INSERTSIZE is reached, or sooner if a COMMIT, PURGE, or different SQL statement is encountered.
The feature is not active for non-prepared SQL statements such as `INSERT INTO INTO MYTABLE VALUES (1,'One')`.

SQL requests sent using JLINK applications must also set `setAutoCommit(false)` on the Java connection to override the `java.sql.Connection` behavior of sending commits after each transaction.

**Update Locking Behavior**

The OpenVMS Adapter for RMS now allows the locked record wait period for `LOCKMODE=KEEP` to be set and, in order to match the behavior when `LOCKMODE` is not declared for the RMS file, the setting `LOCKMODE=REJECT` can be invoked. If an explicit wait period is not set for `KEEP`, the default wait period is 10 seconds, as it was prior to the implementation of this feature.

The syntax is

```
ENGINE RMS SET WAIT_LOCKED n
```

where:

```
n
```

Is the number of seconds to wait.

**Social Media Adapters**

This section provides detailed descriptions of new features for the Social Media adapters.

**Adapter for Google Analytics**

Enhancements have been made to the filtering capabilities within the Adapter for Google Analytics.

For example.

The following filter returns records where `/products` is at the beginning of the page path.

```
WHERE PAGEPATH LIKE '/products%'
```

The following filter returns records where `/products` is at the end of the page path.

```
WHERE PAGEPATH LIKE '%$/products'
```

The following filter returns records where `/products` exists anywhere within the page path.

```
WHERE PAGEPATH LIKE '%$/products$'
```
The following filter will not return records where /products is at the beginning of the page path.

WHERE PAGEPATH NOT LIKE '/products%'

The following filter will not return records where /products is at the end of the page path.

WHERE PAGEPATH NOT LIKE '%/products'

The following filter will not return records where /products exists anywhere within the page path.

WHERE PAGEPATH NOT LIKE '%/products%'

The following filter returns records where /products exists anywhere within the page path.

WHERE PAGEPATH CONTAINS '/products'

The following filter will not return records where /products exists anywhere within the page path.

WHERE PAGEPATH OMITS '/products'

The following filter returns records where the USERS value is greater or equal to 30 and less than or equal to 800.

WHERE USERS FROM 30 TO 800

**Adapter for Facebook: Enhanced Create Synonym Functionality**

Version 2.7 of the Facebook API is now supported.

**Social Media Demonstration Application**

In WebFOCUS Release 8.2 Version 01M and higher, a social media demonstration ("demo") application is available for deployment, which highlights the capabilities of WebFOCUS Social Media Integration when performing various levels of social media analysis.

The first level of analysis is known as the engagement level, where you analyze if, when, and how much activity took place. As companies mature, they start to focus on not just the activity but the actual conversations themselves.

WebFOCUS Social Media Integration offers techniques such as sentiment and word frequency analysis to help quantify and visualize this textual data to more effectively see trends and patterns in the data.
The second level of analysis is integrating your social data with enterprise data to better understand social impact on business outcomes. The data analysis that is used in this demo was obtained when Walmart was running a very specific campaign around their meat products. Note that the sales data was fabricated to provide better understanding, especially since there was no conclusive access to their real sales data. However, the social data is real.

The WebFOCUS social media demo files are located in the following folder of your WebFOCUS installation:
<drive>:\ibi\WebFOCUS82\samples\social_media_demo

For more information on installing and using the social media demo that is packaged with WebFOCUS, see the WebFOCUS Social Media Integration Guide.

XML-Based Adapters

This section provides detailed descriptions of new features for the XML-based adapters.

**Adapter for JSON**

The following features have been added for the Adapter for JSON.

**Adapter for JSON Support for UPDATE**

Using DataMigrator, data can be loaded into new JSON targets in a data flow.

**Support for ON TABLE HOLD FORMAT JSON**

A report can now save output in JSON (Java Script Object Notation) format.

```
ON TABLE HOLD [AS name] FORMAT JSON
```

A JSON file and accompanying synonym are generated.

**Ability to Set Encoding in the XML Header for XML Write**

Previously, the XML Write Adapter always created the XML file with encoding="UTF-8" in the XML header. Now encoding can be specified in the synonym and is written to the XML header.

**Redirection Support for HTTP Status Codes 302, 303, and 307**

Support has been added for HTTP calls made from the XML, Web Services, REST and Social Media adapters that require a redirection to a different URI. This support includes HTTP status code responses 302, 303, and 307.
DataMigrator Enhancements

This section describes the new features for DataMigrator.

DataMigrator represents a broad category of tools designed to facilitate and automate the extraction and integration of data. From source extraction through target load, data is transformed through the application of business rules. Once the transformation is complete, the data is loaded into table structures that have been optimized for a particular application.

For more information on any of these new features, see the DataMigrator User’s Guide.

In this chapter:

- Adapters
- Calculators
- Data and Process Flows
- Data Management Console
- Data Profiling
- Quick Copy
- Reporting
- Synonym Editor

Adapters

The following section provides descriptions of new features for adapters.

Change Data Capture for MySQL

The DataMigrator support for Change Data Capture now includes the MySQL database.

In order to use CDC with MySQL, you must install Java and MySQL client utilities on the system where the DataMigrator server is installed. Additionally, on that system, you must have the Environment Variable CLASSPATH pointing to the location where the Connector/J is installed.
Connection for [Delimited] Flat File Targets

In prior releases when using DataMigrator to write a Flat File or Delimited Flat file the first configured connection was always used.

There is now an option to select `<local>` for the local file system where the DataMigrator server is running. You can also select a connection that refers to an FTP or SFTP server that has been configured for the server.

Extended Bulk Load Availability

Extended Bulk Load is now available on EXASol, Informix, MS SQL Server, Netezza, ORACLE, and PostgreSQL.

Extended Bulk Load for Hive and Impala

Data can now be loaded to HDFS using Extended Bulk Load. When the Hive or Impala server is remote, FTP is used to move the file to load.

File Attributes from File Listener

There are now virtual fields available when using the File Listener with the MONITOR pick up mode. In previous releases, these fields were only available with a pickup strategy of MONITOR, which does not actually read the files. They are now also available when reading the file with a pickup strategy of IMMEDIATE.

The available virtual fields include:

**INSTANCE_FULL**

The full path and name of the file. This field cannot be used in conjunction with INSTANCE.

**INSTANCE_SIZE**

The size of the file in bytes.

**INSTANCE_DATE**

The date that the file was last modified.

**INSTANCE_TIME**

The time that the file was last modified.

Additional fields are available for UNIX and Linux systems.

Support SFTP Source from a DM Server on Windows
Unlike Linux, Windows does not include an SFTP client program. However, there are programs available that implement the protocol. You can use an SFTP server as a source or target from a DataMigrator Server running on Windows by installing a SFTP client program, such as PuTTY.

**MS SQL Server ODBC Adapter in Linux**

The MS SQL Server ODBC adapter is now supported on Linux Red Hat 6 Intel x64 platform, in addition to Windows.

**Change Data Capture for MySQL and MariaDB**

The DMC now supports Change Data Capture (CDC) for both MySQL and MariaDB adapters with a load type of IUD.

**Calculators**

The following section provides descriptions of new features for calculators.

**Allow a Variable in WHERE Filter for IN**

In the Data Management Console, you can now use a variable in a Data Flow with a WHERE filter that contains an IN clause. For example, you could create a WHERE filter such as:

```
PLANT IN (&PLIST)
```

When you run the flow, you could use the following values:

'`BOS', 'ATL'`

These values would be used to return rows with either value.

**Notifications for Format Auto Correct**

Under certain conditions, when working in the DMC Synonym Editor or a Data Flow calculator, the format of a new field is recalculated based on the expression. A new balloon tip has been added, which informs users that the format has been recalculated.

**Enhanced DMTRUNC Function**

The simplified function DTRUNC, which returns the first day of a time period (YEAR, MONTH, QUARTER), has been extended to return the first day of a WEEK and the last day of any time period.

**Optimization of REPLACE Function**
A new, simplified REPLACE function has been introduced. This function has the following syntax:

```
REPLACE ( string, pattern, replacement )
```

The function will replace or remove a substring within a string. The input parameters can have CHAR, VARCHAR, or TEXT as data types. The result is always VARCHAR, with a length that is determined from the input parameters.

**Using a Regular Expression in a SQL WHERE Condition**

A Regular Expression can now be used in WHERE conditions in a SQL SELECT statement. A Regular Expression, or regexp, is a special sequence of characters that describe a search string. The syntax is:

```
string [NOT] RLIKE regexp
```

**Data and Process Flows**

The following section provides descriptions of new features for data and process flows.

**CMRUN on All Platforms**

The CMRUN program, which DataMigrator uses to initiate flows or stored procedures, takes an optional parameter of x=filename. This parameter allows you to store the password in a protected file.

In previous releases, the parameter was a DDNAME and only used on IBM zSeries (MVS).

Now it can be used on any other platform, including Windows and UNIX, where it is specified as a file name in the current directory or as the relative or full path to a file name.
Disallowing Multiple Instances of a Flow

When working with a Process Flow that initiates another data flow, you can now prevent the called flow from starting if another instance of the flow is running, by selecting the *Do not allow multiple instances* check box. This option is available in the Properties pane of a flow object.

Expand and Collapse Functionality for Objects

In the DMC, when viewing a flow with expanded objects, multiple objects can be selected by pressing the Ctrl key while clicking them.

If there are no objects selected in either a data or process flow, the new Expand All or Collapse All options are available.

Long Names in Flows

In the Data Management Console, if the name of an object is too long to fully display, it is truncated and shown with ellipses. You can see the full name by hovering over the object.

Optimize Load Option to Support MERGE

DataMigrator has a new option that allows for faster loading when a source and target are in the same relational database, resulting in an increase in performance. When working with:

- Databases that support the MERGE command, the merge is generated and passed through to the database so that both the extract and load is done in a single SQL statement. This option allows specifying different calculations for insert and update operations. Updates can use values from both the source and target tables. Databases that support a SQL MERGE command include MS SQL Server 2008 and Db2 9.1 (and later).
Insert only operations, an INSERT INTO ... SELECT command is generated and passed through to the relational database.

In either case, because all operations are done in the database, a much faster throughput is possible.

You can enable this option by opening the flow properties of a data flow and selecting the Optimize Load checkbox under the Execution section.

**Note:** Since the operation is done as a single SQL command rather than row-at-a-time processing, some capabilities are not available when this option is selected. These include Record Logging and use of VALIDATE to reject individual rows. Also, a count of the number of rows inserted, updated, or deleted is not available.

The user should consider both modes of operation to determine which is best suited for their application.

### Specifying Display Name in Target Properties

Target Properties has a new attribute, Display Name, that allows user specification of the name that appears on the flow. The default is the application name, if enabled in flow properties, and the synonym name.

### Updates to Email Functionality

The following new features are available for emails:

- **Email File Types for a Process Flow**
  Any file type can be attached to a DataMigrator process flow email.

- **Email Recipients**
  In previous releases, when sending email messages, all recipients appeared in the To line. Now, recipients can also appear on the Cc line, or hidden as Bcc.

- **Sending HTML Output on Message Failure**
  The Send HTML output as inline message located in the Properties panel of a Process Flow Start object now allows you to send HTML output from a report in an email message on failure, as well as on successful completion.
Updating Values in Target Transformations

You can now update, instead of replace, values in a target table using the On Match Expressions tab that can be found in the Transformations dialog box. For example, you can add the value from an incoming row to the value in the target table instead if replacing it completely.

Data Management Console

The following section provides descriptions of new features for the Data Management Console (DMC).

Downloading Files

A new Download option is available for all files on Application tree in the Data Management Console and Web Console. When a user downloads files, they are saved in the Downloads folder of their computer. In Windows, this location is C:\users\username\Downloads, by default.

Improvements in Column Name Display Strategy

When the strategy is something other than Name, the header label now shows, for example Display Name (Title). This ensures that the column name is clearly displayed. In addition, the field Name can now be displayed even when the display strategy is not Name.

Improvements to Text Editor

When using the Text Editor in the DataMigrator Console with a stored procedure (focexec) that has one more lines selected, new shortcut menu options allow for commenting or uncommenting multiples lines at once. The options are Comment Selection or Uncomment Selection.

Kerberos and Single Sign-On From DMC

A new Single Sign-On capability is now available when connecting from the Data Management Console to a DataMigrator or WebFOCUS Reporting Server on Linux with Kerberos enabled.

The IWA (Integrated Windows Authentication) that is available on the Server Node Configuration dialog box has been replaced with two new options: NTLM (NT Lan Manager) for Windows and Kerberos for use with Linux. When you select either of these options, you do not need to enter a user ID and password. The user ID and password you used to log in to Windows is used.

Once you connect to the DMC, a message appears in the console log indicating the authentication method and user ID that was used.
Using Find in Properties Panes

In the Data Management Console, you can right-click the Properties pane for a synonym or a flow and use the new Find option in the shortcut menu to search for text in the Attributes column.

Data Profiling

The following section provides descriptions of new features for data profiling.

Data Profiling Hex View Report

A new Data Profiling report called Hex View shows the hexadecimal value of a field. This report can be accessed in the Synonym Editor by right-clicking a field name, pointing to Data Profiling, and clicking Hex View.

Data Profiling Value Charts Option to Omit Missing Values

Data Profiling bar and pie charts have a new option that allows you to omit any missing (null) values. For data that contains any missing values, this results in a more readable and useful chart. When you enable this option, a label appears on the chart indicating that missing values were omitted.

In the DMC, you can select this option by clicking the Options button in the Home tab, and opening the Run Options from the navigation tree.

Quick Copy

The following section describes the new feature for the Quick Copy tool.

Quick Copy Allows Delimited Flat File Targets

From the Data Management Console or Web Console, the Quick Copy facility can now be used to create delimited flat files, in addition to loading database tables.

Reporting

The following section provides descriptions of new features for reporting.

Data Lineage Report

For a selected flow or procedure, a new Data Lineage report shows the name of each field used in the procedure and how it is used, its synonym name, and the name of the table and column in the underlying database.
Dependencies Analysis Report

The new Dependencies Analysis report lists all dependencies for a selected object. This report can be accessed from the shortcut menu of a flow, procedure, or synonym that allows the report to run.

Up to 256 Characters Allowed in Application Path

Starting with Release 7.7.07, the maximum length allowed for an application path has been increased from 64 to 256 characters. Users upgrading DataMigrator to Release 7.7.07 must recreate their Log and Statistics tables from the Data Management Console or the Web Console to enable this capability.

Synonym Editor

The following section provides descriptions of new features for the Synonym Editor.

Assigning SCD Columns on a Synonym

Previously, using the DataMigrator capability of loading a Slowly Changing Dimension (SCD) table required identifying the SCD type of each column in the target table. This was done in the Synonym Editor one column at a time.

A new dialog box provides a quicker, simpler way for assigning the SCD types. A new panel shows all the column names and folders for each SCD type. Columns can be assigned to the appropriate type by drag-and-drop.

Identifying Internal Fields

There are new icons in the Synonym Editor Field View that help to identify fields with types Internal and Needs Value. These field types do not allow sample data. The icons help to provide a visual indication that these fields are different.

In addition, the new Access Property column shows these values for fields with these properties. You can access this column by right-clicking the Name header in the Synonym Editor, pointing to Customize, and clicking Access Property.

New Parameterize Value Attribute

The new Parameterize Value dialog box allows you to overwrite the values for the CDC attributes with the existing, corresponding, CDC variables or with new variables.
Pivoting Columns and Rows

It is now simpler to edit a synonym to indicate that data should be pivoted, columns turned into rows or single column turned into multiple columns, for retrieval.

This can be done by either selecting a group of columns or a single character column that contains a delimited list of values or a JSON or XML document. A new dialog box allows specification for how the columns should be pivoted.

Also, data sources that support array structures in a field, such as Hive (for data managed by Hadoop) or MongoDB a synonym can now be created to represent an array so that the individual elements can be accessed.

Archiving Files on a Local Server

When the File Listener processes remotely located files, users can now archive these files on either a local or remote system by using the new ARCHIVE_LOCAL option that is available in the File Listener Attributes.
This section describes the new features for Resource Analyzer.

Resource Analyzer provides Information Systems (IS) organizations with the ability to manage the growing volume and unpredictable nature of ad hoc data access.

Resource Governor controls monitoring, system configuration parameters, and governing rules. It provides preemptive governing for requests issued to both relational and non-relational data sources.

Together, Resource Analyzer and its partner product, Resource Governor, are designed specifically to help IS organizations analyze and control end user data access.

**In this chapter:**

- New Top n Reports
- New Field in SMCOLUMNS Table
- Converting Existing Repositories to New Repositories
- Vertica Now Supported
- Special Characters Now Supported

**New Top n Reports**

Resource Management has been enhanced with new Top n reports. These reports show the usage data for the top number of users, procedures, or data sources specified for the report. The available options for top number are 5, 10, 15, 20, and 25. The default value is 10.

**New Field in SMCOLUMNS Table**

A new field, called SMDVPROP, has been added to the Resource Management Repository to indicate if a field in the request has a Dimension View property. The value of this new field can be M for Measure, D for Dimension, or blank if it is not known or not part of a Dimension View. SMDVPROP has a format of A1 and is an attribute of the SMCOLNMS.SMCOLUMN table.
Converting Existing Repositories to New Repositories

A Repository conversion process has been added to allow FOCUS repository to be converted to an SQL adapter repository. This feature also allows an existing SQL repository to be converted to a new SQL adapter connection or to the same SQL adapter connection using a new owner name. The new SQL adapter connection can be any of the SQL adapters that are certified for Resource Management repository usage.

Vertica Now Supported

Vertica has been added to the list of supported environments that can be selected for the Resource Management Repository.

Special Characters Now Supported

The Add to Monitor page for Relational Data Sources has been enhanced to support: *{}<>\/:!@#$%^+?~ and blank special characters in the data source name. Any data source with special characters will be quoted using the RDBMS quoted identifier character. The data source will be stored as a 2 part name, owner.table, and monitored using uppercase characters, except for any quoted special characters.
WebFOCUS is a complete information control system with comprehensive features for retrieving and analyzing data. It enables you to create reports quickly and easily. It also provides facilities for creating highly complex reports, but its strength lies in the simplicity of the request language. You can begin with simple queries and progress to complex reports as you learn about additional facilities.

Release 8.2.01 includes the Reporting Language new features available in Server Release 7707.

In this chapter:

- Additional Support for Alignment Grid
- Defining Internal Borders in an Alignment Grid
- Enhancements to PPTX Output Format
- Enhanced PDF Support for JPG Compression
- Defining Hyperlink Colors
- Automatically Resizing the Width of a Window or Frame With HFREEZE
- Enhanced Implementation of Peer Graphs
- SET FLOATMAPPING: Expanded Numeric Functionality
- New LOCALE Settings
- Applying Selection Criteria to the Internal Matrix Prior to COMPUTE Processing
- PARTITION_AGGR: Creating Rolling Calculations
- Limiting the Number of Stacked Commands and Loop Iterations
- New Report-Level Prefix Operators
- Controlling Processing of Delimited Files
- Format Display Option for Percentage Calculation
- Start of Week Parameter for the DTRUNC Function
- Function DTRUNC Returns the Last Day of a Calendar Period
- Simplified Numeric Functions
- Simplified Character Functions
- CHECKMD5: Computing an MD5 Hash Check Value
- CHECKSUM: Computing a Hash Sum
- Simplified Conversion Functions
- Simplified Date Functions
- Simplified System Functions
- Simplified Geography Functions
Additional Support for Alignment Grid

The HEADALIGN=BODY (Alignment Grid) feature has been expanded to support the following additional formats: PPTX and DHTML. The HEADALIGN=BODY feature enables you to insert a grid of cells in a heading to support more sophisticated styling.

For more information on the HEADALIGN=BODY feature, see the Creating Reports With WebFOCUS Language manual.

Defining Internal Borders in an Alignment Grid

The BORDERALL feature allows for the definition of internal borders within a HEADALIGN=BODY (Alignment Grid) element of a report.

The following example shows a DHTML request against the GGSALES data source. The request contains a subhead with HEADALIGN=BODY and BORDERALL=ON. The alignment grid in the subhead aligns with the data, and each item within the subhead is presented as fully bordered individual cells.
Enhancements to PPTX Output Format

The following are enhancements to the PPTX output format.

PNG Charts Embedded in Slides

The output image format for charts embedded in slides has been changed from JPG to PNG. PNG images are preferred as they produce better quality graph images, especially when displaying gradients, as well as provide support for transparency.
The PPTXGRAPHTYPE attribute enhances the quality of charts embedded into PowerPoint (PPTX) slides. As of Release 8.2.01M, you can use the PNG output format to enhance the image and text quality and support transparency.

This is useful for a number of important scenarios, including use of templates with background color and for overlapping a chart with other components and drawing objects.

The syntax is:

```
SET PPTXGRAPHTYPE={PNG|PNG_NOSCALE|JPEG}
```

where:

**PNG**

Scales the PNG image to twice its dimensions to get significantly improved quality. This may cause problems if you have non-scalable items in the chart, such as text with absolute point sizes (including embedded scales headings). The output file is also larger due to the larger bitmap. Text within the chart is noticeable sharper than the legacy JPEG format.

PNG preserves font sizes in the chart when it is internally rescaled for increased resolution. It converts absolute font sizes set in the stylesheet (*GRAPH_SCRIPT) to sizes expressed in virtual coordinates (which are relative to the dimensions of the chart) and generates font sizes for embedded headings and footings in virtual coordinates.

**PNG_NOSCALE**

Renders in PNG, but does not scale. This produces slightly better quality than JPEG. Going from JPEG to PNG_NOSCALE makes the chart sharper, but has only a slight effect on the text.

**JPEG**

Indicates legacy format. This is the default value.

For more information, see the *Creating Reports With WebFOCUS Language* manual.

**PPTX Template Masters and Slide Layouts for Expanded Styling**

A Microsoft PPTX template can contain one or more Slide Masters, defining a variety of different Slide Layouts. A Slide Master is the top slide in a hierarchy of slides that stores information about the theme and Slide Layouts of a presentation, including the background, color, fonts, effects, placeholder sizes, and positioning. You can incorporate two or more different styles or themes, such as backgrounds, color schemes, fonts, and effects, by inserting an individual Slide Master into the template for each different theme.
**Note:** Additional information on Microsoft PowerPoint Slide Layouts is available in an article titled What is a slide layout? on the Microsoft support site.

By default, the first Slide Layout in the first Slide Master is applied to slides on which WebFOCUS data is displayed.

With this new feature, WebFOCUS enables the developer to select any Slide Layout in any Slide Master in a PowerPoint template (POTX/POTM) or Presentation file (PPTX/PPTM). One Slide Layout may be applied to a slide or slides, displaying the output of a standard report, while one or different Slide Layouts may be applied to each Page Layout in a PPTX formatted Compound Document. The WebFOCUS generated output is placed on top of the styling on the selected Slide Layout.

For more information, see the Creating Reports With WebFOCUS Language manual.

**Support for Page Size and Orientation**

You can select the page size and orientation (Portrait or Landscape) for a PPTX formatted report, using the PAGESIZE and ORIENTATION style sheet attributes.

For more information, see the Creating Reports With WebFOCUS Language manual.

**ReportCaster Bursting Support**

Bursting PPTX formatted reports, including single reports and compound reports, is supported. Drill downs can be included in any of these burst reports.

For more information on ReportCaster bursting, see the ReportCaster manual.

**Enhanced PDF Support for JPG Compression**

The compression algorithm used for JPG images embedded in PDF documents has been enhanced to generate considerably smaller PDF files, and to allow users to control elements of the JPEG compression method. The following new SET commands allow users to select the compression methodology (SET JPEGENCODE) and the image quality (SET JPEGQUALITY).

**Syntax:** How to Select the Compression Algorithm

```
SET JPEGENCODE = [FLATE|DCT]
```

where:

```
FLATE
  Fixed quality defined
```
Is a compression method, introduced in WebFOCUS Release 7.7 Version 03, which allows images within PDF files to be individually compressed. This method is the default for WebFOCUS Release 7.7 Version 03 through WebFOCUS Release 7.7 Version 06.

**DCT**
Discrete Cosine Transform

Is a compression method, introduced in WebFOCUS Release 8.2 Version 01, which allows for the designation of the percentage of quality to retain. This method is the default in WebFOCUS Release 8.2 Version 01.

**Note:** In some images, quality loss is not noticeable when higher degrees of compression are applied. Generally, the higher the compression applied, the lower the image quality. Use SET JPEGQUALITY with the DCT setting to set the quality value.

**Syntax:** How to Select the Image Quality

```plaintext
SET JPEGQUALITY = n
```

where:

n
Is the percentage of quality used with the DCT setting. This value can be from 1-100. The default value is 100, where no quality is lost and minimal compression is applied.

**Defining Hyperlink Colors**

You can use the HYPERLINK-COLOR attribute to designate a color for a hyperlink within a report. This applies to all hyperlinks generated in the report. You can define a single color for the entire report or different colors for each individual element.

**Syntax:** How to Set Hyperlink Colors

```plaintext
TYPE = type, HYPERLINK-COLOR = color
```

where:

*type*
Is the report component you wish to affect. You can apply this keyword to the entire report using TYPE=REPORT. The attribute can also individually be set for any other element of the report.

*color*
Can use any style sheet supported color value designation.

For more information, see the *Creating Reports With WebFOCUS Language* manual.
Automatically Resizing the Width of a Window or Frame With HFREEZE

In an HTML HFREEZE report, you can use the SET AUTOFIT command to automatically resize HTML report output to 100% of the current container width of the window or frame.

Enhanced Implementation of Peer Graphs

The following are enhancements to peer graphs.

HEX and RGB Color Designations

For HTML output format, in addition to named colors, you can now use hex and RGB for color designations in a StyleSheet.

The syntax is:

\[ \text{GRAPHCOLOR} = \{ \text{color} | \text{RGB}((r \ g \ b | \text{#hexcolor})}) \]

For more information, see the Creating Reports With WebFOCUS Language manual.

Setting a Color for Data Visualization Bar Graphs

For all output formats, you can use the GRAPHCOLORNEG StyleSheet attribute to set a color for the data visualization bar graphs that represent negative values.

The syntax is:

\[ \text{GRAPHNEGCOLOR} = \{ \text{color} | \text{RGB}((r \ g \ b | \text{#hexcolor})}) \]

For more information, see the Creating Reports With WebFOCUS Language manual.

SET FLOATMAPPING: Expanded Numeric Functionality

SET FLOATMAPPING enables you to take advantage of decimal-based precision numbers available in DB2 and Oracle, and extends that functionality to all numeric processing for floating point numbers. With this processing, you gain both precision, including improved rounding, and enhanced performance.

The syntax is

\[ \text{SET FLOATMAPPING} = \{ \text{D|M|X} \} \]

where:

- **D**
  
  Uses the standard double-precision processing. This is the default value.
Uses a new internal format that provides decimal precision for double-precision floating point numbers up to 16 digits.

Uses a new internal format that provides decimal precision for double-precision floating point numbers up to 34 digits.

**Note:** If the field is passed to a HOLD file, the internal data types X or M will be propagated to the USAGE and ACTUAL formats in the HOLD Master File.

### New LOCALE Settings

New set parameters have been introduced to override locale-based attributes in a Master File. These locale parameters can be accessed in the Web Console from the Workspace page, using the LOCALE button on the ribbon.

When set from this page, these parameter values are stored in the server global profile, edasprof.prf. You can also set them in a TABLE request, a FOCEXEC, or in any supported profile. Some of these parameters also have been implemented at the field level as display format options.

Note that the LANGUAGE parameter, which used to be stored in nlscfg.err, will also be saved in edasprof.prf.

**Reference:** Aliases for CDN Parameter Values

Aliases have been added to the existing CDN parameter values to make their meanings more clear. The following list describes the values from prior releases and the aliases that can now be used.

- **OFF.** Uses a dot (.) as the decimal separator and comma (,) as the thousands separator. The alias for OFF is **COMMAS_DOT**. This is the default value.
- **ON.** Uses a comma as the decimal separator and dot as the thousands separator. The alias for ON is DOTS_COMMA.

- **SPACE.** Uses a comma as the decimal separator and space as the thousands separator. The alias for SPACE is SPACES_COMMA.

- **QUOTE.** Uses a comma as the decimal separator and single quotation mark (') as the thousands separator. The alias for QUOTE is QUOTES_COMMA.

- **QUOTEP.** Uses a dot as the decimal separator and single quotation mark as the thousands separator. The alias for QUOTEP is QUOTES_DOT.

**Reference:** New Currency Locale Parameters and Display Options

The following SET parameters have been added for specifying locale-based currency display, when the :C currency display option is used. All other currency display options are unaffected by these settings.

**CURRENCY_ISO_CODE**

This parameter defines the ISO code for the currency symbol to use.

The syntax is:

```plaintext
SET CURRENCY_ISO_CODE = iso
```

where:

```plaintext
iso
```

Is a standard three-character currency code such as USD for US dollars or JPY for Japanese yen. The default value is default, which uses the currency code for the configured language code.

**CURRENCY_DISPLAY**

This parameter defines the position of the currency symbol relative to the monetary number.

The syntax is:

```plaintext
SET CURRENCY_DISPLAY = pos
```
where:

**pos**

Defines the position of the currency symbol relative to a number. The default value is `default`, which uses the position for the format and currency symbol in effect. Valid values are:

- **LEFT_FIXED.** The currency symbol is left-justified preceding the number.
- **LEFT_FIXED_SPACE.** The currency symbol is left-justified preceding the number, with at least one space between the symbol and the number.
- **LEFT_FLOAT.** The currency symbol precedes the number, with no space between them.
- **LEFT_FLOAT_SPACE.** The currency symbol precedes the number, with one space between them.
- **TRAILING.** The currency symbol follows the number, with no space between them.
- **TRAILING_SPACE.** The currency symbol follows the number, with one space between them.

**CURRENCY_PRINT_ISO**

This parameter defines what will happen when the currency symbol cannot be displayed by the code page in effect.

The syntax is:

```
SET CURRENCY_PRINT_ISO = {DEFAULT|ALWAYS|NEVER}
```

where:

**DEFAULT**

Replaces the currency symbol with its ISO code, when the symbol cannot be displayed by the code page in effect. This is the default value.

**ALWAYS**

Always replaces the currency symbol with its ISO code.

**NEVER**

Never replaces the currency symbol with its ISO code. If the currency symbol cannot be displayed by the code page in effect, it will not be printed at all.

**Note:** Using a Unicode environment allows the printing of all currency symbols, otherwise this setting is needed.
Currency Display Options

The CURRENCY_ISO_CODE, CURRENCY_DISPLAY, and CURRENCY_PRINT_ISO parameters can be applied on the field level as display parameters in a Master File DEFINE, a DEFINE command, or in a COMPUTE using the :C display option. The syntax is:

```
fld/fmt:C(CURRENCY_DISPLAY='pos',
    CURRENCY_ISO_CODE='iso',CURRENCY_PRINT_ISO='prt')= expression;
```

where:

- **fld**
  - Is the field to which the parameters are to be applied.

- **fmt**
  - Is a numeric format that supports a currency value.

- **iso**
  - Is a standard three-character currency code, such as USD for US dollars or JPY for Japanese yen. The default value is default, which uses the currency code for the configured language code.

- **pos**
  - Defines the position of the currency symbol relative to a number. The default value is default, which uses the position for the format and currency symbol in effect. Valid values are:
    - LEFT_FIXED. The currency symbol is left-justified preceding the number.
    - LEFT_FIXED_SPACE. The currency symbol is left-justified preceding the number, with at least one space between the symbol and the number.
    - LEFT_FLOAT. The currency symbol precedes the number, with no space between them.
    - LEFT_FLOAT_SPACE. The currency symbol precedes the number, with one space between them.
    - TRAILING. The currency symbol follows the number, with no space between them.
    - TRAILING_SPACE. The currency symbol follows the number, with one space between them.

- **prt**
  - Can be one of the following print options.
    - DEFAULT
      - Replaces the currency symbol with its ISO code, when the symbol cannot be displayed by the code page in effect. This is the default value.
**ALWAYS**
Always replaces the currency symbol with its ISO code.

**NEVER**
Never replaces the currency symbol with its ISO code. If the currency symbol cannot be displayed by the code page in effect, it will not be printed at all.

*expression*
Is the expression that creates the virtual field.

**Note:** If currency parameters are specified at multiple levels, the order of precedence is:

1. Field level parameters.
2. Parameters set in a request (ON TABLE SET).
3. Parameters set in a FOCEXEC outside of a request.
4. Parameters set in a profile, using the precedence for profile processing.

**Example:**  **Specifying Currency Parameters in a DEFINE**

The following request creates a virtual field named Currency_parms that displays the currency symbol on the right using the ISO code for Japan, 'JPY'

```
DEFINE FILE WF_RETAIL_LITE
Currency_parms/D20.2:C(CURRENCY_DISPLAY='TRAILING',CURRENCY_ISO_CODE='JPY') = COGS_US;
END
TABLE FILE WF_RETAIL_LITE
SUM COGS_US Currency_parms
BY BUSINESS_REGION AS 'Region'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Region</th>
<th>Cost of Goods</th>
<th>Currency_parms</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>$1,247,925.00</td>
<td>$1,247,925.00¥</td>
</tr>
<tr>
<td>North America</td>
<td>$1,457,020.00</td>
<td>$1,457,020.00¥</td>
</tr>
<tr>
<td>Oceania</td>
<td>$9,613.00</td>
<td>$9,613.00¥</td>
</tr>
<tr>
<td>South America</td>
<td>$235,800.00</td>
<td>$235,800.00¥</td>
</tr>
</tbody>
</table>
Reference: New Date and Time Locale Parameters

The following SET parameters have been added to specify the order of date components, the date separator character, and the time separator character for the &TOD system variable.

**DATE_ORDER**

This parameter defines the order of date components for display. The syntax is:

```
SET DATE_ORDER = {DEFAULT|DMY|MDY|YMD}
```

where:

- **DEFAULT**
  
  Respects the original order of date components. This is the default value.

- **DMY**
  
  Displays all dates in day/month/year order.

- **MDY**
  
  Displays all dates in month/day/year order.

- **YMD**
  
  Displays all dates in year/month/day order.

**DATE_SEPARATOR**

This parameter defines the separator for date components for display.

The syntax is:

```
SET DATE_SEPARATOR = separator
```

where:

- **separator**
  
  Can be one of the following values.

  - **DEFAULT**, which respects the separator defined by the USAGE format of the field.
  - **SLASH**, which uses a slash (/) to separate date components.
  - **DASH**, which uses a dash (-) to separate date components.
  - **BLANK**, which uses a blank to separate date components.
  - **DOT**, which uses a dot (.) to separate date components.
  - **NONE**, which does not separate date components.
**TIME_SEPARATOR**

This parameter defines the separator for time components for the &TOD system variable.

The syntax is:

```
SET TIME_SEPARATOR = {DOT|COLON}
```

where:

- **DOT**
  
  Uses a dot (.) to separate time components. This is the default value.

- **COLON**
  
  Uses a colon (:) to separate time components.

**Reference:** Usage Notes for Locale-Based Date and Time Parameters

- **DATE_ORDER** and **DATE_SEPARATOR** override the specified date order for all date and date-time displays unless they include a translation display option (T, Tr, t, or tr), in which case the specified order is produced. To limit the scope to a request, use the ON TABLE SET phrase.

- To use these settings with the Dialogue Manager system variables, (for example, &DATE, &TOD, &YMD, &DATEfmt, and &DATXfmt) append the suffix .DATE_LOCALE to the system variable. This allows system variables that are localized to coexist with non-localized system variables.

**Example:** Setting Date and Time Parameters for System Variables

The following applies the **DATE_ORDER** and **DATE_SEPARATOR** parameters to the &DATE system variable.

```
SET DATE_SEPARATOR = DASH
SET DATE_ORDER = DMY
-TYPE NON-LOCALIZED: &DATE
-TYPE LOCALIZED: &DATE.DATE_LOCALE
```

The output is:

```
NON-LOCALIZED: 04/07/17
LOCALIZED: 07-04-17
```
Applying Selection Criteria to the Internal Matrix Prior to COMPUTE Processing

WHERE TOTAL tests are applied to the rows of the internal matrix after COMPUTE calculations are processed in the output phase of the report. WHERE_GROUPED tests are applied to the internal matrix values prior to COMPUTE calculations. The processing then continues with COMPUTE calculations, and then WHERE TOTAL tests. This allows the developer to control the evaluation, and is particularly useful in recursive calculations.

Syntax: How to Apply WHERE_GROUPED Selection Criteria

WHERE_GROUPED expression

where:

expression

Is an expression that does not refer to more than one row in the internal matrix. For example, it cannot use the LAST operator to refer to or retrieve a value from a prior record.

Example: Using a WHERE_GROUPED Test

The following request has two COMPUTE commands. The first COMPUTE checks to see if the business region value has changed, incrementing a counter if it has. This allows us to sequence the records in the matrix. The second COMPUTE creates a rolling total of the days delayed within the business region.

TABLE FILE WF_RETAIL_LITE
SUM DAYSDELAYED AS DAYS
COMPUTE CTR/I3 = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN CTR+1 ELSE 1;
COMPUTE NEWDAYS = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN NEWDAYS +DAYSDELAYED ELSE DAYSDELAYED;
BY BUSINESS_REGION AS Region
    BY TIME_MTH
WHERE BUSINESS_REGION NE 'Oceania'
ON TABLE SET PAGE NOPAGE
END
Applying Selection Criteria to the Internal Matrix Prior to COMPUTE Processing

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Region</th>
<th>Sale Month</th>
<th>DAYS</th>
<th>CTR</th>
<th>NEWDAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>1</td>
<td>191</td>
<td>1.00</td>
<td>191.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>205</td>
<td>2.00</td>
<td>396.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>224</td>
<td>3.00</td>
<td>620.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>213</td>
<td>4.00</td>
<td>833.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>185</td>
<td>5.00</td>
<td>1,018.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>234</td>
<td>6.00</td>
<td>1,252.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>223</td>
<td>7.00</td>
<td>1,475.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>234</td>
<td>8.00</td>
<td>1,709.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>249</td>
<td>9.00</td>
<td>1,958.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>290</td>
<td>10.00</td>
<td>2,248.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1</td>
<td>11.00</td>
<td>2,249.00</td>
</tr>
<tr>
<td>North America</td>
<td>1</td>
<td>253</td>
<td>1.00</td>
<td>253.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>205</td>
<td>2.00</td>
<td>458.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>293</td>
<td>3.00</td>
<td>751.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>319</td>
<td>4.00</td>
<td>1,070.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>273</td>
<td>5.00</td>
<td>1,343.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>268</td>
<td>6.00</td>
<td>1,611.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>219</td>
<td>7.00</td>
<td>1,830.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>248</td>
<td>8.00</td>
<td>2,078.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>220</td>
<td>9.00</td>
<td>2,298.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>326</td>
<td>10.00</td>
<td>2,624.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>8</td>
<td>11.00</td>
<td>2,632.00</td>
</tr>
<tr>
<td>South America</td>
<td>1</td>
<td>64</td>
<td>1.00</td>
<td>64.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>87</td>
<td>2.00</td>
<td>151.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>29</td>
<td>3.00</td>
<td>180.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>39</td>
<td>4.00</td>
<td>219.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>29</td>
<td>5.00</td>
<td>248.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>49</td>
<td>6.00</td>
<td>297.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>16</td>
<td>7.00</td>
<td>313.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>26</td>
<td>8.00</td>
<td>339.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>54</td>
<td>9.00</td>
<td>393.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>54</td>
<td>10.00</td>
<td>447.00</td>
</tr>
</tbody>
</table>
The following version of the request adds a WHERE TOTAL test to select only those months where DAYSDELAYED exceeded 200 days.

```
TABLE FILE WF_RETAIL_LITE
SUM DAYSDELAYED AS DAYS
COMPUTE CTR/I3 = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN CTR+1 ELSE 1;
COMPUTE NEWDAYS= IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN NEWDAYS +DAYSDELAYED ELSE DAYSDELAYED;
BY BUSINESS_REGION AS Region
BY TIME_MTH
WHERE BUSINESS_REGION NE 'Oceania'
WHERE TOTAL DAYSDELAYED GT 200
ON TABLE SET PAGE NOPAGE
END
```
The output is shown in the following image. The COMPUTE calculations for CTR and NEWDAYS was processed prior to eliminating the rows in which TOTAL DAYSDELAYED were 200 or less, so their values are the same as in the original output. This does not correctly reflect the sequence of records and the rolling total of the values that are actually displayed on the output. To do this, we need to select the appropriate months (DAYSDELAYED GT 200) before the COMPUTE expressions are evaluated. This requires WHERE_GROUPED.

<table>
<thead>
<tr>
<th>Region</th>
<th>Sale Month</th>
<th>DAYS</th>
<th>CTR</th>
<th>NEWDAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>2</td>
<td>205</td>
<td>2.00</td>
<td>396.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>224</td>
<td>3.00</td>
<td>620.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>213</td>
<td>4.00</td>
<td>833.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>234</td>
<td>6.00</td>
<td>1,252.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>223</td>
<td>7.00</td>
<td>1,475.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>234</td>
<td>8.00</td>
<td>1,709.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>249</td>
<td>9.00</td>
<td>1,958.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>290</td>
<td>10.00</td>
<td>2,248.00</td>
</tr>
<tr>
<td>North America</td>
<td>1</td>
<td>253</td>
<td>1.00</td>
<td>253.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>205</td>
<td>2.00</td>
<td>458.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>293</td>
<td>3.00</td>
<td>751.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>319</td>
<td>4.00</td>
<td>1,070.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>273</td>
<td>5.00</td>
<td>1,343.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>268</td>
<td>6.00</td>
<td>1,611.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>219</td>
<td>7.00</td>
<td>1,830.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>248</td>
<td>8.00</td>
<td>2,078.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>220</td>
<td>9.00</td>
<td>2,298.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>326</td>
<td>10.00</td>
<td>2,624.00</td>
</tr>
</tbody>
</table>
The following version of the request replaces the WHERE TOTAL test with a WHERE_GROUPED test.

```
TABLE FILE WF_RETAIL_LITE
SUM DAYSDELAYED AS DAYS
COMPUTE CTR/I3 = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN CTR+1 ELSE 1;
COMPUTE NEWDAYS= IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN NEWDAYS +DAYSDELAYED ELSE DAYSDELAYED;
BY BUSINESS_REGION AS Region
BY TIME_MTH
WHERE BUSINESS_REGION NE 'Oceania'
WHERE_GROUPED DAYSDELAYED GT 200
ON TABLE SET PAGE NOPAGE
END
```
The output is shown in the following image. The COMPUTE calculation for NEWDAYS was processed after eliminating the rows in which TOTAL DAYSDELAYED were 200 or less, so its values are based on fewer rows than the calculations in the original request. This is verified by the CTR values, which are now in a continuous sequence. The rolling total now reflects the values that are actually displayed on the report output.

<table>
<thead>
<tr>
<th>Region</th>
<th>Sale Month</th>
<th>DAYS</th>
<th>CTR</th>
<th>NEWDAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>2</td>
<td>205</td>
<td>1.00</td>
<td>205.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>224</td>
<td>2.00</td>
<td>429.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>213</td>
<td>3.00</td>
<td>642.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>234</td>
<td>4.00</td>
<td>876.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>223</td>
<td>5.00</td>
<td>1,099.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>234</td>
<td>6.00</td>
<td>1,333.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>249</td>
<td>7.00</td>
<td>1,582.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>290</td>
<td>8.00</td>
<td>1,872.00</td>
</tr>
<tr>
<td>North America</td>
<td>1</td>
<td>253</td>
<td>1.00</td>
<td>253.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>205</td>
<td>2.00</td>
<td>458.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>293</td>
<td>3.00</td>
<td>751.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>319</td>
<td>4.00</td>
<td>1,070.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>273</td>
<td>5.00</td>
<td>1,343.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>268</td>
<td>6.00</td>
<td>1,611.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>219</td>
<td>7.00</td>
<td>1,830.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>248</td>
<td>8.00</td>
<td>2,078.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>220</td>
<td>9.00</td>
<td>2,298.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>326</td>
<td>10.00</td>
<td>2,624.00</td>
</tr>
</tbody>
</table>
**Reference:** Usage Notes for WHERE_GROUPED

- If the expression refers to multiple rows in the internal matrix, the following message is generated and processing stops.

  (FOC32692) WHERE_GROUPED CANNOT REFER TO OTHER LINES OF REPORT

- A COMPUTE that does not reference multiple lines will be evaluated prior to WHERE_GROUPED tests, and may, therefore, be used in an expression and evaluated as part of a WHERE_GROUPED test.

- WHERE_GROUPED can be optimized for SQL data sources by creating a GROUP BY `fieldname` HAVING `expression` clause, where the expression is the WHERE_GROUPED selection criteria.

**PARTITION_AGGR: Creating Rolling Calculations**

Using the PARTITION_AGGR function, you can generate rolling calculations based on a block of rows from the internal matrix of a TABLE request. In order to specify how to determine the limits of the rolling calculations, you specify a partition of the data based on either a sort field or the entire TABLE. Within either type of break, you can start calculating from the beginning of the break or a number of rows prior to the current row. You can stop the rolling calculation at the current row or the end of the partition.

By default, the field values used in the calculations are the summed values of a measure in the request. Certain prefix operators can be used to add a column to the internal matrix and use that column in the rolling calculations. The rolling calculation can be SUM, AVE, CNT, MIN, MAX, FST, or LST.

**Syntax:** How to Generate Rolling Calculations Using PARTITION_AGGR

```
PARTITION_AGGR([prefix.]measure,{sortfield|TABLE},from,to,operation)
```

where:

`prefix`.

Defines an aggregation operator to apply to the measure before using it in the rolling calculation. Valid operators are:

- **SUM.** which calculates the sum of the measure field values. SUM is the default operator.

- **CNT.** which calculates a count of the measure field values.
AVE. which calculates the average of the measure field values.

MIN. which calculates the minimum of the measure field values.

MAX. which calculates the maximum of the measure field values.

FST. which retrieves the first value of the measure field.

LST. which retrieves the last value of the measure field.

Note: The operators PCT., RPCT., TOT., MDN., and DST. are not supported. COMPUTEs that reference those unsupported operators are also not supported.

measure
Is the measure field to be aggregated. It can be a real field in the request or a calculated value generated with the COMPUTE command, as long as the COMPUTE does not reference an unsupported prefix operator.

sortfield
Is a BY or ACROSS field that defines the boundary of the partition. Operations will not cross a boundary. In the request the BY HIGHEST phrase to sort high-to-low is supported. ACROSS COLUMNS AND is also supported, but BY ROWS OVER and FOR are not supported.

Specifying TABLE as the boundary makes the partition boundary the entire internal matrix.

For example, if the sort is BY YEAR BY MONTH, with data from both 2014 and 2015, specifying the boundary as YEAR means that January 2015 - 2 will be valued as zero (0) or MISSING, as two months prior to January 2015 would cross the YEAR boundary. However, specifying TABLE as the boundary and requesting - 2 months would return the data for November 2014.

from
Identifies the starting point for the rolling calculation. Valid values are:

- -n, which starts the calculation n rows back from the current row.

- B, which starts the calculation at the beginning of the current sort break (the first line with the same sort field value as the current line).

to
Identifies the ending point of the rolling calculation. Valid values are:

- C, which ends the rolling calculation at the current row in the internal matrix.

- E, which ends the rolling calculation at the end of the sort break (the last line with the same sort value as the current row.)
operation

Specifies the rolling calculation used on the values in the internal matrix. Supported operations are:

- **SUM.** which calculates a rolling sum.
- **AVE.** which calculates a rolling average.
- **CNT.** which counts the rows in the partition.
- **MIN.** which returns the minimum value in the partition.
- **MAX.** which returns the maximum value in the partition.
- **FST.** which returns the first value in the partition.
- **LST.** which returns the last value in the partition.

The calculation is performed prior to any WHERE TOTAL tests, but after WHERE_GROUPED tests.

**Example: Calculating a Rolling Average**

The following request calculates a rolling average of the current line and the previous line in the internal matrix within the quarter.

```
TABLE FILE WF_RETAIL_LITE
SUM COGS_US
COMPUTE AVE1/D12.2M = PARTITION_AGGR(COGS_US, TIME_QTR, -1, C, AVE);
BY BUSINESS_REGION
BY TIME_QTR
BY TIME_MTH
WHERE BUSINESS_REGION EQ 'North America' OR 'South America'
ON TABLE SET PAGE NOLEAD
END
```
The output is shown in the following image. Within each quarter, the first average is just the value from Q1, as going back 1 would cross a boundary. The second average is calculated using the first two rows within that quarter, and the third average is calculated using rows 2 and 3 within the quarter.

<table>
<thead>
<tr>
<th>Customer Business Region</th>
<th>Sale Quarter</th>
<th>Sale Month</th>
<th>Cost of Goods</th>
<th>AVE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>1</td>
<td>1</td>
<td>$131,745.00</td>
<td>$131,745.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$138,150.00</td>
<td>$134,947.50</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$166,097.00</td>
<td>$152,123.50</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$147,770.00</td>
<td>$147,770.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$143,461.00</td>
<td>$145,615.50</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$144,101.00</td>
<td>$143,781.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>$140,641.00</td>
<td>$140,641.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$138,756.00</td>
<td>$139,698.50</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$131,858.00</td>
<td>$135,307.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>$172,837.00</td>
<td>$172,837.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
<td>$1,604.00</td>
<td>$87,220.50</td>
</tr>
<tr>
<td>South America</td>
<td>1</td>
<td>1</td>
<td>$33,895.00</td>
<td>$33,895.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$35,753.00</td>
<td>$34,824.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$13,753.00</td>
<td>$24,753.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$24,339.00</td>
<td>$24,339.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$29,076.00</td>
<td>$26,707.50</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$22,726.00</td>
<td>$25,901.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>$16,475.00</td>
<td>$16,475.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$12,504.00</td>
<td>$14,489.50</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$19,753.00</td>
<td>$16,128.50</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>$27,526.00</td>
<td>$27,526.00</td>
</tr>
</tbody>
</table>
The following changes the rolling average to start from the beginning of the sort break.

\[
\text{COMPUTE AVE1/D12.2M = PARTITION_AGGR(COGS_US, TIME_QTR , B, C, AVE);}\]

The output is shown in the following image. Within each quarter, the first average is just the value from Q1, as going back would cross a boundary. The second average is calculated using the first two rows within that quarter, and the third average is calculated using rows 1 through 3 within the quarter.

<table>
<thead>
<tr>
<th>Customer Business Region</th>
<th>Sale Quarter</th>
<th>Sale Month</th>
<th>Cost of Goods</th>
<th>AVE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>1</td>
<td>1</td>
<td>$131,745.00</td>
<td>$131,745.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$138,150.00</td>
<td>$134,947.50</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$166,097.00</td>
<td>$145,330.67</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>$147,770.00</td>
<td>$147,770.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$143,461.00</td>
<td>$145,615.50</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$144,101.00</td>
<td>$145,110.67</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>$140,641.00</td>
<td>$140,641.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$138,756.00</td>
<td>$139,698.50</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$131,858.00</td>
<td>$137,085.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>$172,837.00</td>
<td>$172,837.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>11</td>
<td>$1,604.00</td>
<td>$87,220.50</td>
</tr>
<tr>
<td>South America</td>
<td>1</td>
<td>1</td>
<td>$33,895.00</td>
<td>$33,895.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$35,753.00</td>
<td>$34,824.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$13,753.00</td>
<td>$27,800.33</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>$24,339.00</td>
<td>$24,339.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$29,076.00</td>
<td>$26,707.50</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$22,726.00</td>
<td>$25,380.33</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>$16,475.00</td>
<td>$16,475.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$12,504.00</td>
<td>$14,489.50</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$19,753.00</td>
<td>$16,244.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>$27,526.00</td>
<td>$27,526.00</td>
</tr>
</tbody>
</table>
The following command uses the partition boundary TABLE.

\[
\text{COMPUTE AVE1/D12.2M = PARTITION_AGGR(COGS_US, TABLE, B, C, AVE);} 
\]

The output is shown in the following image. The rolling average keeps adding the next row to the average until a break in the business region sort field.

<table>
<thead>
<tr>
<th>Customer Business Region</th>
<th>Sale Quarter</th>
<th>Sale Month</th>
<th>Cost of Goods</th>
<th>AVE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>1</td>
<td>1</td>
<td>$131,745.00</td>
<td>$131,745.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$138,150.00</td>
<td>$134,947.50</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$166,097.00</td>
<td>$145,330.67</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$147,770.00</td>
<td>$145,940.50</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$143,461.00</td>
<td>$145,444.60</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$144,101.00</td>
<td>$145,220.67</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>$140,641.00</td>
<td>$144,566.43</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$138,756.00</td>
<td>$143,840.13</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$131,858.00</td>
<td>$142,508.78</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>$172,837.00</td>
<td>$145,541.60</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>11</td>
<td>$1,604.00</td>
<td>$132,456.36</td>
</tr>
<tr>
<td>South America</td>
<td>1</td>
<td>1</td>
<td>$33,895.00</td>
<td>$124,242.92</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$35,753.00</td>
<td>$117,436.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$13,753.00</td>
<td>$110,030.07</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$24,339.00</td>
<td>$104,317.33</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$29,076.00</td>
<td>$99,614.75</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$22,726.00</td>
<td>$95,091.88</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>$16,475.00</td>
<td>$90,724.28</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$12,504.00</td>
<td>$86,607.42</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$19,753.00</td>
<td>$83,264.70</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>$27,526.00</td>
<td>$80,610.48</td>
</tr>
</tbody>
</table>
Reference: Usage Notes for PARTITION_AGGR

- Fields referenced in the PARTITION_AGGR parameters but not previously mentioned in the request, will not be counted in column notation or propagated to HOLD files.

- Using the WITHIN phrase for a sum is the same as computing PARTITION_AGGR on the WITHIN sort field from B (beginning of sort break) to E (end of sort break) using SUM, as in the following example.

```sql
TABLE FILE WF_RETAIL_LITE
SUM COGS_US WITHIN TIME_QTR AS 'WITHIN Qtr'
COMPUTE PART_WITHIN_QTR/D12.2M = PARTITION_AGGR(COGS_US, TIME_QTR, B, E, SUM);
BY BUSINESS_REGION AS Region
BY TIME_QTR
BY TIME_MTH
WHERE BUSINESS_REGION EQ 'North America' OR 'South America'
ON TABLE SET PAGE NOPAGE
END
```
The output is shown in the following image.

<table>
<thead>
<tr>
<th>Region</th>
<th>Sale Quarter</th>
<th>Sale Month</th>
<th>WITHIN Qtr</th>
<th>PART_WITHIN_QTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>1</td>
<td>1</td>
<td>$435,992.00</td>
<td>$435,992.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$435,992.00</td>
<td>$435,992.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$435,992.00</td>
<td>$435,992.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$435,332.00</td>
<td>$435,332.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$435,332.00</td>
<td>$435,332.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$435,332.00</td>
<td>$435,332.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>$411,255.00</td>
<td>$411,255.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$411,255.00</td>
<td>$411,255.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$411,255.00</td>
<td>$411,255.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>$174,441.00</td>
<td>$174,441.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>11</td>
<td>$174,441.00</td>
<td>$174,441.00</td>
</tr>
<tr>
<td>South America</td>
<td>1</td>
<td>1</td>
<td>$83,401.00</td>
<td>$83,401.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>$83,401.00</td>
<td>$83,401.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>$83,401.00</td>
<td>$83,401.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$76,141.00</td>
<td>$76,141.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>$76,141.00</td>
<td>$76,141.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>$76,141.00</td>
<td>$76,141.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>$48,732.00</td>
<td>$48,732.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>$48,732.00</td>
<td>$48,732.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>$48,732.00</td>
<td>$48,732.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>$27,526.00</td>
<td>$27,526.00</td>
</tr>
</tbody>
</table>
With other types of calculations, the results are not the same. For example, the following request calculates the average within quarter using the WITHIN phrase and the average with quarter using PARTITION_AGG.

```sql
TABLE FILE WF_RETAIL_LITE
SUM COGS_US AS Cost
CNT.COGS_US AS Count AVE.COGS_US WITHIN TIME_QTR AS 'Ave Within'
COMPUTE PART_WITHIN_QTR/D12.2M = PARTITION_AGGR(COGS_US, TIME_QTR, B, E, AVE);
BY BUSINESS_REGION AS Region
BY TIME_QTR
ON TIME_QTR SUBTOTAL COGS_US CNT.COGS_US
BY TIME_MTH
WHERE BUSINESS_REGION EQ 'North America'
ON TABLE SET PAGE NOPAGE
END
```
The output is shown in the following image. The average using the WITHIN phrase divides the total cost for the quarter by the total count of instances for the quarter (for example, $435,992.00/1514 = $287.97), while PARTITION_AGGR divides the total cost for the quarter by the number of report rows in the quarter (for example, $435,992.00/3 = $145,330.67).

<table>
<thead>
<tr>
<th>Region</th>
<th>Sale Quarter</th>
<th>Sale Month</th>
<th>Cost</th>
<th>Count</th>
<th>Ave Within</th>
<th>PART_WITHIN_QTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>1</td>
<td>1</td>
<td>$131,745.00</td>
<td>483</td>
<td>$287.97</td>
<td>$145,330.67</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>$147,770.00</td>
<td>497</td>
<td>$298.79</td>
<td>$145,110.67</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>7</td>
<td>$140,641.00</td>
<td>451</td>
<td>$299.97</td>
<td>$137,085.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$138,756.00</td>
<td>451</td>
<td>$299.97</td>
<td>$137,085.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>10</td>
<td>$172,837.00</td>
<td>569</td>
<td>$302.32</td>
<td>$87,220.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1,604.00</td>
<td>8</td>
<td>$302.32</td>
<td>$87,220.50</td>
</tr>
</tbody>
</table>

*TOTAL TIME_QTR 1 $435,992.00 1514

*TOTAL TIME_QTR 2 $435,332.00 1457

*TOTAL TIME_QTR 3 $411,255.00 1371

*TOTAL TIME_QTR 4 $174,441.00 577

TOTAL $1,457,020.00 4919

If you use PARTITION_AGGR to perform operations for specific time periods using an offset, for example, an operation on the quarters for different years, you must make sure that every quarter is represented. If some quarters are missing for some years, the offset will not access the correct data. In this case, generate a HOLD file that has every quarter represented for every year (you can use BY QUARTER ROWS OVER 1 OVER 2 OVER 3 OVER 4) and use PARTITION_AGGR on the HOLD file.
PARTITION_REF: Using Prior Field Values in Calculations

Use of LAST in a calculation retrieves the LAST value of the specified field the last time this calculation was performed. The PARTITION_REF function enables you to specify both how many rows back to go in order to retrieve a prior value, and a sort break within which the prior value calculation will be contained.

**Syntax:**

How to Retrieve Prior Field Values for Use in a Calculation

\[
\text{PARTITION\_REF}([\text{prefix.}]\text{field}, \{\text{srtfield|TABLE}\}, -\text{offset})
\]

where:

- **prefix**
  Is optional. If used, it can be one of the following aggregation operators:

  - AVE. Average
  - MAX. Maximum
  - MIN. Minimum
  - CNT. Count
  - SUM. Sum

- **field**
  Is the field whose prior value is to be retrieved.

- **srtfield|TABLE**
  Is the sort break within which to go back to retrieve the value. TABLE means retrieve the value without regard to sort breaks. Operations will not cross a partition boundary.

  The Sort field may use BY HIGHEST to indicate a HIGH-TO-LOW sort. ACROSS COLUMNS AND is supported. BY ROWS OVER and FOR are not supported.

- **offset**
  Is the integer number of records back to go to retrieve the value.

If the offset is prior to the partition boundary sort value, the return will be the default value for the field. The calculation is performed prior to any WHERE TOTAL tests, but after WHERE_GROUPED tests.
**Example:**  Retrieving a Previous Record With PARTITION_REF

The following request retrieves the previous record within the sort field PRODUCT_CATEGORY.

```plaintext
TABLE FILE WF_RETAIL_LITE
SUM DAYSDELAYED
COMPUTE NEWDAYS/I5=PARTITION_REF(DAYSDELAYED, PRODUCT_CATEGORY, -1);
BY PRODUCT_CATEGORY
BY PRODUCT_SUBCATEG
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image. The first value within each sort break is zero because there is no prior record to retrieve.

<table>
<thead>
<tr>
<th>Product Category</th>
<th>Product Subcategory</th>
<th>Days Delayed</th>
<th>NEWDAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>Charger</td>
<td>164</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Headphones</td>
<td>335</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Universal Remote Controls</td>
<td>264</td>
<td>335</td>
</tr>
<tr>
<td>Camcorder</td>
<td>Handheld</td>
<td>430</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Professional</td>
<td>21</td>
<td>430</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>324</td>
<td>21</td>
</tr>
<tr>
<td>Computers</td>
<td>Smartphone</td>
<td>342</td>
<td>0</td>
</tr>
<tr>
<td>Media Player</td>
<td>Blu Ray</td>
<td>1,102</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DVD Players</td>
<td>31</td>
<td>1102</td>
</tr>
<tr>
<td></td>
<td>Streaming</td>
<td>85</td>
<td>31</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>Home Theater Systems</td>
<td>652</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Receivers</td>
<td>220</td>
<td>652</td>
</tr>
<tr>
<td></td>
<td>Speaker Kits</td>
<td>404</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>iPod Docking Station</td>
<td>520</td>
<td>404</td>
</tr>
<tr>
<td>Televisions</td>
<td>Flat Panel TV</td>
<td>144</td>
<td>0</td>
</tr>
<tr>
<td>Video Production</td>
<td>Video Editing</td>
<td>317</td>
<td>0</td>
</tr>
</tbody>
</table>
The following request retrieves the average cost of goods from two records prior to the current record within the PRODUCTCATEGORY sort field.

```
TABLE FILE WF_RETAIL_LITE
SUM COGS_US AVE.COGS_US AS Average
COMPUTE PartitionAve/D12.2M=PARTITION_REF(AVE.COGS_US, PRODUCTCATEGORY, -2);
BY PRODUCTCATEGORY
BY PRODUCTSUBCATEGORY
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Product Category</th>
<th>Product Subcategory</th>
<th>Cost of Goods</th>
<th>Average</th>
<th>PartitionAve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>Charger</td>
<td>$8,573.00</td>
<td>$26.96</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Headphones</td>
<td>$199,325.00</td>
<td>$292.69</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Universal Remote Controls</td>
<td>$134,979.00</td>
<td>$278.31</td>
<td>$26.96</td>
</tr>
<tr>
<td>Camcorder</td>
<td>Handheld</td>
<td>$90,607.00</td>
<td>$114.11</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Professional</td>
<td>$159,048.00</td>
<td>$3,614.73</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>$203,550.00</td>
<td>$351.55</td>
<td>$114.11</td>
</tr>
<tr>
<td>Computers</td>
<td>Smartphone</td>
<td>$109,281.00</td>
<td>$182.13</td>
<td>$0.00</td>
</tr>
<tr>
<td>Media Player</td>
<td>Blu Ray</td>
<td>$754,249.00</td>
<td>$374.32</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>DVD Players</td>
<td>$15,096.00</td>
<td>$215.66</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Streaming</td>
<td>$10,248.00</td>
<td>$62.49</td>
<td>$374.32</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>Home Theater Systems</td>
<td>$245,581.00</td>
<td>$199.98</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Receivers</td>
<td>$159,790.00</td>
<td>$369.88</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Speaker Kits</td>
<td>$343,113.00</td>
<td>$458.09</td>
<td>$199.98</td>
</tr>
<tr>
<td></td>
<td>iPod Docking Station</td>
<td>$108,558.00</td>
<td>$116.85</td>
<td>$369.88</td>
</tr>
<tr>
<td>Televisions</td>
<td>Flat Panel TV</td>
<td>$227,820.00</td>
<td>$853.26</td>
<td>$0.00</td>
</tr>
<tr>
<td>Video Production</td>
<td>Video Editing</td>
<td>$180,540.00</td>
<td>$279.91</td>
<td>$0.00</td>
</tr>
</tbody>
</table>
Replacing the function call with the following syntax changes the partition boundary to TABLE.

```plaintext
COMPUTE PartitionAve/D12.2M=PARTITION_REF(AVE.COGS_US, TABLE, -2);
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Product Category</th>
<th>Product Subcategory</th>
<th>Cost of Goods</th>
<th>Average</th>
<th>PartitionAve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>Charger</td>
<td>$8,573.00</td>
<td>$26.96</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Headphones</td>
<td>$199,325.00</td>
<td>$292.69</td>
<td>$0.00</td>
</tr>
<tr>
<td></td>
<td>Universal Remote Controls</td>
<td>$134,979.00</td>
<td>$278.31</td>
<td>$26.96</td>
</tr>
<tr>
<td>Camcorder</td>
<td>Handheld</td>
<td>$90,607.00</td>
<td>$114.11</td>
<td>$292.69</td>
</tr>
<tr>
<td></td>
<td>Professional</td>
<td>$159,048.00</td>
<td>$3,614.73</td>
<td>$278.31</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>$203,550.00</td>
<td>$351.55</td>
<td>$114.11</td>
</tr>
<tr>
<td>Computers</td>
<td>Smartphone</td>
<td>$109,281.00</td>
<td>$182.13</td>
<td>$3,614.73</td>
</tr>
<tr>
<td>Media Player</td>
<td>Blu Ray</td>
<td>$754,249.00</td>
<td>$374.32</td>
<td>$351.55</td>
</tr>
<tr>
<td></td>
<td>DVD Players</td>
<td>$15,096.00</td>
<td>$215.66</td>
<td>$182.13</td>
</tr>
<tr>
<td></td>
<td>Streaming</td>
<td>$10,248.00</td>
<td>$62.49</td>
<td>$374.32</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>Home Theater Systems</td>
<td>$245,581.00</td>
<td>$199.98</td>
<td>$215.66</td>
</tr>
<tr>
<td></td>
<td>Receivers</td>
<td>$159,790.00</td>
<td>$369.88</td>
<td>$62.49</td>
</tr>
<tr>
<td></td>
<td>Speaker Kits</td>
<td>$343,113.00</td>
<td>$458.09</td>
<td>$199.98</td>
</tr>
<tr>
<td></td>
<td>iPod Docking Station</td>
<td>$108,558.00</td>
<td>$116.85</td>
<td>$369.88</td>
</tr>
<tr>
<td>Televisions</td>
<td>Flat Panel TV</td>
<td>$227,820.00</td>
<td>$853.26</td>
<td>$458.09</td>
</tr>
<tr>
<td>Video Production</td>
<td>Video Editing</td>
<td>$180,540.00</td>
<td>$279.91</td>
<td>$116.85</td>
</tr>
</tbody>
</table>

**Reference:** Usage Notes for PARTITION_REF

- Fields referenced in the PARTITION_REF parameters but not previously mentioned in the request, will not be counted in column notation or propagated to HOLD files.
Increased Number of ACROSS Columns in a Report

The limit on the number of ACROSS columns has been removed. The total number of sort fields (BY and ACROSS) still has a maximum of 128.

Increased Number of Verb Objects in a TABLE Request

The number of verb objects supported in a TABLE request is unlimited. However, errors can occur if the output is being sent to a format that has a limit on the number of columns, if the system has a maximum record length that cannot contain the number of columns of output, or if the amount of memory needed to store the output is unavailable.

Specifying a Geographic Role in a DEFINE or COMPUTE

If you define a virtual field or create a calculated value that is location-related, you can specify a geographic role. This may be helpful when using the field in a location-based chart request.

Syntax: How to Specify a Geographic Role in a DEFINE or COMPUTE

```
DEFINE name/fmt (GEOGRAPHIC_ROLE = georole)
   [MISSING ON NEEDS {SOME|ALL} DATA] = expression;
```

```
COMPUTE name/fmt (GEOGRAPHIC_ROLE = georole)
   [MISSING ON NEEDS {SOME|ALL} DATA] = expression;
```

where:

```
name
   Is a name for the virtual field or calculated value.
```

```
fmt
   Is a valid format specification for the geographic value.
```

```
georole
   Is a valid geographic role. Geographic roles can be names, postal codes, ISO (International Organization for Standardization) codes, FIPS (Federal Information Processing Standards) codes, or NUTS (Nomenclature of Territorial Units for Statistics) codes. The following is a list of supported geographic roles.
```

- ADDRESS_FULL. Full address.
- ADDRESS_LINE. Number and street name.
- CITY. City name.
- CONTINENT. Continent name.
 Specifying a Geographic Role in a DEFINE or COMPUTE

- **CONTINENT_ISO2.** Continent ISO-3166 code.
- **COUNTRY.** Country name.
- **COUNTRY_FIPS.** Country FIPS code.
- **COUNTRY_ISO2.** Country ISO-3166-2 code
- **COUNTRY_ISO3.** Country ISO-3166-3 code
- **GEOMETRY_AREA.** Geometry area.
- **GEOMETRY_LINE.** Geometry line.
- **GEOMETRY_POINT.** Geometry point.
- **LATITUDE.** Latitude.
- **LONGITUDE.** Longitude.
- **NUTS0.** Country name (NUTS level 0).
- **NUTS0_CC.** Country code (NUTS level 0).
- **NUTS1.** Region name (NUTS level 1).
- **NUTS1_CC.** Region code (NUTS level 1).
- **NUTS2.** Province name (NUTS level 2).
- **NUTS2_CC.** Province code (NUTS level 2).
- **NUTS3.** District name (NUTS level 3).
- **NUTS3_CC.** District code (NUTS level 3).
- **POSTAL_CODE.** Postal code.
- **STATE.** State name.
- **STATE_FIPS.** State FIPS code.
- **STATE_ISO_SUB.** US State ISO subdivision code.
- **USSCITY.** US city name.
- **USCITY_FIPS.** US city FIPS code.
- **USCOUNTY.** US county name.
Example:  Specifying a Geographic Role in a DEFINE

The following defines a field whose geographic role is the state name.

```
DEFINE FILE WF_RETAIL_LITE
STATENAME/A20 (GEOGRAPHIC_ROLE = STATE) = STATE_PROV_NAME;
END
```

Limiting the Number of Stacked Commands and Loop Iterations

As WebFOCUS commands are encountered in a procedure, they are placed on a stack in memory (called FOCSTACK). The procedure is executed and the stack counter reset to zero (0) as a result a -RUN command, certain system variables, or the end of the procedure.

The parameter DMH_STACKLIM sets the maximum number of lines allowed in the FOCSTACK. This parameter is designed to prevent runaway tasks that generate a FOCSTACK that uses an inordinate amount of memory.

If users are not careful in designing loops in Dialogue Manager, they can inadvertently cause an infinite looping condition.

The parameter DMH_LOOPLIM sets the maximum number of Dialogue Manager loop iterations allowed, using -REPEAT or -GOTO commands.

DMH_STACKLIM and DMH_LOOPLIM should be set high enough to run your existing reports and procedures without error for your entire session. It is recommended that if you set these parameters, you should set them in a profile.

Syntax:  How to Limit the Number of Stacked Commands

```
SET DMH_STACKLIM = n
```
where:

\[ n \]

Sets the maximum number of lines allowed in FOCSTACK. The default value is zero (0), which does not limit the number of stacked commands.

**Syntax:** How to Limit the Number of Loop Iterations

```plaintext
SET DMH_LOOPLIM = n
```

where:

\[ n \]

Sets the maximum number of loop iterations allowed. The default value is zero (0), which does not limit the number of loop iterations.

**New Report-Level Prefix Operators**

Prefix operators have been added for headings, footings, subheadings, subfootings, verb objects, and calculated values (COMPUTEs) that calculate the average, maximum, minimum, and count for the entire report. They are based on the TOT. operator, which calculates total values to include in a heading.

These operators cannot be referenced in WHERE or WHERE TOTAL tests. However, they can be used in a COMPUTE command to generate a calculated value that can be used in a WHERE TOTAL test.

**Syntax:** How to Calculate Report-Level Average, Maximum, Minimum, and Count Values

```plaintext
operator.field
```

where:

- **operator**
  - Can be one of the following prefix operators.
    - **TOTAVE.** Calculates the average value of the field for the entire table.
    - **TOTMAX.** Calculates the maximum value of the field for the entire table.
    - **TOTMIN.** Calculates the minimum value of the field for the entire table.
    - **TOTCNT.** Calculates the count of the field instances for the entire table.

- **field**
  - Is a verb object or calculated value in the request.
Example: Using Prefix Operators in a Heading

The following request uses prefix operators in the heading.

```
TABLE FILE WF_RETAIL_LITE
HEADING
"Heading Calculations:"
"Total:" <TOT.COGS_US
"Count:" <TOTCNT.COGS_US
"Average:" <TOTAVE.COGS_US
"Minimum:" <TOTMIN.COGS_US
"Maximum:" <TOTMAX.COGS_US
SUM COGS_US CNT.COGS_US AS Count AVE.COGS_US AS Average
MIN.COGS_US AS Minimum MAX.COGS_US AS Maximum
BY BUSINESS_REGION AS Region
BY PRODUCT_CATEGORY AS Category
WHERE BUSINESS_REGION NE 'Oceania'
ON TABLE SUBTOTAL COGS_US CNT.COGS_US AS Total
ON TABLE SET PAGE NOPAGE
ON TABLE SET SHOWBLANKS ON
ON TABLE SET STYLE *
type=report,grid=off, size=11,$
ENDSTYLE
END
```
### Heading Calculations:
- **Total:** $2,940,745.00
- **Count:** 9,961
- **Minimum:** $295.23
- **Maximum:** $9,750.00

<table>
<thead>
<tr>
<th>Region</th>
<th>Category</th>
<th>Cost of Goods</th>
<th>Count</th>
<th>Average</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>Accessories</td>
<td>$143,587.00</td>
<td>630</td>
<td>$228.55</td>
<td>$16.00</td>
<td>$2,850.00</td>
</tr>
<tr>
<td></td>
<td>Camcorder</td>
<td>$187,000.00</td>
<td>611</td>
<td>$306.06</td>
<td>$60.00</td>
<td>$8,610.00</td>
</tr>
<tr>
<td></td>
<td>Computers</td>
<td>$47,616.00</td>
<td>266</td>
<td>$179.01</td>
<td>$81.00</td>
<td>$668.00</td>
</tr>
<tr>
<td></td>
<td>Media Player</td>
<td>$328,401.00</td>
<td>956</td>
<td>$343.52</td>
<td>$36.00</td>
<td>$1,640.00</td>
</tr>
<tr>
<td></td>
<td>Stereo Systems</td>
<td>$361,756.00</td>
<td>1412</td>
<td>$256.20</td>
<td>$48.00</td>
<td>$2,295.00</td>
</tr>
<tr>
<td></td>
<td>Televisions</td>
<td>$100,443.00</td>
<td>120</td>
<td>$837.02</td>
<td>$275.00</td>
<td>$6,500.00</td>
</tr>
<tr>
<td></td>
<td>Video Production</td>
<td>$78,722.00</td>
<td>276</td>
<td>$285.22</td>
<td>$78.00</td>
<td>$1,850.00</td>
</tr>
<tr>
<td>North America</td>
<td>Accessories</td>
<td>$171,306.00</td>
<td>743</td>
<td>$230.56</td>
<td>$16.00</td>
<td>$2,040.00</td>
</tr>
<tr>
<td></td>
<td>Camcorder</td>
<td>$242,967.00</td>
<td>692</td>
<td>$351.11</td>
<td>$60.00</td>
<td>$6,000.00</td>
</tr>
<tr>
<td></td>
<td>Computers</td>
<td>$53,329.00</td>
<td>284</td>
<td>$187.78</td>
<td>$81.00</td>
<td>$668.00</td>
</tr>
<tr>
<td></td>
<td>Media Player</td>
<td>$386,681.00</td>
<td>1118</td>
<td>$345.87</td>
<td>$36.00</td>
<td>$1,600.00</td>
</tr>
<tr>
<td></td>
<td>Stereo Systems</td>
<td>$412,036.00</td>
<td>1636</td>
<td>$251.86</td>
<td>$48.00</td>
<td>$2,700.00</td>
</tr>
<tr>
<td></td>
<td>Televisions</td>
<td>$101,212.00</td>
<td>121</td>
<td>$836.46</td>
<td>$275.00</td>
<td>$6,500.00</td>
</tr>
<tr>
<td></td>
<td>Video Production</td>
<td>$89,489.00</td>
<td>325</td>
<td>$275.35</td>
<td>$78.00</td>
<td>$1,410.00</td>
</tr>
<tr>
<td>South America</td>
<td>Accessories</td>
<td>$25,723.00</td>
<td>103</td>
<td>$249.74</td>
<td>$16.00</td>
<td>$1,470.00</td>
</tr>
<tr>
<td></td>
<td>Camcorder</td>
<td>$22,686.00</td>
<td>110</td>
<td>$206.24</td>
<td>$60.00</td>
<td>$1,000.00</td>
</tr>
<tr>
<td></td>
<td>Computers</td>
<td>$7,845.00</td>
<td>47</td>
<td>$166.91</td>
<td>$31.00</td>
<td>$501.00</td>
</tr>
<tr>
<td></td>
<td>Media Player</td>
<td>$60,183.00</td>
<td>161</td>
<td>$373.81</td>
<td>$36.00</td>
<td>$1,480.00</td>
</tr>
<tr>
<td></td>
<td>Stereo Systems</td>
<td>$81,823.00</td>
<td>284</td>
<td>$288.11</td>
<td>$48.00</td>
<td>$2,700.00</td>
</tr>
<tr>
<td></td>
<td>Televisions</td>
<td>$25,800.00</td>
<td>25</td>
<td>$1,032.00</td>
<td>$275.00</td>
<td>$9,750.00</td>
</tr>
<tr>
<td></td>
<td>Video Production</td>
<td>$11,740.00</td>
<td>41</td>
<td>$286.34</td>
<td>$78.00</td>
<td>$1,410.00</td>
</tr>
</tbody>
</table>

**Total** $2,940,745.00  9,961

### Reference: Usage Notes for Report-Level Prefix Operators

- These operators can be used on a field in a heading or footing without being referenced in a display command in the request.
They work in a heading or footing for real or virtual (DEFINE) fields. They work in a display command field list on real fields, virtual (DEFINE) fields, and calculated (COMPUTE) values that are calculated prior to their use in the request.

They can be used in subheadings and subfootings to reference the total value for the entire report.

Controlling Processing of Delimited Files

The SET PCOMMA=DFIX command causes delimited files with SUFFIX=COM, COMT, TAB, and TABT to be processed through the Adapter for DFIX. This processing provides more complete and meaningful messages and some changes to the processing of missing values when two delimiters in a row are encountered. With DFIX processing, a missing value is assigned to the field.

In order to be eligible for DFIX processing, the delimited file must satisfy the following requirements.

- Each record must be completely contained on one line and terminated with the crlf (carriage return/line feed) character combination. If a break is found within a record, the following message displays.

  (FOC229) ERROR READING EXTERNAL DATA FILE: file

- The ENCLOSURE can be only in the first position after the delimiter for COM (new) and COMT records. Otherwise, it will not be recognized.

- The number of fields on a line cannot exceed the number of fields defined in the Master File. If there are more, the following message displays.

  (FOC416) MORE VALUES SUPPLIED THAN FIELDS IN THE MASTER FILE DESCRIPTION

Format Display Option for Percentage Calculation

The lowercase p display option converts a number to a percent by multiplying it by 100. It then displays the number followed by a percent symbol (%). This is different from the % format option, which does not calculate a percent, it just displays a percent symbol after the number.

**Note:** This option is not supported for output formats XLSX and EXL2K.

**Syntax:** How to Display a Percentage

```
field/floatfmt p
```
where:

- **field**
  Is a numeric field.

- **floatfmt**
  Is a numeric format of type F or D.

**Example:** **Displaying a Percentage Using Display Option p**

The following request displays the PCT_INC field (which is a decimal value) and redisplays it as a percentage:

```plaintext
TABLE FILE EMPLOYEE
PRINT PCT_INC PCT_INC/F10p
END
```

The output is:

<table>
<thead>
<tr>
<th>PCT_INC</th>
<th>PCT_INC</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>10%</td>
</tr>
<tr>
<td>.12</td>
<td>12%</td>
</tr>
<tr>
<td>.10</td>
<td>10%</td>
</tr>
<tr>
<td>.04</td>
<td>4%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.05</td>
<td>5%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.10</td>
<td>10%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.15</td>
<td>15%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.07</td>
<td>7%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.04</td>
<td>4%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
<tr>
<td>.05</td>
<td>5%</td>
</tr>
<tr>
<td>.00</td>
<td>0%</td>
</tr>
</tbody>
</table>

**Start of Week Parameter for the DTRUNC Function**

Given a date or date-time value and a time period, the DTRUNC function returns the date that is the beginning of that period. The parameter WEEK has been added, which returns the date that represents the first date of the given week. By default, the first day will be Sunday, but this can be changed using the WEEKFIRST parameter.
**Example:** Using the Start of Week Parameter for DTRUNC

The following request returns the date that is the start of the week for the start date of certain employees:

```plaintext
DEFINE FILE WF_RETAIL
DAY1/WT = DTRUNC(START_DATE, DAY);
WKSTART/YYMD = DTRUNC(START_DATE, WEEK);
DAY2/WT = DTRUNC(WKSTART, DAY);
END
TABLE FILE WF_RETAIL
PRINT START_DATE
DAY1 AS 'DOW 1'
WKSTART AS 'Start, of Week'
DAY2 AS 'DOW 2'
BY EMPLOYEE_NUMBER
WHERE START_DATE GT '20130101'
WHERE EMPLOYEE_NUMBER CONTAINS 'AH'
ON TABLE SET PAGE NOPAGE
END
```

The output is:

<table>
<thead>
<tr>
<th>Employee Number</th>
<th>Start Date</th>
<th>Start DOW 1</th>
<th>Start of Week</th>
<th>Start DOW 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AH118</td>
<td>2013/01/15</td>
<td>TUE</td>
<td>2013/01/13</td>
<td>SUN</td>
</tr>
<tr>
<td>AH2272</td>
<td>2013/01/17</td>
<td>THU</td>
<td>2013/01/13</td>
<td>SUN</td>
</tr>
<tr>
<td>AH288</td>
<td>2013/11/11</td>
<td>MON</td>
<td>2013/11/10</td>
<td>SUN</td>
</tr>
<tr>
<td>AH3520</td>
<td>2013/09/23</td>
<td>MON</td>
<td>2013/09/22</td>
<td>SUN</td>
</tr>
<tr>
<td>AH3591</td>
<td>2013/09/22</td>
<td>SUN</td>
<td>2013/09/22</td>
<td>SUN</td>
</tr>
<tr>
<td>AH5177</td>
<td>2013/07/21</td>
<td>SUN</td>
<td>2013/07/21</td>
<td>SUN</td>
</tr>
</tbody>
</table>

**Function DTRUNC Returns the Last Day of a Calendar Period**

Given a date or timestamp and a component, DTRUNC returned the first date within the period specified by that component.

The following components have been added that return the last date within the period specified by the component.

- YEAR_END
- QUARTER_END
- MONTH_END
- WEEK_END

This function is available in both WebFOCUS and SQL.

The calculators in the Synonym Editor and Data Flow show the new parameter values.
**Note:** You can change which day is considered the first day of the week using the WEEKFIRST parameter. By default, it is Sunday.

**Example:** **Returning the Date of the First and Last Days of a Week**

The following request returns the dates that corresponds to the first day of the week and the last day of the week for the given date.

```plaintext
DEFINE FILE WF_RETAIL
WEEKSTART/YYMD = DTRUNC(START_DATE, WEEK);
WEEKEND/YYMD = DTRUNC(START_DATE, WEEK_END);
END
TABLE FILE WF_RETAIL
PRINT START_DATE WEEKSTART AS 'Start, of Week'
WEEKEND AS 'End, of Week'
BY EMPLOYEE_NUMBER
WHERE EMPLOYEE_NUMBER CONTAINS 'AH1'
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Employee Number</th>
<th>Start Date</th>
<th>Start of Week</th>
<th>End of Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>AH118</td>
<td>2013/01/15</td>
<td>2013/01/13</td>
<td>2013/01/19</td>
</tr>
<tr>
<td>AH1994</td>
<td>2006/01/01</td>
<td>2006/01/01</td>
<td>2006/01/07</td>
</tr>
</tbody>
</table>

**Simplified Numeric Functions**

New numeric functions have been developed that make it easier to understand and enter the required arguments. These functions have streamlined parameter lists, similar to those used by SQL functions. In some cases, these simplified functions provide slightly different functionality than previous versions of similar functions.

The simplified functions do not have an output argument. Each function returns a value that has a specific data type.

When used in a request against a relational data source, these functions are optimized (passed to the RDBMS for processing).
Note:

- The simplified numeric functions are supported in Dialogue Manager.
- The simplified numeric functions are not supported in WebFOCUS Maintain.

**CEILING: Returning the Smallest Integer Value Greater Than or Equal to a Value**

CEILING returns the smallest integer value that is greater than or equal to a number.

**Syntax:**

How to Return the Smallest Integer Greater Than or Equal to a Number

CEILING(number)

where:

- **number**
  - Numeric
    - Is the number whose ceiling will be returned. The output data type is the same as the input data type.

**Example:**

Returning the Ceiling of a Number

In the following request, CEILING returns the smallest integer greater than or equal to the GROSS_PROFIT_US value.

```plaintext
DEFINE FILE WF_RETAIL_LITE
CEIL1/D7.2= CEILING(GROSS_PROFIT_US);
END
TABLE FILE WF_RETAIL_LITE
PRINT GROSS_PROFIT_US/D9.2  CEIL1
ON TABLE SET PAGE NOPAGE
END

CEILING returns the smallest integer larger than the value in GROSS_PROFIT_US:

CEILING(GROSS_PROFIT_US)
```
The partial output follows. Note that even though the value returned is an integer, it is returned with the same format as the CEIL1 field (D7.2):

<table>
<thead>
<tr>
<th>Gross Profit</th>
<th>CEIL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>165.00</td>
<td>165.00</td>
</tr>
<tr>
<td>13.99</td>
<td>14.00</td>
</tr>
<tr>
<td>60.99</td>
<td>61.00</td>
</tr>
<tr>
<td>225.98</td>
<td>226.00</td>
</tr>
<tr>
<td>79.99</td>
<td>80.00</td>
</tr>
<tr>
<td>44.59</td>
<td>45.00</td>
</tr>
<tr>
<td>94.30</td>
<td>95.00</td>
</tr>
<tr>
<td>238.50</td>
<td>239.00</td>
</tr>
<tr>
<td>199.99</td>
<td>200.00</td>
</tr>
<tr>
<td>68.99</td>
<td>69.00</td>
</tr>
<tr>
<td>63.58</td>
<td>64.00</td>
</tr>
<tr>
<td>129.99</td>
<td>130.00</td>
</tr>
<tr>
<td>37.49</td>
<td>38.00</td>
</tr>
<tr>
<td>75.99</td>
<td>76.00</td>
</tr>
<tr>
<td>13.99</td>
<td>14.00</td>
</tr>
<tr>
<td>119.00</td>
<td>119.00</td>
</tr>
<tr>
<td>-30.01</td>
<td>-30.00</td>
</tr>
<tr>
<td>54.99</td>
<td>55.00</td>
</tr>
<tr>
<td>189.98</td>
<td>190.00</td>
</tr>
<tr>
<td>44.59</td>
<td>45.00</td>
</tr>
<tr>
<td>91.98</td>
<td>92.00</td>
</tr>
<tr>
<td>89.00</td>
<td>89.00</td>
</tr>
<tr>
<td>59.50</td>
<td>60.00</td>
</tr>
<tr>
<td>129.99</td>
<td>130.00</td>
</tr>
<tr>
<td>54.00</td>
<td>54.00</td>
</tr>
<tr>
<td>109.98</td>
<td>110.00</td>
</tr>
<tr>
<td>98.99</td>
<td>99.00</td>
</tr>
<tr>
<td>98.99</td>
<td>99.00</td>
</tr>
<tr>
<td>99.99</td>
<td>100.00</td>
</tr>
<tr>
<td>44.59</td>
<td>45.00</td>
</tr>
</tbody>
</table>

**EXPONENT: Raising e to a Power**

EXPONENT raises the constant e to a power.

**Syntax:**

How to Raise the Constant e to a Power

EXPONENT(*power*)

where:

*power*

Numeric

Is the power to which to raise e. The output data type is numeric.
Example: Raising e to a Power

The following request prints the value of e and the value of e raised to the fifth power.

```
DEFINE FILE WF_RETAIL_LITE
EXP1/D12.5 = EXPONENT(1);
EXP2/D12.5 = EXPONENT(5);
END
TABLE FILE WF_RETAIL_LITE
PRINT EXP1 EXP2
BY BUSINESS_REGION AS Region
WHERE BUSINESS_REGION EQ 'EMEA'
WHERE RECORDLIMIT EQ 1
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Region</th>
<th>EXP1</th>
<th>EXP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>2.71828</td>
<td>148.41316</td>
</tr>
</tbody>
</table>

FLOOR: Returning the Largest Integer Less Than or Equal to a Value

FLOOR returns the largest integer value that is less than or equal to a number.

Syntax: How to Return the Largest Integer Less Than or Equal to a Number

```
FLOOR(number)
```

where:

- `number` Numeric
  
  Is the number whose floor will be returned. The output data type is the same as the input data type.
Example: Returning the Floor of a Number

In the following request, FLOOR returns the largest integer less than or equal to the GROSS_PROFIT_US value.

```
DEFINE FILE WF_RETAIL_LITE
FLOOR1/D7.2= FLOOR(GROSS_PROFIT_US);
END
TABLE FILE WF_RETAIL_LITE
PRINT GROSS_PROFIT_US/D9.2  FLOOR1
ON TABLE SET PAGE NOPAGE
END
```

FLOOR returns the smallest integer larger than the value in GROSS_PROFIT_US:

```
FLOOR(GROSS_PROFIT_US)
```

Partial output follows. Note that even though the value returned is an integer, it is returned with the same format as the FLOOR1 field (D7.2):

<table>
<thead>
<tr>
<th>Gross Profit</th>
<th>FLOOR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>165.00</td>
<td>165.00</td>
</tr>
<tr>
<td>13.99</td>
<td>13.00</td>
</tr>
<tr>
<td>60.99</td>
<td>60.00</td>
</tr>
<tr>
<td>225.98</td>
<td>225.00</td>
</tr>
<tr>
<td>79.99</td>
<td>79.00</td>
</tr>
<tr>
<td>44.59</td>
<td>44.00</td>
</tr>
<tr>
<td>94.30</td>
<td>94.00</td>
</tr>
<tr>
<td>238.50</td>
<td>238.00</td>
</tr>
<tr>
<td>199.99</td>
<td>199.00</td>
</tr>
<tr>
<td>68.99</td>
<td>68.00</td>
</tr>
<tr>
<td>63.58</td>
<td>63.00</td>
</tr>
<tr>
<td>129.99</td>
<td>129.00</td>
</tr>
<tr>
<td>37.49</td>
<td>37.00</td>
</tr>
<tr>
<td>75.99</td>
<td>75.00</td>
</tr>
<tr>
<td>13.99</td>
<td>13.00</td>
</tr>
<tr>
<td>119.00</td>
<td>119.00</td>
</tr>
<tr>
<td>-30.01</td>
<td>-31.00</td>
</tr>
<tr>
<td>54.99</td>
<td>54.00</td>
</tr>
<tr>
<td>189.98</td>
<td>189.00</td>
</tr>
<tr>
<td>44.59</td>
<td>44.00</td>
</tr>
<tr>
<td>91.98</td>
<td>91.00</td>
</tr>
<tr>
<td>89.00</td>
<td>89.00</td>
</tr>
<tr>
<td>59.50</td>
<td>59.00</td>
</tr>
<tr>
<td>129.99</td>
<td>129.00</td>
</tr>
<tr>
<td>54.00</td>
<td>54.00</td>
</tr>
<tr>
<td>109.98</td>
<td>109.00</td>
</tr>
<tr>
<td>98.99</td>
<td>98.00</td>
</tr>
<tr>
<td>98.99</td>
<td>98.00</td>
</tr>
<tr>
<td>99.99</td>
<td>99.00</td>
</tr>
<tr>
<td>44.59</td>
<td>44.00</td>
</tr>
</tbody>
</table>
MOD: Calculating the Remainder From a Division

MOD calculates the remainder from a division. The output data type is the same as the input data type.

Syntax: How to Calculate the Remainder From a Division

MOD(dividend, divisor)

where:

dividend
Numeric
Is the value to divide.

divisor
Numeric
Is the value to divide by.

If the divisor is zero (0), the dividend is returned.

Example: Calculating the Remainder From a Division

In the following request, MOD returns the remainder of PRICE_DOLLARS divided by DAYSDELAYED:

DEFINE FILE WF_RETAIL_LITE
MOD1/D7.2= MOD(PRICE_DOLLARS, DAYSDELAYED);
END
TABLE FILE WF_RETAIL_LITE
PRINT PRICE_DOLLARS/D7.2 daysdelayed/I5 MOD1
WHERE DAYSDELAYED GT 1
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT WP
END
Partial output follows:

<table>
<thead>
<tr>
<th>Price Dollars</th>
<th>Days Delayed</th>
<th>MOD1</th>
</tr>
</thead>
<tbody>
<tr>
<td>399.00</td>
<td>3</td>
<td>.00</td>
</tr>
<tr>
<td>489.99</td>
<td>3</td>
<td>.99</td>
</tr>
<tr>
<td>786.50</td>
<td>2</td>
<td>.50</td>
</tr>
<tr>
<td>599.99</td>
<td>4</td>
<td>3.99</td>
</tr>
<tr>
<td>29.99</td>
<td>4</td>
<td>1.99</td>
</tr>
<tr>
<td>169.00</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>219.99</td>
<td>2</td>
<td>1.99</td>
</tr>
<tr>
<td>280.00</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>79.99</td>
<td>4</td>
<td>3.99</td>
</tr>
<tr>
<td>145.99</td>
<td>2</td>
<td>1.99</td>
</tr>
<tr>
<td>399.99</td>
<td>3</td>
<td>.99</td>
</tr>
<tr>
<td>349.99</td>
<td>3</td>
<td>1.99</td>
</tr>
<tr>
<td>169.00</td>
<td>3</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**POWER: Raising a Value to a Power**

POWER raises a base value to a power.

**Syntax:**

How to Raise a Value to a Power

\[
\text{POWER}(base, \ power)
\]

where:

- **base**
  - Numeric
  - Is the value to raise to a power. The output value has the same data type as the base value. If the base value is integer, negative power values will result in truncation.

- **power**
  - Numeric
  - Is the power to which to raise the base value.
**Example:** Raising a Base Value to a Power

In the following request, POWER returns the value COGS_US/20.00 raised to the power stored in DAYSDELAYED:

```plaintext
DEFINE FILE WF_RETAIL_LITE
BASE=COGS_US/20.00;
POWER1= POWER(COGS_US/20.00,DAYSDELAYED);
END
TABLE FILE WF_RETAIL_LITE
PRINT BASE IN 15 DAYSDELAYED POWER1
BY PRODUCT_CATEGORY
WHERE PRODUCT_CATEGORY EQ 'Computers'
WHERE DAYSDELAYED NE 0
ON TABLE SET PAGE NOPAGE
END
```

Partial output follows:

<table>
<thead>
<tr>
<th>Product Category</th>
<th>BASE</th>
<th>Days Delayed</th>
<th>POWER1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computers</td>
<td>12.15</td>
<td>3</td>
<td>1,793.61</td>
</tr>
<tr>
<td></td>
<td>16.70</td>
<td>2</td>
<td>278.89</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>1</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>8.10</td>
<td>2</td>
<td>65.61</td>
</tr>
<tr>
<td></td>
<td>4.05</td>
<td>1</td>
<td>4.05</td>
</tr>
<tr>
<td></td>
<td>4.05</td>
<td>2</td>
<td>16.40</td>
</tr>
<tr>
<td></td>
<td>4.05</td>
<td>4</td>
<td>269.04</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>1</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>16.70</td>
<td>1</td>
<td>16.70</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>3</td>
<td>582.18</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>1</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>4.05</td>
<td>1</td>
<td>4.05</td>
</tr>
<tr>
<td></td>
<td>4.05</td>
<td>1</td>
<td>4.05</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>4</td>
<td>4,861.23</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>-1</td>
<td>.12</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>1</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>8.35</td>
<td>3</td>
<td>582.18</td>
</tr>
</tbody>
</table>

**Simplified Character Functions**

New simplified character functions CONCAT, REGEX, REPLACE, and SPLIT functions have been added.

**LAST_NONBLANK:** Retrieving the Last Field Value That is Neither Blank nor Missing

LAST_NONBLANK retrieves the last field value that is neither blank nor missing. If all previous values are either blank or missing, LAST_NONBLANK returns a missing value.
**Syntax:**  How to Return the Last Value That is Neither Blank nor Missing

\[
\text{LAST\_NONBLANK}(\text{field})
\]

where:

\textit{field}  
Is the field name whose last non-blank value is to be retrieved. If the current value is not blank or missing, the current value is returned.

**Note:** LAST\_NONBLANK cannot be used in a compound expression, for example, as part of an IF condition.

**Example:**  Retrieving the Last Non-Blank Value

The following request shows the SALARY field, the LAST\_NONBLANK value from a DEFINE, and the LAST\_NONBLANK value from a COMPUTE.

```
DEFINE FILE EMPLOYEE
MYSAL/I5 MISSING ON=IF DEPARTMENT EQ 'MIS' THEN SALARY ELSE MISSING;
LAST\_NONBLANK1/I5 MISSING ON= LAST\_NONBLANK(MYSAL);
END
TABLE FILE EMPLOYEE
PRINT LAST\_NAME MYSAL  LAST\_NONBLANK1
COMPUTE LAST\_NONBLANK2/I5 MISSING ON=LAST\_NONBLANK(MYSAL);
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
END
```
The output is shown in the following image.

<table>
<thead>
<tr>
<th>LAST_NAME</th>
<th>MYSAL</th>
<th>LAST_NONBLANK1</th>
<th>LAST_NONBLANK2</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEVENS</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>STEVENS</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>SMITH</td>
<td>13200</td>
<td>13200</td>
<td>13200</td>
</tr>
<tr>
<td>JONES</td>
<td>18480</td>
<td>18480</td>
<td>18480</td>
</tr>
<tr>
<td>JONES</td>
<td>17750</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>SMITH</td>
<td>.</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>SMITH</td>
<td>.</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>BANNING</td>
<td>.</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>IRVING</td>
<td>.</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>IRVING</td>
<td>.</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>ROMANS</td>
<td>.</td>
<td>17750</td>
<td>17750</td>
</tr>
<tr>
<td>MCCOY</td>
<td>18480</td>
<td>18480</td>
<td>18480</td>
</tr>
<tr>
<td>BLACKWOOD</td>
<td>21780</td>
<td>21780</td>
<td>21780</td>
</tr>
<tr>
<td>MCKNIGHT</td>
<td>.</td>
<td>21780</td>
<td>21780</td>
</tr>
<tr>
<td>MCKNIGHT</td>
<td>.</td>
<td>21780</td>
<td>21780</td>
</tr>
<tr>
<td>GREENSPAN</td>
<td>9000</td>
<td>9000</td>
<td>9000</td>
</tr>
<tr>
<td>GREENSPAN</td>
<td>8650</td>
<td>8650</td>
<td>8650</td>
</tr>
<tr>
<td>CROSS</td>
<td>27062</td>
<td>27062</td>
<td>27062</td>
</tr>
<tr>
<td>CROSS</td>
<td>25775</td>
<td>25775</td>
<td>25775</td>
</tr>
</tbody>
</table>

**REGEX: Matching a String to a Regular Expression**

The REGEX function matches a string to a regular expression and returns true (1) if it matches and false (0) if it does not match.

A regular expression is a sequence of special characters and literal characters that you can combine to form a search pattern.

Many references for regular expressions exist on the web.

For a basic summary, see the section **Summary of Regular Expressions** in Chapter 2, Security, of the Server Administration manual.

**Syntax:** How to Match a String to a Regular Expression

```
REGEX(string, regular_expression)
```
where:

**string**
Alphanumeric

Is the character string to match.

**regular_expression**
Alphanumeric

Is a regular expression enclosed in single quotation marks ('') constructed using literals and metacharacters. The following metacharacters are supported:

- . represents any single character
- * represents zero or more occurrences
- + represents one or more occurrences
- ? represents zero or one occurrence
- ^ represents beginning of line
- $ represents end of line
- [] represents any one character in the set listed within the brackets
- [^] represents any one character not in the set listed within the brackets
- | represents the Or operator
- \ is the Escape Special Character
- () contains a character sequence

For example, the regular expression '^Ste(v|ph)en$' matches values starting with Ste followed by either ph or v, and ending with en.

**Note:** The output value is numeric.

**Example:**  **Matching a String Against a Regular Expression**

The following request matches the FIRSTNAME field against the regular expression '^Sara(h?)$', which matches Sara or Sarah:

```
TABLE FILE WF_RETAIL_LITE
PRINT FIRSTNAME AND COMPUTE
REG1/I1=REGEX(FIRSTNAME,'^Sara(h?)$') ;
BY LASTNAME/A10
WHERE LASTNAME EQ 'Allen'
END
```
The output is

<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>Name</th>
<th>REG1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allen</td>
<td>Penny</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Rosemary</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Amber</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Julie</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Sarah</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Leo</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Margret</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Donna</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Damian</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Alexander</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Diego</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Amber</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Susan</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Amber</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Sara</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Sara</td>
<td>1</td>
</tr>
</tbody>
</table>

**CONCAT: Concatenating Strings After Removing Trailing Blanks From the First**

CONCAT removes trailing blanks from a string and then concatenates another string to it. The output is returned as variable length alphanumeric.

**Syntax:**

How to Concatenate Strings After Removing Trailing Blanks From the First

`CONCAT(string1, string2)`

where:

`string2`

Alphanumeric

Is a string whose trailing blanks will be removed.

`string1`

Alphanumeric

Is a string whose leading and trailing blanks will be preserved.
Example:  Concatenating Strings After Removing Blanks From the First

The following request concatenates city names with state names. Note that the city and state names are converted to fixed length alphanumeric fields before concatenation.

```
DEFINE FILE WF_RETAIL_LITE
CITY/A50 = CITY_NAME;
STATE/A50 = STATE_PROV_NAME;
CONCAT_CS/A100 = CONCAT(CITY,STATE);
END

TABLE FILE WF_RETAIL_LITE
SUM CITY AS City STATE AS State CONCAT_CS AS Concatenation
BY STATE_PROV_NAME NOPRINT
WHERE COUNTRY_NAME EQ 'United States'
WHERE STATE LE 'Louisiana'
ON TABLE SET PAGE NOLEAD
END
```
REPLACE: Replacing a String

REPLACE replaces all instances of a search string in an input string with the given replacement string. The output is always variable length alphanumeric with a length determined by the input parameters.
**Syntax:** How to Replace all Instances of a String

\[ \text{REPLACE}(\text{input\_string}, \text{search\_string}, \text{replacement}) \]

where:

- **input\_string**
  - Alphanumeric or text (An, AnV, TX)
  - Is the input string.

- **search\_string**
  - Alphanumeric or text (An, AnV, TX)
  - Is the string to search for within the input string.

- **replacement**
  - Alphanumeric or text (An, AnV, TX)
  - Is the replacement string to be substituted for the search string. It can be a null string ("").

**Example:** Replacing a String

REPLACE replaces the string 'South' in the Country Name with the string 'S.'

```
SET TRACEUSER = ON
SET TRACEON = STMTRACE//CLIENT
SET TRACESTAMP=OFF
DEFINE FILE WF_RETAIL_LITE
NEWNAME/A20 = REPLACE(COUNTRY_NAME, 'SOUTH', 'S.');
END
TABLE FILE WF_RETAIL_LITE
SUM COUNTRY_NAME
BY NEWNAME AS 'New,Name'
WHERE COUNTRY_NAME LIKE 'S%'
ON TABLE SET PAGE NOLEAD
END

REPLACE(COUNTRY_NAME, 'SOUTH', 'S.');
```

For South Africa, the result is S. Africa.

The generated SQL passes the REPLACE function to the DBMS REPLACE function.
SELECT
  REPLACE(T3."COUNTRY_NAME","SOUTH","S.'"),
  MAX(T3."COUNTRY_NAME")
FROM
  wrd_wf_retail_geography T3
WHERE
  (T3."COUNTRY_NAME" LIKE 'S%')
GROUP BY
  REPLACE(T3."COUNTRY_NAME","SOUTH","S.'")
ORDER BY
  REPLACE(T3."COUNTRY_NAME","SOUTH","S.'");

The output is shown in the following image.

<table>
<thead>
<tr>
<th>New Name</th>
<th>Customer Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. Africa</td>
<td>South Africa</td>
</tr>
<tr>
<td>S. Korea</td>
<td>South Korea</td>
</tr>
<tr>
<td>Singapore</td>
<td>Singapore</td>
</tr>
<tr>
<td>Spain</td>
<td>Spain</td>
</tr>
<tr>
<td>Sweden</td>
<td>Sweden</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Switzerland</td>
</tr>
</tbody>
</table>

**Example:**  Replacing All Instances of a String

In the following request, the virtual field DAYNAME1 is the string DAY1 with all instances of the string 'DAY' replaced with the string 'day'. The virtual field DAYNAME2 has all instances of the string 'DAY' removed.

```
DEFINE FILE WF_RETAIL
DAY1/A30 = 'SUNDAY MONDAY TUESDAY';
DAYNAME1/A30 = REPLACE(DAY1, 'DAY', 'day');
DAYNAME2/A30 = REPLACE(DAY1, 'DAY', '');
END
TABLE FILE WF_RETAIL
PRINT DAY1 OVER
DAYNAME1 OVER
DAYNAME2
WHERE EMPLOYEE_NUMBER EQ 'AH118'
ON TABLE SET PAGE NOPAGE
END
```
The output is:

```
DAY1      SUNDAY MONDAY TUESDAY
DAYNAME1  SUNday MONday TUESday
DAYNAME2  SUN MON TUES
```

**SPLIT: Extracting an Element From a String**

The SPLIT function returns a specific type of element from a string. The output is returned as variable length alphanumeric.

**Syntax:**

**How to Extract an Element From a String**

```
SPLIT(element, string)
```

where:

- **element**
  - Can be one of the following keywords:
    - **EMAIL_DOMAIN.** Is the domain name portion of an email address in the string.
    - **EMAIL_USERID.** Is the user ID portion of an email address in the string.
    - **URL_PROTOCOL.** Is the URL protocol in the string.
    - **URL_HOST.** Is the host name of the URL in the string.
    - **URL_PORT.** Is the port number of the URL in the string.
    - **URL_PATH.** Is the URL path in the string.
    - **NAME_FIRST.** Is the first token (group of characters) in the string. Tokens are delimited by blanks.
    - **NAME_LAST.** Is the last token (group of characters) in the string. Tokens are delimited by blanks.

- **string**
  - Alphanumeric
  - Is the string from which the element will be extracted.
**Example:** Extracting an Element From a String

The following request defines strings and extracts elements from them.

```plaintext
DEFINE FILE WF_RETAIL_LITE
STRING1/A50 WITH COUNTRY_NAME= 'http://www.informationbuilders.com';
STRING2/A20 = 'user1@ibi.com';
STRING3/A20 = 'Louisa May Alcott';
Protocol/A20 = SPLIT(URL_PROTOCOL, STRING1);
Path/A50 = SPLIT(URL_PATH, STRING1);
Domain/A20 = SPLIT(EMAIL_DOMAIN, STRING2);
User/A20 = SPLIT(EMAIL_USERID, STRING2);
First/A10 = SPLIT(NAME_FIRST, STRING3);
Last/A10 = SPLIT(NAME_LAST, STRING3);
END
TABLE FILE WF_RETAIL_LITE
SUM Protocol Path User Domain First Last
ON TABLE SET PAGE NOLEAD
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Path</th>
<th>User</th>
<th>Domain</th>
<th>First</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td><a href="http://www.informationbuilders.com">http://www.informationbuilders.com</a></td>
<td>user1</td>
<td>ibi.com</td>
<td>Louisa</td>
<td>May Alcott</td>
</tr>
</tbody>
</table>

**CHECKMD5: Computing an MD5 Hash Check Value**

CHECKMD5 takes an alphanumeric input value and returns a 128-bit value in a fixed length alphanumeric string, using the MD5 hash function. A hash function is any function that can be used to map data of arbitrary size to data of fixed size. The values returned by a hash function are called hash values. They can be used for assuring the integrity of transmitted data.

**Syntax:** How to Compute an MD5 Hash Check Value

```plaintext
CHECKMD5(buffer)
```

where:

**buffer**

Is a data buffer whose hash value is to be calculated. It can be a set of data of different types presented as a single field, or a group field in one of the following data type formats: An, AnV, or TXn.
Example:  Calculating an MD5 Hash Check Value

The following request calculates an MD5 hash check value and converts it to an alphanumeric hexadecimal value for display.

```
DEFINE FILE WF_RETAIL_LITE
MD5/A32 = HEXTYPE(CHECKMD5(PRODUCT_CATEGORY));
END
TABLE FILE WF_RETAIL_LITE
SUM MD5
BY PRODUCT_CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
   TYPE=REPORT,FONT=COURIER,$
   ENDSTYLE
END
```

The output is shown in the following image. The monospaced font shows that although the input values have varying length, the output has a fixed length.

<table>
<thead>
<tr>
<th>Product Category</th>
<th>MD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>98EDB85B00D9527AD5ACEBE451B3FAE6</td>
</tr>
<tr>
<td>Camcorder</td>
<td>612A923BDD05C2231F81991B8D12A3A1</td>
</tr>
<tr>
<td>Computers</td>
<td>45888A4DA062F16A099A7F7C6CC15EE0</td>
</tr>
<tr>
<td>Media Player</td>
<td>D34BEA29F24AF9FDE2E10B3E1D857CF9</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>3AA9FFE9806E269A7EB066A84092F0A3</td>
</tr>
<tr>
<td>Televisions</td>
<td>A3B5BC99DD2B42627EF64A4FCAAB0B2</td>
</tr>
<tr>
<td>Video Production</td>
<td>60913E95848330A2C4A5D921E7C8BB14</td>
</tr>
</tbody>
</table>

CHECKSUM: Computing a Hash Sum

CHECKSUM computes a hash sum, called the checksum, of its input parameter, as a whole number in format I11. This can be used for equality search of the fields. A checksum is a hash sum used to ensure the integrity of a file after it has been transmitted from one storage device to another.

Syntax: How to Compute a CHECKSUM Hash Value

```
CHECKSUM(buffer)
```
where:

*buffer*

Is a data buffer whose hash index is to be calculated. It can be a set of data of different types presented as a single field, in one of the following data type formats: An, AnV, or TXn.

**Example:** Calculating a CHECKSUM Hash Value

The following request computes a checksum hash value.

```
DEFINE FILE WF_RETAIL_LITE
CHKSUM/I11 = (CHECKSUM(PRODUCT_CATEGORY));
END
TABLE FILE WF_RETAIL_LITE
PRINT CHKSUM
BY PRODUCT_CATEGORY
WHERE PRODUCT_CATEGORY NE LAST PRODUCT_CATEGORY
ON TABLE SET PAGE NOLEAD
END
```

The output is shown in the following image.

```
<table>
<thead>
<tr>
<th>Product Category</th>
<th>CHKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>-830549649</td>
</tr>
<tr>
<td>Camcorder</td>
<td>-912058982</td>
</tr>
<tr>
<td>Computers</td>
<td>-469201037</td>
</tr>
<tr>
<td>Media Player</td>
<td>-1760917009</td>
</tr>
<tr>
<td>Stereo Systems</td>
<td>-1853215244</td>
</tr>
<tr>
<td>Televisions</td>
<td>810407163</td>
</tr>
<tr>
<td>Video Production</td>
<td>275494446</td>
</tr>
</tbody>
</table>
```

**Simplified Conversion Functions**

New simplified conversion functions CHAR, CTRL_CHAR, EDIT2, HEXTYPE, TO_INTEGER, and TO_NUMBER have been added.
CHAR: Returning a Character Based on a Numeric Code

The CHAR function accepts a decimal integer and returns the character identified by that number converted to ASCII or EBCDIC, depending on the operating environment. The output is returned as variable length alphanumeric. If the number is above the range of valid characters, a null is returned.

Syntax: How to Return a Character Based on a Numeric Code

CHAR(number_code)

where:

number_code

Integer

Is a field, number, or numeric expression whose whole absolute value will be used as a number code to retrieve an output character.

For example, a TAB character is returned by CHAR(9) in ASCII environments, or by CHAR(5) in EBCDIC environments.

Example: Using the CHAR Function to Insert Control Characters Into a String

The following request defines a field with carriage return (CHAR(13)) and line feed (CHAR(10)) characters inserted between the words HELLO and GOODBYE (in an ASCII environment). To show that these characters were inserted, the output is generated in PDF format and the StyleSheet attribute LINEBREAK='CRLF' is used to have these characters respected and print the field value on two lines.

```
DEFINE FILE WF_RETAIL_LITE
MYFIELD/A20 WITH COUNTRY_NAME='HELLO' | CHAR(13) | CHAR(10) | 'GOODBYE';
END
TABLE FILE WF_RETAIL_LITE
SUM MYFIELD
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT,LINEBREAK='CRLF',$
ENDSTYLE
END
```
CTRLCHAR: Returning a Non-Printable Control Character

The CTRLCHAR function returns a nonprintable control character specific to the running operating environment, based on a supported list of keywords. The output is returned as variable length alphanumeric.

**Syntax:**

How to Return a Non-Printable Control Character

CTRLCHAR (ctrl_char)

where:

*ctrl_char*

Is one of the following keywords.

- **NUL** returns a null character.
- **SOH** returns a start of heading character.
- **STX** returns a start of text character.
- **ETX** returns an end of text character.
- **EOT** returns an end of transmission character.
- **ENQ** returns an enquiry character.
- **ACK** returns an acknowledge character.
- **BEL** returns a bell or beep character.
- **BS** returns a backspace character.
- **TAB** or **HT** returns a horizontal tab character.
- **LF** returns a line feed character.
- **VT** returns a vertical tab character.
- **FF** returns a form feed (top of page) character.
- **CR** returns a carriage control character.
- **SO** returns a shift out character.
- **SI** returns a shift in character.
- **DLE** returns a data link escape character.
- **DC1** or **XON** returns a device control 1 character.
- **DC2** returns a device control 2 character.
- **DC3** or **XOFF** returns a device control 3 character.
- **DC4** returns a device control 4 character.
- **NAK** returns a negative acknowledge character.
- **SYN** returns a synchronous idle character.
- **ETB** returns an end of transmission block character.
- **CAN** returns a cancel character.
- **EM** returns an end of medium character.
- **SUB** returns a substitute character.
- **ESC** returns an escape, prefix, or altmode character.
- **FS** returns a file separator character.
- **GS** returns a group separator character.
- **RS** returns a record separator character.
- **US** returns a unit separator character.
- **DEL** returns a delete, rubout, or interrupt character.
**Example:** Using the CTRLCHAR Function to Insert Control Characters Into a String

The following request defines a field with carriage return (CTRLCHAR(CR)) and line feed (CTRLCHAR(LF)) characters inserted between the words HELLO and GOODBYE. To show that these characters were inserted, the output is generated in PDF format and the StyleSheet attribute LINEBREAK='CRLF' is used to have these characters respected and print the field value on two lines.

```plaintext
DEFINE FILE WF_RETAIL_LITE
  MYFIELD/A20 WITH COUNTRY_NAME='HELLO' | CTRLCHAR(CR) | CTRLCHAR(LF) | 'GOODBYE';
END
TABLE FILE WF_RETAIL_LITE
  SUM MYFIELD
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
  TYPE=REPORT,LINEBREAK='CRLF',$
ENDSTYLE
END
```

The output is shown in the following image.

```
MYFIELD
  HELLO
  GOODBYE
```

**EDIT2: Converting a Numeric, Date, or Date-Time Value to a Character String**

The EDIT2 function converts a numeric, date, or date-time value to a character string based on a specified format. The format must be valid for the data supplied. The output is returned as variable length alphanumeric.

**Syntax:**

How to Convert a Numeric, Date, or Date_Time Value to a Character String

```
EDIT2(in_value, 'format')
```

where:

- `in_value`
  Numeric, date, or date-time

  Is any numeric value or a date in either standard date or date-time format. If the date is given in standard date format, all of its time components are assumed to be zero.
'format'
    Is a numeric, date, or date-time format enclosed in single quotation marks (').

**Example:** Converting a Date to a Character String

The following request defines a date field as YYMD format and converts it to a character string (CharDate) in YYMtrD format.

```sql
DEFINE FILE WF_RETAIL_LITE
DATE1/YYMD = TIME_DATE_DAY_COMPONENT;
CharDate/A20 = EDIT2(DATE1,'YYMtrD');
END
TABLE FILE WF_RETAIL_LITE
SUM COGS_US
BY CharDate
WHERE TIME_MTHNAME   EQ 'FEB'
ON TABLE SET PAGE NOLEAD
END
```
The output is shown in the following image.

<table>
<thead>
<tr>
<th>CharDate</th>
<th>Cost of Goods</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012, February 1</td>
<td>$10,511.00</td>
</tr>
<tr>
<td>2012, February 2</td>
<td>$8,393.00</td>
</tr>
<tr>
<td>2012, February 3</td>
<td>$8,194.00</td>
</tr>
<tr>
<td>2012, February 4</td>
<td>$13,928.00</td>
</tr>
<tr>
<td>2012, February 5</td>
<td>$10,756.00</td>
</tr>
<tr>
<td>2012, February 6</td>
<td>$6,704.00</td>
</tr>
<tr>
<td>2012, February 7</td>
<td>$8,195.00</td>
</tr>
<tr>
<td>2012, February 8</td>
<td>$13,625.00</td>
</tr>
<tr>
<td>2012, February 9</td>
<td>$9,455.00</td>
</tr>
<tr>
<td>2012, February 10</td>
<td>$13,433.00</td>
</tr>
<tr>
<td>2012, February 11</td>
<td>$6,320.00</td>
</tr>
<tr>
<td>2012, February 12</td>
<td>$8,060.00</td>
</tr>
<tr>
<td>2012, February 13</td>
<td>$7,036.00</td>
</tr>
<tr>
<td>2012, February 14</td>
<td>$8,799.00</td>
</tr>
<tr>
<td>2012, February 15</td>
<td>$9,983.00</td>
</tr>
<tr>
<td>2012, February 16</td>
<td>$12,824.00</td>
</tr>
<tr>
<td>2012, February 17</td>
<td>$7,368.00</td>
</tr>
<tr>
<td>2012, February 18</td>
<td>$9,613.00</td>
</tr>
<tr>
<td>2012, February 19</td>
<td>$12,316.00</td>
</tr>
<tr>
<td>2012, February 20</td>
<td>$10,335.00</td>
</tr>
<tr>
<td>2012, February 21</td>
<td>$9,081.00</td>
</tr>
<tr>
<td>2012, February 22</td>
<td>$8,208.00</td>
</tr>
<tr>
<td>2012, February 23</td>
<td>$6,755.00</td>
</tr>
<tr>
<td>2012, February 24</td>
<td>$10,849.00</td>
</tr>
<tr>
<td>2012, February 25</td>
<td>$9,603.00</td>
</tr>
<tr>
<td>2012, February 26</td>
<td>$11,080.00</td>
</tr>
<tr>
<td>2012, February 27</td>
<td>$10,684.00</td>
</tr>
<tr>
<td>2012, February 28</td>
<td>$8,346.00</td>
</tr>
</tbody>
</table>
HEXTYPE: Returning the Hexadecimal View of an Input Value

The HEXTYPE function returns the hexadecimal view of an input value of any data type. The result is returned as variable length alphanumeric. The alphanumeric field to which the hexadecimal value is returned must be large enough to hold two characters for each input character. The value returned depends on the running operating environment.

Syntax:  

How to Returning the Hexadecimal View of an Input Value

HEXTYPE(in_value)

where:

in_value
    Is an alphanumeric or integer field, constant, or expression.

Example:  

Returning a Hexadecimal View

The following request returns a hexadecimal view of the country names and the sum of the days delayed.

DEFINE FILE WF_RETAIL_LITE
Days/I8 = DAYSDELAYED;
Country/A20 = COUNTRY_NAME;
HexCountry/A30 = HEXTYPE(Country);
END
TABLE FILE WF_RETAIL_LITE
SUM COUNTRY_NAME NOPRINT Country HexCountry Days
COMPUTE HexDays/A40 = HEXTYPE(Days);
BY COUNTRY_NAME NOPRINT
WHERE COUNTRY_NAME LT 'P'
ON TABLE SET PAGE NOPAGE
END
5. Reporting Language Enhancements

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Country</th>
<th>HexCountry</th>
<th>Days</th>
<th>HexDays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argentina</td>
<td>417267656E74696E61202020202020</td>
<td>84</td>
<td>000000054</td>
</tr>
<tr>
<td>Australia</td>
<td>4175737472616C6961202020202020</td>
<td>27</td>
<td>0000001B</td>
</tr>
<tr>
<td>Austria</td>
<td>417573747269612020202020202020</td>
<td>798</td>
<td>0000031E</td>
</tr>
<tr>
<td>Belgium</td>
<td>42656C6769756D2020202020202020</td>
<td>14</td>
<td>0000000E</td>
</tr>
<tr>
<td>Brazil</td>
<td>4272617A696C202020202020202020</td>
<td>204</td>
<td>000000CC</td>
</tr>
<tr>
<td>Canada</td>
<td>43616E616461202020202020202020</td>
<td>584</td>
<td>00000248</td>
</tr>
<tr>
<td>Chile</td>
<td>4368696C6520202020202020202020</td>
<td>45</td>
<td>0000002D</td>
</tr>
<tr>
<td>China</td>
<td>4368696E6120202020202020202020</td>
<td>1</td>
<td>00000001</td>
</tr>
<tr>
<td>Colombia</td>
<td>436F6C6F6D62696120202020202020</td>
<td>114</td>
<td>00000072</td>
</tr>
<tr>
<td>Denmark</td>
<td>44656E6D61726B2020202020202020</td>
<td>0</td>
<td>00000000</td>
</tr>
<tr>
<td>Egypt</td>
<td>4567797074202020202020202020</td>
<td>3</td>
<td>00000003</td>
</tr>
<tr>
<td>Finland</td>
<td>46696E6C616E642020202020202020</td>
<td>3</td>
<td>00000003</td>
</tr>
<tr>
<td>France</td>
<td>4672616E6365202020202020202020</td>
<td>49</td>
<td>00000031</td>
</tr>
<tr>
<td>Germany</td>
<td>4765726D616E792020202020202020</td>
<td>498</td>
<td>000001F2</td>
</tr>
<tr>
<td>Greece</td>
<td>477265656365202020202020202020</td>
<td>9</td>
<td>00000009</td>
</tr>
<tr>
<td>Hungary</td>
<td>48756E676172792020202020202020</td>
<td>7</td>
<td>00000007</td>
</tr>
<tr>
<td>India</td>
<td>496E64696120202020202020202020</td>
<td>23</td>
<td>00000017</td>
</tr>
<tr>
<td>Ireland</td>
<td>4972656C616E642020202020202020</td>
<td>7</td>
<td>00000007</td>
</tr>
<tr>
<td>Israel</td>
<td>49737261656C202020202020202020</td>
<td>2</td>
<td>00000002</td>
</tr>
<tr>
<td>Italy</td>
<td>4974616C7920202020202020202020</td>
<td>7</td>
<td>00000007</td>
</tr>
<tr>
<td>Japan</td>
<td>4A6170616E20202020202020202020</td>
<td>12</td>
<td>0000000C</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>4C7578656D626F7572672020202020</td>
<td>0</td>
<td>00000000</td>
</tr>
<tr>
<td>Malaysia</td>
<td>4D616C617973696120202020202020</td>
<td>20</td>
<td>00000014</td>
</tr>
<tr>
<td>Mexico</td>
<td>4D657869636F202020202020202020</td>
<td>170</td>
<td>000000AA</td>
</tr>
<tr>
<td>Netherlands</td>
<td>4E6574685726C616E64732020202020</td>
<td>8</td>
<td>00000008</td>
</tr>
<tr>
<td>Norway</td>
<td>4E6F72776179202020202020202020</td>
<td>0</td>
<td>00000000</td>
</tr>
</tbody>
</table>
PHONETIC: Returning a Phonetic Key for a String

PHONETIC calculates a phonetic key for a string, or a null value on failure. Phonetic keys are useful for grouping alphanumeric values, such as names, that may have spelling variations. This is done by generating an index number that will be the same for the variations of the same name based on pronunciation. One of two phonetic algorithms can be used for indexing, Metaphone and Soundex. Metaphone is the default algorithm, except on z/OS where the default is Soundex.

You can set the algorithm to use with the following command.

```
SET PHONETIC_ALGORITHM = {METAPHONE|SOUNDEX}
```

Most phonetic algorithms were developed for use with the English language. Therefore, applying the rules to words in other languages may not give a meaningful result.

Metaphone is suitable for use with most English words, not just names. Metaphone algorithms are the basis for many popular spell checkers.

**Note:** Metaphone is not optimized in generated SQL. Therefore, if you need to optimize the request for an SQL DBMS, the SOUNDEX setting should be used.

Soundex is a legacy phonetic algorithm for indexing names by sound, as pronounced in English.

**Reference:** Return a Phonetic Key

```
PHONETIC(string)
```

where:

```
string
  Alphanumeric
```

Is a string for which to create the key. A null value will be returned on failure.
**Example:** Generating a Phonetic Key

The following request changes the spelling of the last name for MARY SMITH to SMYTHE and generates a phonetic key for each last name.

```
DEFINE FILE EMPLOYEE
LAST_NAME2/A16 = IF LAST_NAME EQ 'SMITH' AND FIRST_NAME EQ 'MARY' THEN 'SMYTHE' ELSE LAST_NAME;
PKEY/A10 = PHONETIC(LAST_NAME2);
END
TABLE FILE EMPLOYEE
PRINT FIRST_NAME LAST_NAME2
BY PKEY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image. Note that the two spellings for SMITH are assigned the same index number.
TO_INTEGER: Converting a Character String to an Integer Value

TO_INTEGER converts a character string that contains a valid number consisting of digits and an optional decimal point to an integer value. If the value contains a decimal point, the value after the decimal point is truncated. If the value does not represent a valid number, zero (0) is returned.

Reference: Convert a Character String to an Integer

\[ \text{TO_INTEGER}(\text{string}) \]

where:

\[ \text{string} \]

Is a character string enclosed in single quotation marks or a character field that represents a number containing digits and an optional decimal point.

Example: Converting a Character String to an Integer Value

The following request converts character strings to integers. Digits following the decimal point are truncated.

```
DEFINE FILE WF_RETAIL_LITE
INT1/I8 = TO_INTEGER('56.78');
INT2/I8 = TO_INTEGER('.5678');
INT3/I8 = TO_INTEGER('5678');
END
TABLE FILE WF_RETAIL_LITE
PRINT INT1 INT2 INT3
BY BUSINESS_REGION AS Region
WHERE READLIMIT EQ 1
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Region</th>
<th>INT1</th>
<th>INT2</th>
<th>INT3</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>56</td>
<td>0</td>
<td>5678</td>
</tr>
</tbody>
</table>
TO_NUMBER: Converting a Character String to a Numeric Value

TO_NUMBER converts a character string that contains a valid number consisting of digits and an optional decimal point to the numeric format most appropriate to the context. If the value does not represent a valid number, zero (0) is returned.

Reference: Convert a Character String to a Number

TO_NUMBER(string)

where:

string

Is a character string enclosed in single quotation marks or a character field that represents a number containing digits and an optional decimal point. This string will be converted to a double-precision floating point number.

Example: Converting a Character String to a Number

The following request converts character strings to double-precision floating point numbers.

```
DEFINE FILE WF_RETAIL_LITE
NUM1/D12.1 = TO_NUMBER('56.78');
NUM2/D12.2 = TO_NUMBER('0.5678');
END
TABLE FILE WF_RETAIL_LITE
PRINT NUM1 NUM2
BY BUSINESS_REGION AS Region
WHERE READLIMIT EQ 1
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF, $
ENDSTYLE
END
```

The output is shown in the following image.

```
<table>
<thead>
<tr>
<th>Region</th>
<th>NUM1</th>
<th>NUM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMEA</td>
<td>56.8</td>
<td>.57</td>
</tr>
</tbody>
</table>
```

Simplified Date Functions

The new simplified date functions DT_CURRENT_DATE, DT_CURRENT_DATETIME, and DT_CURRENT_TIME have been added.
DT_CURRENT_DATE: Returning the Current Date

The DT_CURRENT_DATE function returns the current date-time provided by the running operating environment in date-time format. The time portion of the datetime is set to zero.

Syntax: How to Return the Current Date

```
DT_CURRENT_DATE()
```

Example: Returning the Current Date

The following request returns the current date.

```
DEFINE FILE WF_RETAIL_LITE
CURRDATE/YYMD WITH COUNTRY_NAME = DT_CURRENT_DATE();
END
TABLE FILE WF_RETAIL_LITE
SUM CURRDATE
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

```
CURRDATE
2016/09/08
```

DT_CURRENT_DATETIME: Returning the Current Date and Time

DT_CURRENT_DATETIME returns the current date and time provided by the running operating environment in date-time format, with a specified time precision.

Syntax: How to Return the Current Date and Time

```
DT_CURRENT_DATETIME(component)
```

where:

```
component
```

Is one of the following time precisions.

- SECOND.
- MILLISECOND.
- MICROSECOND
Note: The field to which the value is returned must have a format that supports the time precision requested.

Example: Returning the Current Date and Time
The following request returns the current date and time, with the time specified in microseconds.

```
DEFINE FILE WF_RETAIL_LITE
CURRDATE/HYYMDm WITH COUNTRY_NAME = DT_CURRENT_DATETIME(MICROSECOND);
END
TABLE FILE WF_RETAIL_LITE
SUM CURRDATE
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

```
CURRDATE
2016/09/08 17:10:31.605718
```

DT_CURRENT_TIME: Returning the Current Time
The DT_CURRENT_TIME function returns the current time provided by the running operating environment in date-time format, with a specified time precision. The date portion of the returned date-time value is set to zero.

Syntax: How to Return the Current Time
```
DT_CURRENT_TIME(component)
```
where:

- **component**
  Is one of the following time precisions.
  - SECOND.
  - MILLISECOND.
  - MICROSECOND

Note: The field to which the value is returned must have a format that supports the time precision requested.
Example: Returning the Current Time

The following request returns the current time, with the time precision set to milliseconds.

```plaintext
DEFINE FILE WF_RETAIL_LITE
CURRTIME/HHISS WITH COUNTRY_NAME = DT_CURRENT_TIME(MILLISECOND);
END
TABLE FILE WF_RETAIL_LITE
SUM CURRTIME
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

![Current Time](CURRTIME
17:23:13.098)

Simplified System Functions

Simplified system functions `ENCRYPT`, `GETENV`, and `PUTENV` have been added.

**ENCRYPT: Encrypting a Password**

The `ENCRYPT` function encrypts an alphanumeric input value using the encryption algorithm configured in the server. The result is returned as variable length alphanumeric.

**Syntax:**

How to Encrypt a Password

```plaintext
ENCRYPT(password)
```

where:

- **password**
  - Fixed length alphanumeric
  - Is the value to be encrypted.

**Example:** Encrypting a Password

The following request encrypts the value `guestpassword` using the encryption algorithm configured in the server.

```plaintext
-SET &P1 = ENCRYPT('guestpassword');
-TYPE &P1
```

The returned encrypted value is (AES)963AFA754E1763ABE697E8C5E764115E.
GETENV: Retrieving the Value of an Environment Variable

The GETENV function takes the name of an environment variable and returns its value as a variable length alphanumeric value.

Syntax: How to Retrieve the Value of an Environment Variable

```
GETENV(var_name)
```

where:

- **var_name**
  - fixed length alphanumeric
  - Is the name of the environment variable whose value is being retrieved.

Example: Retrieving the Value of an Environment Variable

The following request retrieves the value of the server variable EDAEXTSEC.

```
-SET &E1 = GETENV('EDAEXTSEC');
-TYPE &E1
```

GETENV retrieves the value of the server variable EDAEXTSEC.

```
GETENV('EDAEXTSEC')
```

The value returned is ON if the server was started with security on or OFF if the server was started with security off.

PUTENV: Assigning a Value to an Environment Variable

The PUTENV function assigns a value to an environment variable. The function returns an integer return code whose value is 1 (one) if the assignment is not successful or 0 (zero) if it is successful.

Syntax: How to Assign a Value to an Environment Variable

```
PUTENV(var_name, var_value)
```

where:

- **var_name**
  - Fixed length alphanumeric
  - Is the name of the environment variable to be set.
**Example:** Assigning a Value to the UNIX PS1 Variable

The following request assigns the value `FOCUS/Shell:` to the UNIX PS1 variable.

```
-SET &P1 = PUTENV('PS1','FOCUS/Shell:');
```

PUTENV assigns the value `FOCUS/Shell:` to the UNIX PS1 variable.

```
PUTENV('PS1','FOCUS/Shell:')
```

This causes UNIX to display the following prompt when the user issues the UNIX shell command `SH`:

```
FOCUS/Shell:
```

The following request creates a variable named `xxxx` and sets it to the value `this is a test`. It then retrieves the value using `GETENV`.

```
-SET &XXXX=PUTENV(xxxx,'this is a test');
-SET &YYYY=GETENV(xxxx);
-TYPE  Return Code: &XXXX,  Variable value: &YYYY
```

The output is:

```
Return Code: 0,  Variable value: this is a test
```

**Simplified Geography Functions**

The simplified geography functions perform location-based calculations and retrieve geocoded points for various types of location data. They are used by the WebFOCUS location intelligence products that produce maps and charts. Some of the geography functions use GIS services and require valid credentials for accessing Esri ArcGIS proprietary data.
Sample Geography Files

Some of the examples for the geography functions use geography sample files. One file, esri-citibike.csv has station names, latitudes, and longitudes and trip start times and end times. The other file, esri-geo10036.ftm has geometry data. To run the examples that use these files, create an application named esri, and place the following files into the application folder.

esri-citibike.mas

FILENAME=ESRI-CITIBIKE, SUFFIX=DFIX, 
DATASET=esri/esri-citibike.csv, 
SEGMENT=CITIBIKE_TRIPDATA, SEGTYPE=S0, 
  FIELDNAME=TRIPDURATION, ALIAS=tripduration, USAGE=I7, ACTUAL=A5V, 
    TITLE='tripduration', 
  FIELDNAME=STARTTIME, ALIAS=starttime, USAGE=HMDYYS, ACTUAL=A18, 
    TITLE='starttime', 
  FIELDNAME=STOPTIME, ALIAS=stoptime, USAGE=HMDYYS, ACTUAL=A18, 
    TITLE='stoptime', 
  FIELDNAME=START_STATION_ID, ALIAS='start station id', USAGE=I6, 
    ACTUAL=A4V, 
    TITLE='start station id', 
  FIELDNAME=START_STATION_NAME, ALIAS='start station name', USAGE=A79V, 
    ACTUAL=A79BV, 
    TITLE='start station name', 
  FIELDNAME=START_STATION_LATITUDE, ALIAS='start station latitude', 
    USAGE=P20.15, ACTUAL=A18V, 
    TITLE='start station latitude', 
    GEOGRAPHIC_ROLE=LATITUDE, 
  FIELDNAME=START_STATION_LONGITUDE, ALIAS='start station longitude', 
    USAGE=P20.14, ACTUAL=A18V, 
    TITLE='start station longitude', 
    GEOGRAPHIC_ROLE=LONGITUDE, 
  FIELDNAME=END_STATION_ID, ALIAS='end station id', USAGE=I6, 
    ACTUAL=A4V, TITLE='end station id', 

5. Reporting Language Enhancements
Note: Each complete record must be on a single line. Therefore, you must remove any line breaks that may have been inserted due to the page width in this document.

1094, 11/1/2015 0:00, 11/1/2015 0:18, 537, Lexington Ave & E 24 St, 40.74025878, -73.98409214, 531, Forsyth St & Broome St, 40.71893904, -73.99266288, 23959, Subscriber, 1980, 1

520, 11/1/2015 0:00, 11/1/2015 0:08, 536, 1 Ave & E 30 St, 40.74144387, -73.97536082, 498, Broadway & W 32 St, 40.74854862, -73.98808416, 22251, Subscriber, 1988, 1

753, 11/1/2015 0:00, 11/1/2015 0:12, 229, Great Jones St, 40.72743423, -73.99379025, 328, Watts St & Greenwich St, 40.72405549, -74.00965965, 15869, Subscriber, 1981, 1

353, 11/1/2015 0:00, 11/1/2015 0:06, 285, Broadway & E 14 St, 40.73454567, -73.99074142, 151, Cleveland Pl & Spring St, 40.72210379, -73.99724901, 21645, Subscriber, 1987, 1
5. Reporting Language Enhancements

1285,11/1/2015 0:00,11/1/2015 0:21,268,Howard St & Centre St, 40.71910537,-73.99973337,476,E 31 St & 3 Ave,40.74394314,-73.97966069,14788,Customer,,0

477,11/1/2015 0:00,11/1/2015 0:08,379,W 31 St & 7 Ave,40.749156,-73.9916,546,E 30 St & Park Ave S,40.74444921,-73.98303529,21128,Subscriber,1962,2

362,11/1/2015 0:00,11/1/2015 0:06,407,Henry St & Poplar St, 40.700469,-73.991454,310,State St & Smith St,40.68926942,-73.983012867,21016,Subscriber,1978,1

2316,11/1/2015 0:00,11/1/2015 0:39,147,Greenwich St & Warren St, 40.71542197,-74.01121978,441,E 52 St & 2 Ave,40.756014,-73.967416,24117,Subscriber,1988,2

627,11/1/2015 0:00,11/1/2015 0:11,521,8 Ave & W 31 St, 40.75096735,-73.99444208,285,Broadway & E 14 St,40.73454567,-73.99074124,17048,Subscriber,1986,2

1484,11/1/2015 0:01,11/1/2015 0:26,281,Grand Army Plaza & Central Park S, 40.7643971,-73.97371465,367,E 53 St & Lexington Ave,40.75828065,-73.97069431,16779,Customer,,0

284,11/1/2015 0:01,11/1/2015 0:06,247,Perry St & Bleecker St, 40.73535398,-74.00483091,453,W 22 St & 8 Ave,40.74475148,-73.99915362,17272,Subscriber,1976,2

886,11/1/2015 0:01,11/1/2015 0:16,492,W 33 St & 7 Ave,40.75019995,-73.99093085,377,6 Ave & Canal St,40.72243797,-74.00566443,23019,Subscriber,1982,1

1379,11/1/2015 0:01,11/1/2015 0:24,512,W 29 St & 9 Ave,40.7500727,-73.99839279,445,E 10 St & Avenue A,40.72740794,-73.98142006,23843,Subscriber,1962,2

179,11/1/2015 0:01,11/1/2015 0:04,319,Fulton St & Broadway, 40.711066,-74.009447,264,Maiden Ln & Pearl St,40.70706456,-74.00731853,22538,Subscriber,1981,1

309,11/1/2015 0:01,11/1/2015 0:07,160,E 37 St & Lexington Ave, 40.748238,-73.98311,362,Broadway & W 37 St,40.75172632,-73.98753523,22042,Subscriber,1988,1

616,11/1/2015 0:02,11/1/2015 0:12,479,9 Ave & W 45 St,40.76019252,-73.9912551,440,E 45 St & 3 Ave,40.75255434,-73.97282625,22699,Subscriber,1982,1

852,11/1/2015 0:02,11/1/2015 0:16,346,Bank St & Hudson St, 40.73652889,-74.00618026,375,Mercer St & Bleecker St,40.72679454,-73.99659094,21011,Subscriber,1991,1

1854,11/1/2015 0:02,11/1/2015 0:33,409,DeKalb Ave & Skillman St, 40.6906495,-73.95643107,3103,N 11 St & Wythe Ave,40.72153267,-73.95782357,22011,Subscriber,1992,1
esri-geo10036.mas

FILENAME=ESRI-GEO10036, SUFFIX=FIX
DATASET=esri/esri-geo10036.ftm (LRECL 80 RECFM V, IOTYPE=STREAM, $
SEGMENT=ESRIGEO, SEGTYPE=S0, $
FIELDNAME=GEOMETRY, ALIAS=GEOMETRY, USAGE=TX80L, ACTUAL=TX80,
MISSING=ON, $

esri-geo10036.ftm

{"rings":
  [[-73.9803889998524,40.7541490002762],
   [-73.9808779999197,40.7534830001404],
   [-73.9814419998484,40.7537140000011],
   [-73.9824040001445,40.7541199998382],
   [-73.9824610000075,40.7541344001978],
   [-73.9825620002361,40.7541850001377],
   [-73.9832877000673,40.7544888999428],
   [-73.9833499997027,40.7545150000673],
   [-73.983644399],
   [969,40.7546397998869],
   [-73.9836849998628,40.7546570003204],
   [-73.9841276003885,40.7548161002829],
   [-73.984399700086,40.7544544999752],
   [-73.9846140004357,40.7541650001147],
   [-73.9848719997914],
   [-73.9866590003126,40.7550369998577],
   [-73.9874449996869,40.755372000178],
   [-73.9902640001834,40.756570999552],
   [-73.9914340001789,40.7570449998269],
   [-73.9918260002697,40.7572149995726],
   [-73.992429001982,40.7574769999636],
   [-73.9927679999634],
   [-73.9930690000343,40.7576240004473],
   [-73.9930690000343,40.7578009996165],
   [-73.9931059999419,40.7577600004237],
   [-73.9932120003335,40.7576200004012],
   [-73.9933250001486,40.7576770001934],
   [-73.9935390001247,40.7577669]
5. Reporting Language Enhancements

998472], [-73.9937259999755, 40.7578459998931],
[-73.9939599997542, 40.757937999639],
[-73.9940989998689, 40.7579839999617], [-73.9941529996611, 40.75799599996157],
[-73.99422001452, 40.7580159996387], [-73.9943040003293, 40.7580300002843],
[-73.9945660002591, 40.75803699997078],
[-73.99446499986, 40.758099999893],
[-73.994365000444, 40.7580999999383],
[-73.9944169999092, 40.7582882001],
[-73.9943810002829, 40.7583400001909],
[-73.9953849998179, 40.7587409997973], [-73.9959560000693, 40.7589690004191], [-73.9960649996999, 40.7590149998424],
[-73.996830000888, 40.7593419996336], [-73.996975000296, 40.7593809996335],
[-73.9973149999383], [-73.9977009997759, 40.7597030000935],
[-73.998039999946, 40.7598479995856], [-73.99834000014, 40.759970999818],
[-73.9987769997587, 40.7601570003453], [-73.9990089996656, 40.7602540003219],
[-74.000015059997021, 40.761292999672]

2], [-74.0016340002089, 40.7613299995799],
[-74.0015530001401, 40.7614539999022], [-74.0014580001865, 40.7615479997405], [-74.0013640003483, 40.7616560002242],
[-74.0000770003229, 40.7617199999784], [-74.0011890003721, 40.7618369995779],
[-74.0010579997, 40.7619609999903], [-74.0009659999808, 40.7620389999],
[-74.0008649999198, 40.76213299995799],
[-74.0007639995669, 40.76226100],
[-74.0006319999175, 40.762375000162],
[-74.0003759997525, 40.7624849997829], [-74.0002840000066, 40.762551000186], [-73.9998659996161, 40.762850999574],
[-73.9998769996999, 40.7628779999198], [-73.9995749996864, 40.7630590001727],
[-73.999312000], 40.7632720001028], [-73.9991639996189, 40.763398999642],
[-73.999841000127, 40.7636250001936], [-73.9987589998279, 40.7638580001466],
[-73.9986331999622, 40.7640277004181], [-73.9986084002574, 40.7640632002565],
[-73.9984199996445, 40.7642340003989], [-73.9983469997142, 40.764419999831],
[-73.998171999738, 40.7646699996823], [-73.9980319995771, 40.7648580003964], [-73.9979881998955, 40.7649204996813],
[-73.997977017001]
**GIS_DISTANCE: Calculating the Distance Between Geometry Points**

The GIS_DISTANCE function uses a GIS service to calculate the distance between two geometry points.

**Syntax:**

GIS_DISTANCE(geo_point1, geo_point2)

where:

*geo_point1, geo_point2*

Fixed length alphanumeric, large enough to hold the JSON describing the point (for example, A200).

Are the geometry points for which you want to calculate the distance.

**Note:** You can generate a geometry point using the GIS_POINT function.
Example: Calculating the Distance Between Two Geometry Points

The following uses a citibike .csv file that contains station names, latitudes, and longitudes and trip start times and end times. It uses the GIS_POINT function to define geometry points for start stations and end stations. It then uses GIS_DISTANCE to calculate the distance between them.

```
DEFINE FILE esri/esri-citibike
STARTPOINT/A200 = GIS_POINT('4326', START_STATION_LONGITUDE, START_STATION_LATITUDE);
ENDPOINT/A200 = GIS_POINT('4326', END_STATION_LONGITUDE, END_STATION_LATITUDE);
Distance/P10.2 = GIS_DISTANCE(ENDPOINT, STARTPOINT);
END

TABLE FILE esri/esri-citibike
PRINT END_STATION_NAME AS End Distance
BY START_STATION_NAME AS Start
ON TABLE SET PAGE NOLEAD
END
```
GIS_DRIVE_ROUTE: Calculating the Driving Directions Between Geometry Points

The GIS_DRIVE_ROUTE function uses a GIS service to calculate the driving route between two geometry points.

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Ave &amp; E 30 St</td>
<td>Broadway &amp; W 32 St</td>
<td>.83</td>
</tr>
<tr>
<td>8 Ave &amp; W 31 St</td>
<td>Broadway &amp; E 14 St</td>
<td>1.15</td>
</tr>
<tr>
<td></td>
<td>E 20 St &amp; 2 Ave</td>
<td>1.23</td>
</tr>
<tr>
<td>9 Ave &amp; W 45 St</td>
<td>E 45 St &amp; 3 Ave</td>
<td>1.10</td>
</tr>
<tr>
<td>Bank St &amp; Hudson St</td>
<td>Mercer St &amp; Bleecker St</td>
<td>.83</td>
</tr>
<tr>
<td>Broadway &amp; E 14 St</td>
<td>Cleveland Pl &amp; Spring St</td>
<td>.92</td>
</tr>
<tr>
<td>DeKalb Ave &amp; Skillman St</td>
<td>N 11 St &amp; Wythe Ave</td>
<td>2.13</td>
</tr>
<tr>
<td>E 37 St &amp; Lexington Ave</td>
<td>Broadway &amp; W 37 St</td>
<td>.54</td>
</tr>
<tr>
<td>Fulton St &amp; Broadway</td>
<td>Maiden Ln &amp; Pearl St</td>
<td>.30</td>
</tr>
<tr>
<td>Grand Army Plaza &amp; Central Park S</td>
<td>E 53 St &amp; Lexington Ave</td>
<td>.45</td>
</tr>
<tr>
<td>Great Jones St</td>
<td>Watts St &amp; Greenwich St</td>
<td>.87</td>
</tr>
<tr>
<td>Greenwich St &amp; Warren St</td>
<td>E 52 St &amp; 2 Ave</td>
<td>3.62</td>
</tr>
<tr>
<td>Henry St &amp; Poplar St</td>
<td>State St &amp; Smith St</td>
<td>.78</td>
</tr>
<tr>
<td>Howard St &amp; Centre St</td>
<td>E 31 St &amp; 3 Ave</td>
<td>2.01</td>
</tr>
<tr>
<td>Lexington Ave &amp; E 24 St</td>
<td>Forsyth St &amp; Broome St</td>
<td>1.54</td>
</tr>
<tr>
<td>Perry St &amp; Bleecker St</td>
<td>W 22 St &amp; 8 Ave</td>
<td>.71</td>
</tr>
<tr>
<td>S 5 Pl &amp; S 4 St</td>
<td>E 5 St &amp; Avenue C</td>
<td>1.32</td>
</tr>
<tr>
<td>W 29 St &amp; 9 Ave</td>
<td>E 10 St &amp; Avenue A</td>
<td>1.80</td>
</tr>
<tr>
<td>W 31 St &amp; 7 Ave</td>
<td>E 30 St &amp; Park Ave S</td>
<td>.55</td>
</tr>
<tr>
<td>W 33 St &amp; 7 Ave</td>
<td>6 Ave &amp; Canal St</td>
<td>2.07</td>
</tr>
</tbody>
</table>
Syntax: How to Calculate the Drive Route Between Geometry Points

GIS_DRIVE_ROUTE(geo_start_point, geo_end_point)

where:

geo_start_point, geo_point2
Fixed length alphanumeric, large enough to hold the JSON describing the point (for example, A200).

Is the starting point for which you want to calculate the drive route.

Note: You can generate a geometry point using the GIS_POINT function.

geo_end_point, geo_point2
Fixed length alphanumeric, large enough to hold the JSON describing the point (for example, A200).

Is the ending point for which you want to calculate the drive route.

Note: You can generate a geometry point using the GIS_POINT function.

The format of the field to which the drive route will be returned is TX.

Example: Calculating the Drive Route Between Two Geometry Points

The following uses a citibike .csv file that contains station names, latitudes, and longitudes and trip start times and end times. It uses the GIS_POINT function to define geometry points for start stations and end stations. It then uses GIS_DRIVE_ROUTE to calculate the route to get from the end point to the start point.

DEFINE FILE esri/esri-citibike
STARTPOINT/A200 = GIS_POINT('4326', START_STATION_LONGITUDE, START_STATION_LATITUDE);
ENDPOINT/A200 = GIS_POINT('4326', END_STATION_LONGITUDE, END_STATION_LATITUDE);
Route/TX140 (GEOGRAPHIC_ROLE=GEOMETRY_LINE) =
    GIS_DRIVE_ROUTE(ENDPOINT, STARTPOINT);
END

TABLE FILE esri/esri-citibike
PRINT START_STATION_NAME AS Start END STATION NAME AS End Route
WHERE START_STATION_ID EQ 147
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, SIZE-11,$
ENDSTYLE
END
Example:

Charting a Driving Route Between Two Geometry Points

The following request uses GIS_DRIVE_ROUTE to generate a driving route between a station start point and end point and charts the route on an Esri map.

DEFINE FILE esri-citibike
STARTPOINT/A200 = GIS_POINT('4326', START_STATION_LONGITUDE, START_STATION_LATITUDE);
ENDPOINT/A200 = GIS_POINT('4326', END_STATION_LONGITUDE, END_STATION_LATITUDE);
Route/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_LINE) = GIS_DRIVE_ROUTE(ENDPOINT, STARTPOINT);
END
The output is shown in the following image.

GIS_POINT: Building a Geometry Point

Given a WKID (Well-Known ID) spatial reference, longitude, and latitude, the GIS_POINT function builds a JSON point defining a Geometry object with the provided WKID, longitude, and latitude. The function is optimized for those SQL engines that can build a JSON geometry object.

The field to which the point is returned should have fixed length alphanumeric format, large enough to hold the JSON describing the point (for example, A200).

Syntax: How to Build a Geometry Point

GIS_POINT(wkid, longitude, latitude)

where:

wkid
Fixed length alphanumeric

Is a spatial reference code (WKID). WKID is an abbreviation for Well Known ID, which identifies a projected or geographic coordinate system.

longitude
D20.8

Is the longitude for the point.
**Example:**  
**Building a Geometry Point**

The following request uses the spatial reference code 4326 (decimal degrees) and state capital longitudes and latitudes to build a geometry point.

```plaintext
DEFINE FILE WF_RETAIL_LITE
GPOINT/A200 = GIS_POINT('4326', STATE_PROV_CAPITAL_LONGITUDE, STATE_PROV_CAPITAL_LATITUDE);
END
TABLE FILE WF_RETAIL_LITE
SUM FST.STATE_PROV_CAPITAL_LONGITUDE AS Longitude
FST.STATE_PROV_CAPITAL_LATITUDE AS Latitude
FST.GPOINT AS Point
BY STATE_PROV_CAPITAL_NAME AS Capital
WHERE COUNTRY_NAME EQ 'United States'
WHERE STATE_PROV_CAPITAL_NAME LT 'C'
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Capital</th>
<th>Longitude</th>
<th>Latitude</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albany</td>
<td>-73.76000000</td>
<td>42.60000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-73.76000000, 'y':42.60000000})</td>
</tr>
<tr>
<td>Annapolis</td>
<td>-76.49000000</td>
<td>38.95000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-76.49000000, 'y':38.95000000})</td>
</tr>
<tr>
<td>Atlanta</td>
<td>-84.27000000</td>
<td>33.94000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-84.27000000, 'y':33.94000000})</td>
</tr>
<tr>
<td>Augusta</td>
<td>-69.77000000</td>
<td>44.32000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-69.77000000, 'y':44.32000000})</td>
</tr>
<tr>
<td>Austin</td>
<td>-97.75000000</td>
<td>30.40000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-97.75000000, 'y':30.40000000})</td>
</tr>
<tr>
<td>Baton Rouge</td>
<td>-91.17000000</td>
<td>30.38000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-91.17000000, 'y':30.38000000})</td>
</tr>
<tr>
<td>Bismarck</td>
<td>-100.77000000</td>
<td>46.82000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-100.77000000, 'y':46.82000000})</td>
</tr>
<tr>
<td>Boise</td>
<td>-116.16000000</td>
<td>43.60000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-116.16000000, 'y':43.60000000})</td>
</tr>
<tr>
<td>Boston</td>
<td>-71.10000000</td>
<td>42.35000000</td>
<td>('spatialReference': 'wkid' 4326, 'geometryType': 'esriGeometryPoint', 'geometry': {'x':-71.10000000, 'y':42.35000000})</td>
</tr>
</tbody>
</table>

**Example:**  
**Charting Geometry Points**

The following request generates geometry points using GIS_POINT charts them on an Esri map.

```plaintext
DEFINE FILE WF_RETAIL
GPOINT/A200 = GIS_POINT('4326', STATE_PROV_CAPITAL_LONGITUDE, STATE_PROV_CAPITAL_LATITUDE);
END
```
GRAPH FILE WF_RETAIL
PRINT
STATE_PROV_NAME
WHERE STATE_PROV_CAPITAL_LONGITUDE NE MISSING
ON TABLE PCHOLD FORMAT JSCHART
ON TABLE SET LOOKGRAPH BUBBLEMAP
ON TABLE SET EMBEDHEADING ON
ON TABLE SET AUTOFIT ON
ON TABLE SET STYLE *
    TYPE=REPORT, TITLETEXT='Map', PAGESIZE=E, CHART-LOOK=com.esri.map, $
    TYPE=DATA, COLUMN=N1,
    BUCKET=tooltip, $
*GRAPH_JS_FINAL
    "bubbleMarker": {"maxSize": "10%"},
    "legend": {"visible": true},
    "extensions" : { "com.esri.map" : 
        { "scalebar" : 
            { "scalebarUnit": "dual",
            "attachTo": "bottom-left"
        },
        "baseMapInfo": {
            "drawBasemapControl" : false,
            "showArcGISBasemaps" : false,
            "customBaseMaps" : [
                {"ibiBaseLayer" : "gray"}
            ]
        }
    },
    "overlayLayers":
    [{
        "ibiDataLayer": {"map-geometry" : {"map_by_field" : "GPOINT"}},
        "title" : "Report"}
    ],
    "introAnimation": "{\"enabled\":false}"
}
*END
ENDSTYLE
HEADING
"Chart Geometry Points"
END
GIS_GEOCODE_ADDR: Geocoding a Complete Address

GIS_GEOCODE_ADDR uses a GIS geocoding service to obtain the geometry point for a complete address.

**Syntax:**

How to Geocode a Complete Address

```plaintext
GIS_GEOCODE_ADDR(address[, country])
```

where:

- **address**
  - Fixed length alphanumeric
  - Is the complete address to be geocoded.

- **country**
  - Fixed length alphanumeric
  - Is a country name, which is optional if the country is the United States.
Example: Geocoding a Complete Address

The following request creates a complete address by concatenating the street address, city, state, and zip code. It then uses GIS_GEOCODE_ADDR to create a GIS point for the address.

```
DEFINE FILE WF_RETAIL_LITE
GADDRESS/A200 = ADDRESS_LINE_1 || ' ' | CITY_NAME || ' ' | STATE_PROV_NAME || ' ' | POSTAL_CODE;
GEOCODE1/A200 = GIS_GEOCODE_ADDR(GADDRESS);
END

TABLE FILE WF_RETAIL_LITE
PRINT ADDRESS_LINE_1 AS Address GEOCODE1
BY POSTAL_CODE AS Zip
WHERE CITY_NAME EQ 'New York'
WHERE POSTAL_CODE FROM '10013' TO '10020'
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Zip</th>
<th>Address</th>
<th>GEOCODE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10013</td>
<td>125 Worth St</td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7400269, &quot;y&quot;: 71543}}</td>
</tr>
<tr>
<td>10016</td>
<td>123 E 35Th St</td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397911, &quot;y&quot;: 74705}}</td>
</tr>
<tr>
<td>10017</td>
<td>2 United Nations Pl</td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397115, &quot;y&quot;: 75111}}</td>
</tr>
<tr>
<td>405 E 42Nd St</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7396956, &quot;y&quot;: 74867}}</td>
</tr>
<tr>
<td>406 E 42Nd St</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7396956, &quot;y&quot;: 74867}}</td>
</tr>
<tr>
<td>219 E 42Nd St</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397333, &quot;y&quot;: 75030}}</td>
</tr>
<tr>
<td>330 Madison Ave</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397906, &quot;y&quot;: 75316}}</td>
</tr>
<tr>
<td>10018</td>
<td>119 W 40Th St F10</td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7398599, &quot;y&quot;: 75398}}</td>
</tr>
<tr>
<td>11 West 40Th Street</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7398235, &quot;y&quot;: 75245}}</td>
</tr>
<tr>
<td>10019</td>
<td>51 West 32Nd Street</td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397776, &quot;y&quot;: 76044}}</td>
</tr>
<tr>
<td>1301 Ave Of The Americas</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397945, &quot;y&quot;: 76125}}</td>
</tr>
<tr>
<td>1345 Avenue Of The Americas</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7397843, &quot;y&quot;: 76254}}</td>
</tr>
<tr>
<td>1351 Ave Of The Americas</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73976340, &quot;y&quot;: 76077}}</td>
</tr>
<tr>
<td>1357 Ave Of The Americas</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7398139, &quot;y&quot;: 75874}}</td>
</tr>
<tr>
<td>1271 Avenue Of The Americas</td>
<td></td>
<td>{&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 7398018, &quot;y&quot;: 76023}}</td>
</tr>
</tbody>
</table>

GIS_GEOCODE_ADDR_CITY: Geocoding an Address Line, City, and State

GIS_GEOCODE_ADDR_CITY uses a GIS geocoding service to obtain the geometry point for an address line, city, state, and optional country. The returned value is a fixed length alphanumeric format, large enough to hold the JSON describing the geographic location (for example, A200).

Syntax: How to Geocode an Address Line, City, and State

```
GIS_GEOCODE_ADDR_CITY( street_addr, city , state [, country])
```
where:

- **street_addr**
  Fixed length alphanumeric
  Is the street address to be geocoded.

- **city**
  Fixed length alphanumeric
  Is the city name associated with the street address.

- **state**
  Fixed length alphanumeric
  Is the state name associated with the street address.

- **country**
  fixed length alphanumeric
  Is a country name, which is optional if the country is the United States.

**Example:** **Geocoding a Street Address, City, and State**

The following request geocodes a street address using GIS_GEOCODE_ADDR_CITY.

```
DEFINE FILE WF_RETAIL_LITE
GEOCODE1/A200 = GIS_GEOCODE_ADDR_CITY(ADDRESS_LINE_1, CITY_NAME ,
STATE_PROV_NAME);
END
TABLE FILE WF_RETAIL_LITE
PRINT ADDRESS_LINE_1 AS Address GEOCODE1
BY POSTAL_CODE AS Zip
WHERE CITY_NAME EQ 'New York'
WHERE POSTAL_CODE FROM '10013' TO '10020'
ON TABLE SET PAGE NOPAGE
END
```
GIS_GEOCODE_ADDR_POSTAL: Geocoding an Address Line and Postal Code

GIS_GEOCODE_ADDR_POSTAL uses a GIS geocoding service to obtain the geometry point for an address line, postal code and optional country. The returned value is a fixed length alphanumeric format, large enough to hold the JSON describing the geographic location (for example, A200).

**Syntax:** How to Geocode an Address Line and Postal Code

\[
\text{GIS\_GEOCODE\_ADDR\_POSTAL}(\text{street\_addr}, \text{postal\_code} [, \text{country}])
\]

where:

*street_addr*  
fixed length alphanumeric  
Is the street address to be geocoded.  

*postal_code*  
fixed length alphanumeric  
Is the postal code associated with the street address.  

*country*  
fixed length alphanumeric  
Is a country name, which is optional if the country is the United States.
**Example:** Geocoding a Street Address and Postal Code

The following request geocodes a street address using GIS_GEOCODE_ADDR_POSTAL.

```sql
DEFINE FILE WF RETAIL LITE
GEOCODE1/A200 = GIS_GEOCODE_ADDR_POSTAL(ADDRESS_LINE_1, POSTAL_CODE);
END
TABLE FILE WF RETAIL LITE
PRINT ADDRESS_LINE_1 AS Address GEOCODE1
BY POSTAL_CODE AS Zip
WHERE CITY_NAME EQ 'New York'
WHERE POSTAL_CODE FROM '10013' TO '10020'
ON TABLE SET PAGE NOPAGE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Zip</th>
<th>Address</th>
<th>GEOCODE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10013</td>
<td>125 Worth St</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 74.00269, &quot;y&quot;: 40.71543}]</td>
</tr>
<tr>
<td>10016</td>
<td>139 E 35Th St</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97911, &quot;y&quot;: 40.74705}]</td>
</tr>
<tr>
<td>10037</td>
<td>2 United Nations Plk</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97115, &quot;y&quot;: 40.75111}]</td>
</tr>
<tr>
<td>10040</td>
<td>405 E 42Nd St</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.96956, &quot;y&quot;: 40.74867}]</td>
</tr>
<tr>
<td>10040</td>
<td>405 E 42Nd St</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.96956, &quot;y&quot;: 40.74867}]</td>
</tr>
<tr>
<td>219 E 42Nd St</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97333, &quot;y&quot;: 40.75030}]</td>
<td></td>
</tr>
<tr>
<td>530 Madison Ave</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97906, &quot;y&quot;: 40.75316}]</td>
<td></td>
</tr>
<tr>
<td>10018</td>
<td>119 W 40Th St Fl 10</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.98599, &quot;y&quot;: 40.73398}]</td>
</tr>
<tr>
<td>11 West 40Th Street</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.98225, &quot;y&quot;: 40.72245}]</td>
<td></td>
</tr>
<tr>
<td>10019</td>
<td>11 West 52Nd Street</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97776, &quot;y&quot;: 40.76044}]</td>
</tr>
<tr>
<td>1301 Ave Of The Americas</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97945, &quot;y&quot;: 40.76125}]</td>
<td></td>
</tr>
<tr>
<td>1345 Avenue Of The Americas</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.97806, &quot;y&quot;: 40.76309}]</td>
<td></td>
</tr>
<tr>
<td>745 7Th Ave</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.98340, &quot;y&quot;: 40.76077}]</td>
<td></td>
</tr>
<tr>
<td>10020</td>
<td>1221 Avenue Of The Americas</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.98129, &quot;y&quot;: 40.75874}]</td>
</tr>
<tr>
<td>1271 Avenue Of The Americas</td>
<td>[&quot;spatialReference&quot;: {&quot;wkid&quot;: 4326}, &quot;geometryType&quot;: &quot;esriGeometryPoint&quot;, &quot;geometry&quot;: {&quot;x&quot;: 73.98018, &quot;y&quot;: 40.76025}]</td>
<td></td>
</tr>
</tbody>
</table>

**GIS_GEOMETRY:** Building a JSON Geometry Object

The GIS_GEOMETRY function builds a JSON Geometry object given a geometry type, WKID, and a geometry.

**Syntax:** How to Build a JSON Geometry Object

```sql
GIS_GEOMETRY(geotype, wkid, geometry)
```

where:

- **geotype**
  - Alphanumeric
  
  Is a geometry type, for example, 'esriGeometryPolygon', 'esriGeometryPolyline', 'esriGeometryMultipoint', 'EsriGeometryPoint', 'EsriGeometryExtent'..
**wkid**
Alphanumeric

Is a valid spatial reference ID. WKID is an abbreviation for Well Known ID, which identifies a projected or geographic coordinate system.

**geometry**

TX

A geometry in JSON.

The output is returned as TX.

**Example: Building a JSON Geometry Object**

The following request builds a polygon geometry of the area encompassing zip code 10036 in Manhattan. The input geometry object is stored in a text (.ftm) file that is cross-referenced in the esri-citibike Master File. The field containing the geometry object is GEOMETRY.

```sql
DEFINE FILE esri/esri-citibike
WKID/A10  = '4326';
MASTER_GEOMETRY/TX256 (GEOGRAPHIC_ROLE=GEOMETRY_AREA) =
  GIS_GEOMETRY( 'esriGeometryPolygon', WKID , GEOMETRY );
END

TABLE FILE esri/esri-citibike
PRINT
  START_STATION_NAME AS Station
  START_STATION_LATITUDE AS Latitude
  START_STATION_LONGITUDE AS Longitude
  MASTER_GEOMETRY AS 'JSON Geometry Object'
WHERE START_STATION_ID EQ 479
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
type=report, grid=off, size=10,$
ENDSTYLE
END
```
Example: Charting a Geometry Object

The following request uses GIS_GEOMETRY to build a geometry object and chart it on an Esri map.

```
DEFINE FILE esri-citibike
WKID/A10 = '4326';
MASTER_GEOMETRY/TX256 (GEOGRAPHIC_ROLE=GEOMETRY_AREA) =
GIS_GEOMETRY ( 'esriGeometryPolygon', WKID, GEOMETRY );
END

GRAPH FILE ESRI-CITIBIKE
PRINT
START_STATION_NAME
END_STATION_NAME
ON TABLE PC HOLD FORMAT JSC HART
ON TABLE SET LOOK GRAPH CHOROPLETH
ON TABLE SET EMBED HEADING ON
ON TABLE SET AUTOFIT ON
ON TABLE SET STYLE *
TYPE=REPORT, TITLE TEXT='Map', PAGE SIZE=E, CHART-LOOK=COM.ESRI.MAP, $
TYPE=DATA, COLUMN=N1, /*START_STATION_NAME*/
BUCKET=tooltip, $
TYPE=DATA, COLUMN=N2, /*END_STATION_NAME*/
```
The output is shown in the following image.
GIS_IN_POLYGON: Determining if a Point is in a Complex Polygon

Given a point and a polygon definition, the GIS_IN_POLYGON function returns the value 1 (TRUE) if the point is in the polygon or 0 (FALSE) if the point is not in the polygon. The value is returned in integer format.

Syntax: How to Determine if a Point is in a Complex Polygon

GIS_IN_POLYGON(point, polygon_definition)

where:

point
  Alphanumeric or text
  Is the geometry point.

polygon_definition
  Text
  Is the geometry area (polygon) definition.

Example: Determining if a Point is in a Polygon

The following example determines if a station is inside zip code 10036. GIS_IN_POLYGON returns 1 for a point inside the polygon definition and 0 for a point outside. The polygon definition being passed is the same one used in the example for the GIS_GEOMETRY function described previously and defines the polygon for zip code 10036 in Manhattan in New York City. The value 1 is translated to Yes and 0 to No for display on the output.

DEFINE FILE esri/esri-citibike
WKID/A10 = '4326';
MASTER_GEOMETRY/TX256 (GEOGRAPHIC_ROLE=GEOMETRY_AREA) =
  GIS_GEOMETRY('esriGeometryPolygon', WKID, GEOMETRY);
START_STATION_POINT/A200=GIS_POINT(WKID, START_STATION_LONGITUDE,
START_STATION_LATITUDE);
STATION_IN_POLYGON/I4=GIS_IN_POLYGON(START_STATION_POINT, MASTER_GEOMETRY);
IN_POLYGON/A5 = IF STATION_IN_POLYGON EQ 1 THEN 'Yes' ELSE 'No';
END
TABLE FILE esri/esri-citibike
PRINT
  START_STATION_NAME AS Station
  IN_POLYGON AS 'Station in zip, code 10036?'
BY START_STATION_ID AS 'Station ID'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
type=report, grid=off, size=10,$
type=data, column=IN_POLYGON, style=bold, color=red, when = IN_POLYGON eq
  'Yes',$
ENDSTYLE
END
GIS_LINE: Building a JSON Line

Given two geometry points or lines, GIS_LINE builds a JSON line. The output is returned in text format.

Syntax: How to Build a JSON Line

GIS_LINE(geometry1, geometry2)
where:

*geometry1*
Alphanumeric or text
Is the first point or line for defining the beginning of the new line.

*geometry2*
Alphanumeric or text
Is the second point or line for the concatenation of the new line.

**Example: Building a JSON Line**

The following request prints start stations and end stations and builds a JSON line between them.

```sql
DEFINE FILE ESRI/ESRI-CITIBIKE
STARTPOINT/A200 = GIS_POINT('4326', START_STATION_LONGITUDE,
START_STATION_LATITUDE);
ENDPOINT/A200 = GIS_POINT('4326', END_STATION_LONGITUDE,
END_STATION_LATITUDE);
CONNECTION_LINE/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_LINE) =
GIS_LINE(STARTPOINT, ENDPOINT);
END
TABLE FILE ESRI/ESRI-CITIBIKE
PRINT END_STATION_NAME AS End CONNECTION_LINE AS 'Connecting Line'
BY START_STATION_NAME AS Start
WHERE START_STATION_NAME LE 'D'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```
Simplified Geography Functions

The following request generates geometry lines and charts them on an Esri map.

```
DEFINE FILE ESRI-CITIBIKE
CONNECTION_LINE/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_LINE)
=GIS_LINE(START_STATION_POINT, END_STATION_POINT);
DISTANCE/P33.11 TITLE 'Distance'=GIS_DISTANCE(START_STATION_POINT, END_STATION_POINT);
END
```

**Example:** Charting Geometry Lines

The following request generates geometry lines and charts them on an Esri map.

```
DEFINE FILE ESRI-CITIBIKE
CONNECTION_LINE/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_LINE)
=GIS_LINE(START_STATION_POINT, END_STATION_POINT);
DISTANCE/P33.11 TITLE 'Distance'=GIS_DISTANCE(START_STATION_POINT, END_STATION_POINT);
END
```
The GIS_SERVICE_AREA function uses a GIS service to calculate the geometry area with access boundaries within the given time or distance from the provided geometry point. The output is returned in text format.

**Syntax:**

How to Calculate a Geometry Area Around a Point

```
GIS_SERVICE_AREA(geo_point, distance, travel_mode)
```

where:

- **geo_point**
  Alphanumeric
  Is the starting geometry point.

- **distance**
  Alphanumeric
  Is the travel limitation in either time or distance units.
**travel_mode**

Alphanumeric

Is a valid travel mode as defined in gis_serv_area.mas in the Catalog directory under the server installation directory. The accepted travel modes are:

- 'Miles'. This is the default value.
- 'TravelTime'.
- 'TruckTravelTime'.
- 'WalkTime'.
- 'Kilometers'.

**Example: Calculating a Service Area Around a Geometry Point**

The following request calculates the geometry area that is a five-minute walk around a station.

```plaintext
DEFINE FILE esri/esri-citibike
WKID/A10='4326';
START_STATION_POINT/A200=GIS_POINT(WKID, START_STATION_LONGITUDE, START_STATION_LATITUDE);
DISTANCE/A10='5';
TRAVEL_MODE/A10='WalkTime';
STATION_SERVICE_AREA/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_AREA)=GIS_SERVICE_AREA(START_STATION_POINT, DISTANCE, TRAVEL_MODE);
END
TABLE FILE esri/esri-citibike
PRINT
  START_STATION_ID AS 'Station ID'
  START_STATION_NAME AS 'Station Name'
  STATION_SERVICE_AREA AS '5-Minute Walk Service Area Around Station'
WHERE START_STATION_ID EQ 479 OR 512;
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
  TYPE=REPORT, GRID=OFF, SIZE=12,$
ENDSTYLE
END
```
Example: Charting a Geometry Service Area Around a Point

The following request generates service areas that are 5-minute walking distances from start station geometry points and charts them on an Esri map.

```
DEFINE FILE esri-citibike
WKID/A10='4326';
START_STATION_POINT/A200=GIS_POINT(WKID, START_STATION_LONGITUDE, START_STATION_LATITUDE);
DISTANCE/A10='5';
TRAVEL_MODE/A10='WalkTime';
STATION_SERVICE_AREA/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_AREA)=GIS_SERVICE_AREA(START_STATION_POINT, DISTANCE, TRAVEL_MODE);
END
```
GRAPH FILE ESRI-CITIBIKE
PRINT
   START_STATION_NAME
   END_STATION_NAME
   DISTANCE
ON TABLE PCHOLD FORMAT JSCHART
ON TABLE SET LOOKGRAPH CHOROPLETH
ON TABLE SET EMBEDHEADING ON
ON TABLE SET AUTOFIT ON
ON TABLE SET STYLE *
   TYPE=REPORT, TITLETEXT='Map', PAGESIZE=E, CHART-LOOK=com.esri.map, $
   TYPE=DATA, COLUMN=N1, /*START_STATION_NAME*/
      BUCKET=tooltip, $
   TYPE=DATA, COLUMN=N2, /*END_STATION_NAME*/
      BUCKET=tooltip, $
   TYPE=DATA, COLUMN=N3, /*DISTANCE*/
      BUCKET=tooltip, $
*GRAPH_JS_FINAL
"legend": {"visible": true},
"extensions" : { "com.esri.map" :
   { "scalebar" :
      { "scalebarUnit": "dual",
         "attachTo" : "bottom-left"
      },
      "baseMapInfo": { "drawBasemapControl" : false,
         "showArcGISBasemaps" : false,
         "customBaseMaps" : [ {"ibiBaseLayer" : "dark-gray"} ]
      },
      "overlayLayers": [ { "ibiDataLayer": {"map-geometry" : {"map_by_field" : "STATION_SERVICE_AREA"}), "title" : "Chart"]
         },
      "introAnimation": "{"enabled":false}"
   }
   }
*END
ENDSTYLE
HEADING
"Chart Geometry Service Area"
END
GIS_SERV_AREA_XY: Calculating a Service Area Around a Given Coordinate

The GIS_SERV_AREA_XY function uses a GIS service to calculate the geometry area with access boundaries within the given time or distance from the provided coordinate. The output is returned in text format.

Syntax: How to Calculate a Geometry Area Around a Coordinate

GIS_SERV_AREA_XY(longitude, latitude, distance, travel_mode[, wkid])

where:

longitude
   Alphanumeric
   Is the longitude of the starting point.

latitude
   Alphanumeric
   Is the latitude of the starting point.

distance
   Integer
   Is the travel limitation in either time or distance units.
**travel_mode**

Alphanumeric

Is a valid travel mode as defined in gis_serv_area.mas in the Catalog directory under the server installation directory. The accepted travel modes are:

- 'Miles'. This is the default value.
- 'TravelTime'.
- 'TruckTravelTime'.
- 'WalkTime'.
- 'Kilometers'.

**wkid**

Alphanumeric

Is the spatial reference ID for the coordinate. WKID is an abbreviation for Well Known ID, which identifies a projected or geographic coordinate system. The default value is '4326', which represents decimal degrees.

**Example: Calculating a Service Area Around a Coordinate**

The following request calculates the geometry area that is a five-minute walk around a station, using the longitude and latitude that specify the station location.

```sql
DEFINE FILE esri/esri-citibike
DISTANCE/I4=5;
WKID/A10='4326';
TRAVEL_MODE/A10='WalkTime';
STATION_SERVICE_AREA/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_AREA)=
   GIS_SERV_AREA_XY(START_STATION_LONGITUDE, START_STATION_LATITUDE,
   DISTANCE, TRAVEL_MODE, WKID);
END
TABLE FILE esri/esri-citibike
PRINT
   START_STATION_ID AS 'Station ID'
   START_STATION_NAME AS 'Station Name'
   STATION_SERVICE_AREA
   AS '5-Minute Walk Service Area Around Station Coordinate'
WHERE START_STATION_ID EQ 479 OR 512;
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, SIZE=12,$
ENDSTYLE
END
```
Example: Charting a Geometry Service Area Around a Coordinate

The following request generates service areas that are 5-minute walking distances from start station coordinates and charts them on an Esri map.

```sql
DEFINE FILE esri-citibike
WKID/A10='4326';
DISTANCE/A10='5';
TRAVEL_MODE/A10='WalkTime';
STATION_SERVICE_AREA/TX80 (GEOGRAPHIC_ROLE=GEOMETRY_AREA) =
GIS_SERV_AREA_XY(START_STATION_LONGITUDE, START_STATION_LATITUDE, DISTANCE, TRAVEL_MODE, WKID);
END
```
5. Reporting Language Enhancements

GRAPH FILE ESRI-CITIBIKE
PRINT
   START_STATION_NAME
   END_STATION_NAME
   DISTANCE
ON TABLE PCHOLD FORMAT JSCHART
ON TABLE SET LOOKGRAPH CHOROPLETH
ON TABLE SET EMBEDHEADING ON
ON TABLE SET AUTOFIT ON
ON TABLE SET STYLE *
   TYPE=REPORT, TITLETEXT='Map', PAGESIZE=E, CHART-LOOK=com.esri.map, $
   TYPE=DATA, COLUMN=N1, /*START_STATION_NAME*/
      BUCKET=tooltip, $
   TYPE=DATA, COLUMN=N2, /*END_STATION_NAME*/
      BUCKET=tooltip, $
   TYPE=DATA, COLUMN=N3, /*DISTANCE*/
      BUCKET=tooltip, $
*GRAPH_JS_FINAL
"legend": {"visible": true},
"extensions" : { "com.esri.map" :
   { "scalebar" :
      { "scalebarUnit": "dual",
         "attachTo" : "bottom-left"
      },
   "baseMapInfo": {
      "drawBasemapControl" : false,
      "showArcGISBasemaps" : false,
      "customBaseMaps" : [
         {"ibiBaseLayer" : "dark-gray"}
      ]
   },
   "overlayLayers": [
      {"ibiDataLayer": {"map-geometry" : {"map_by_field" : "STATION_SERVICE_AREA"}}, "title" : "Chart"}]
   },
"introAnimation": "{\"enabled\":false}"
}
*END
ENDSTYLE
HEADING
"Chart Geometry Service Area"
END
Support for the Comma Character as the Operation in an Autoprompt Dynamic Multi-Select List

As of WebFOCUS Release 8.2, the comma character is supported as an operation in an Autoprompt dynamic multi-select list with the IN and NOT IN operators. The IN and NOT IN operators optimize performance when running report requests to retrieve data from SQL data sources, as well as FOCUS selection.

In the following example, the comma separator identifies the values that must be quoted for Autoprompt processing.

```
TABLE FILE CAR
PRINT CAR
WHERE COUNTRY IN (&COUNTRIES.(,(FIND COUNTRY IN CAR)).COUNTRY LIST.)
END
```

Both alphanumeric and numeric values can be quoted. Alphanumeric values are required to be quoted for the IN list operator. Quotation marks are optional for numeric values.

When Autoprompt processes the selected values, they are sent to the Reporting Server with FOCUS quote encoding (multiple quotation marks), so that the FOCUS parser processes the value enclosed in quotation marks.

The output is shown in the following image.
Example: Single Value Selection

In the following example, the value ITALY is enclosed in three single quotation marks. Autoprompt processing adds the -SET operator to the request code sent to the Reporting Server.

```
-SET &COUNTRIES="ITALY";
-SET &SEATS="5";
```

Example: Multiple Value Selection

In the following example, the values ENGLAND and FRANCE are selected. Each individual value is enclosed in two single quotation marks and the entire series of values is enclosed in single quotation marks. Autoprompt processing adds the -SET operator to the request code sent to the Reporting Server.

```
-SET &COUNTRIES="ENGLAND", "FRANCE";
```

Automatic Conversion Between Date and Date-Time Formats

The following automatic direct operations are supported between date and date-time formats:

- **Assignment.**
  
  **Assignment of a date field or a date constant to a date-time field.** The time component is set to zero (midnight). The date can be a full component date such as YYMD or a partial component date such as YYM. It cannot be a single component date such as Q, as this type of date, although displayed as a date in reports, is stored as an integer value and is used as an integer value in expressions.

  **Assignment of a date-time field or date-time constant to a date field.** The time components are removed.

- **Comparison and Subtraction.**

  When a date-time value is compared with or subtracted from a date value, or a date value is compared with or subtracted from a date-time value, the date is converted to date-time with the time component set to midnight. They are then compared or subtracted as date-time values.

- **Function parameters.**

  Simplified date functions can use either date or date-time values as their date parameters. Legacy user functions do not support this new functionality. The date-time functions (H functions) use date-time parameters and the new date functions use new dates, which are stored as offsets from a base date.
Recognition and use of date or date-time constants.

- Constants can be expressed as strings, without the DT operator.

- Constants are converted to or from date or date-time values in accordance with the field format they are compared with, subtracted from, or assigned to.

- Unless it is expressed in a non-ambiguous translated or formatted string format with proper delimiters (not as a numeric string or number), the recognition of a constant as a date depends on the format of its field counterpart.

  In this case, the size in terms of number of digits is strictly limited to at least six for a full component date or date-time value, (eight for a four-digit year), three for a partial component date, and one for a single component date.

- When numeric constants are used as function parameters and, therefore, do not have a field counterpart, they are recognized according to YYMD or YMD format. The only exception is a string with a single blank or the number zero which, in reports, will be presented as a blank. Date offset constants are no longer allowed. Blank separators between digits in a string are also not supported.

For information about creating and using date and date-time constants, see the *Describing Data With WebFOCUS Language* manual.

Example: Assigning Date and Date-Time Values

The following request generates a date-time value using the DT_CURRENT_DATETIME function. It then assigns this value to a date field and assigns that date field to a date-time field.

```
TABLE FILE WF_RETAIL_LITE
PRINT QUANTITY_SOLD NOPRINT AND COMPUTE
  DATETIME1/HYYMDm = DT_CURRENT_DATETIME(MILLISECOND);
  AS 'Date-Time 1'
COMPUTE
  DATE1/YYMD = DATETIME1;
  AS 'Date'
COMPUTE
  DATETIME2/HYYMDm = DATE1;
  AS 'Date-Time 2'
WHERE RECORDLIMIT EQ 20
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```
The output is shown in the following image. The original date-time field has a non-zero time component. When assigned to the date field, the time component is removed. When that date is assigned to the second date-time field, a zero time component is added.

<table>
<thead>
<tr>
<th>Date-Time 1</th>
<th>Date</th>
<th>Date-Time 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.855000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
<tr>
<td>2017/08/25 09:10:06.856000</td>
<td>2017/08/25</td>
<td>2017/08/25 00:00:00.000000</td>
</tr>
</tbody>
</table>
**Example:** Comparing Date and Date-Time Values

The following request creates one date-time field and one date field. When quantity sold is 1, they have the same date value and the date-time field has a zero time component. When quantity sold is 2, they have different date values, and the date-time field has a zero time component. In all other cases, the date-time field has the current date with a non-zero time component, and the date field has the current date. The EQUAL1 field compares them to see if they compare as equal.

```plaintext
TABLE FILE WF_RETAIL_LITE
PRINT QUANTITY_SOLD AS Quantity AND COMPUTE
  DATETIME1/HYYMDm = IF QUANTITY_SOLD EQ 1 THEN '2017/06/05'
                  ELSE IF QUANTITY_SOLD EQ 2 THEN '2016/02/29'
                  ELSE DT_CURRENT_DATETIME(MILLISECOND);
  AS 'Date-Time'
COMPUTE
  DATE1/YYMD   = IF QUANTITY_SOLD EQ 1 THEN '2017/06/05'
                  ELSE IF QUANTITY_SOLD EQ 2 THEN '2015/12/30'
                  ELSE DT_CURRENT_DATE();
  AS 'Date'
COMPUTE
  EQUAL1/A1 = IF DATETIME1 EQ DATE1 THEN 'Y' ELSE 'N';
  AS 'Equal?'
WHERE RECORDLIMIT EQ 12
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE,$
END
```
The output is shown in the following image. When a date value is compared to a date-time value, the date is converted to a date-time value with the time component set to zero, and then the values are compared. Therefore, when QUANTITY_SOLD is 1, both the date components are equal and the time component of the date-time field is set to zero, so when the date is converted to a date-time value, they are equal. When QUANTITY_SOLD is 2, the date components are different, so they are not equal. When QUANTITY_SOLD is 3, the date components are the same, but the date-time field has a non-zero time component. Therefore, when the date field is converted to a date-time value with a zero time component and they are compared, they are not equal.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Date-Time</th>
<th>Date</th>
<th>Equal?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.0000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.000000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>2016/02/29 00:00:00.00000000</td>
<td>2015/12/30</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>2016/02/29 00:00:00.00000000</td>
<td>2015/12/30</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>2017/08/25 09:24:45.2030000</td>
<td>2017/08/25</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>2017/06/05 00:00:00.00000000</td>
<td>2017/06/05</td>
<td>Y</td>
</tr>
</tbody>
</table>
Index

C

CEILING function 177
CHAR function 196
character functions, simplified
    CONCAT 187
    LAST_NONBLANK 183
    REGEX 185
    REPLACE 189
    SPLIT 192
CHECKMD5 function 193
CHECKSUM function 194
CONCAT function 187
conversion functions, simplified
    CHAR 196
    CTRLCHAR 197
    EDIT2 199
    HEXTYPE 202
    PHONETIC 204
    TO_INTEGER 206
    TO_NUMBER 207
CTRLCHAR function 197
CURRENCY_DISPLAY parameter 141
CURRENCY_ISO_CODE parameter 141
CURRENCY_PRINT_ISO parameter 142

date functions, simplified
    DT_CURRENT_DATE 208
    DT_CURRENT_DATETIME 208
    DT_CURRENT_TIME 209
DATE_ORDER parameter 145
DATE_SEPARATOR parameter 145
DT_CURRENT_DATE function 208
DT_CURRENT_DATETIME function 208
DT_CURRENT_TIME function 209
DTRUNC function
    MONTH_END parameter 175
    QUARTER_END parameter 175
    WEEK parameter 174
    WEEK_END parameter 175
    YEAR_END parameter 175

E

EDIT2 function 199
ENCRYPT function 210
EXPONENT function 178

F

FLOATMAPPING parameter 139
FLOOR function 179

G

gEOGRAPHY functions 212
    GIS_DISTANCE 218
    GIS_DRIVE_ROUTE 220
Index

geography functions 212
  GIS_GEOCODE_ADDR 227
  GIS_GEOCODE_ADDR_CITY 228
  GIS_GEOCODE_ADDR_POSTAL 230
  GIS_GEOMETRY 231
  GIS_IN_POLYGON 235
  GIS_LINE 236
  GIS_POINT 224
  GIS_SERV_AREA_XY 244
  GIS_SERVICE_AREA 240
GETENV function 211
GIS_DISTANCE function 218
GIS_DRIVE_ROUTE function 220
GIS_GEOCODE_ADDR function 227
GIS_GEOCODE_ADDR_CITY function 228
GIS_GEOCODE_ADDR_POSTAL function 230
GIS_GEOMETRY function 231
GIS_IN_POLYGON function 235
GIS_LINE function 236
GIS_POINT function 224
GIS_SERV_AREA_XY function 244
GIS_SERVICE_AREA function 240

LDAP security with SSL 49

M
MD5 hash value 193
MOD function 181

N
numeric functions, simplified
  CEILING 177
  EXPONENT 178
  FLOOR 179
  MOD 181
  POWER 182

O
ONFIELD SET parameter 133

P
p display option 173
PARTITION_AGGR function 153
PARTITION_REF function 163
PCOMMA parameter 173
percentage display option 173
PHONETIC function 204
POWER function 182
prefix operators 170
prior values 163
PUTENV function 211

H
hash value 193, 194
HEXTYPE function 202

L
lag values 163
LAST_NONBLANK function 183
R

REGEX function 185
REPLACE function 189
rolling calculations 153

S

Secure Sockets Layer (SSL) with LDAP security 49
SET parameters
  CURRENCY_DISPLAY 141
  CURRENCY_ISO_CODE 141
  CURRENCY_PRINT_ISO 142
  DATE_ORDER 145
  DATE_SEPARATOR 145
  FLOATMAPPING 139
  TIME_SEPARATOR 146
SPLIT function 192
SSL/TLS connections 49
  49
system functions, simplified
  ENCRYPT 210
  GETENV 211
  PUTENV 211

T

TIME_SEPARATOR parameter 146
TO_INTEGER function 206
TO_NUMBER function 207
TOTAVE operator 170
TOTCNT operator 170
TOTMAX operator 170
TOTMIN operator 170

W

WHERE_GROUPED 147
Feedback

customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at Sarah_Buccellato@ibi.com.

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at Frances_Gambino@ibi.com.
WebFOCUS

Server New Features

WebFOCUS Reporting Server Release 82 DataMigrator Server Release 7707